Sensor-based Activity Recognition: One Picture Is Worth a Thousand Words

Daniele Riboni*, Marta Murtas

Department of Mathematics and Computer Science, University of Cagliari, via Ospedale 72, I-09122 Cagliari, Italy

Abstract

In several domains, including healthcare and home automation, it is important to unobtrusively monitor the activities of daily living (ADLs) carried out by people at home. A popular approach consists in the use of sensors attached to everyday objects to capture user interaction, and ADL models to recognize the current activity based on the temporal sequence of used objects. Often, ADL models are automatically extracted from labeled datasets of activities and sensor events, using supervised learning techniques. Unfortunately, acquiring such datasets in smart homes is expensive and violates users' privacy. Hence, an alternative solution consists in manually defining ADL models based on common sense, exploiting logic languages such as description logics. However, manual specification of ADL ontologies is cumbersome, and rigid ontological definitions fail to capture the variability of activity execution. In this paper, we introduce a radically new approach enabled by the recent proliferation of tagged visual contents available on the Web. Indeed, thanks to the popularity of social network applications, people increasingly share pictures and videos taken during the execution of every kind of activity. Often, shared contents are tagged with metadata, manually specified by their owners, that concisely describe the depicted activity. Those metadata represent an implicit *activity label* of the picture or video. Moreover, today's computer vision tools support accurate ex-

*Corresponding author
Email address: riboni@unica.it (Daniele Riboni)

traction of *tags* describing the situation and the objects that appear in the visual content. By reasoning with those tags and their corresponding activity labels, we can reconstruct accurate models of a comprehensive set of human activities executed in the most disparate situations. This approach overcomes the main shortcomings of existing techniques. Compared to supervised learning methods, it does not require the acquisition of training sets of sensor events and activities. Compared to knowledge-based methods, it does not involve any manual modeling effort, and it captures a comprehensive array of execution modalities. Through extensive experiments with large datasets of real-world ADLs, we show that this approach is practical and effective.

Keywords: Activity recognition, intelligent systems, pervasive computing, activity models, unsupervised reasoning

1. Introduction

Nowadays, activity recognition is a key requirement in several ICT domains [1], including smart home automation, homeland security, e-health, gaming, manufacturing, pervasive advertising, and smart cities, just to name a few. In particular, advanced healthcare systems rely on continuous monitoring of human activities to support early detection of health issues, to enhance rehabilitation, and to promote healthy and active lifestyles [2, 3]. These applications take into account not only simple physical activities, but also a wide range of complex activities of daily living (ADLs). As a consequence, several techniques to accurately recognize ADLs have been proposed in the last years. A popular approach to activity recognition consists in the use of supervised learning methods applied to datasets of activities and sensor data [4, 5, 6]. Supervised learning proves to be effective in recognizing activities characterized by specific postures or motions, such as physical activities [7]. The supervised learning approach was also successfully applied to the recognition of high-level urban activities based on GPS traces [8, 9]. However, the actual applicability of the supervised approach to recognize complex ADLs at a fine-grained level is ques-

tionable, especially when infrequent or sporadic activities are taken into account. Indeed, acquiring large datasets of ADLs is expensive in terms of annotation costs [10, 11]. Moreover, activity annotation by an external observer, by means of cameras or direct observation, severely violates the user's privacy.

For this reason, different research groups investigated unsupervised approaches for recognizing ADLs. An interesting direction in this sense consists in disregarding the activity semantics, and recognizing recurrent patterns of abstract actions and their temporal variations, as proposed in [12]. That approach can be applied to the early recognition of particular health issues. However, for many other applications domains, knowledge of the activity semantics is of foremost importance. In those domains, most unsupervised methods rely on symbolic modeling of activities in terms of their constituting simpler actions. For instance, the temporal sequence of events "open medicine cabinet; take medicine box; put away medicine box; close medicine cabinet" characterizes the ADL "taking medicines". A popular approach is to manually define those models through formal ontologies expressed in a description logics language [13, 14, 15]. However, manually defining comprehensive ADLs ontologies is cumbersome, and requires specific expertise in formal ontology languages and knowledge engineering. Moreover, the ontological approach is generally based on rigid activity definitions, that fall short in adapting to dynamic context conditions.

In order to overcome the limitations of current methods, in this paper, we introduce a radically new approach. The approach starts from the observation that, thanks to the widespread popularity of social network applications, users are sharing more and more pictures and videos illustrating the execution of every kind of activity in the most disparate situations. Our intuition is that the ability of extracting semantic tags from those visual resources through computer vision tools may enable novel data mining methods to infer activity models at essentially no cost. Indeed, shared contents are frequently labeled by their owners with tags describing the depicted activity. Those tags provide an implicit *activity label*. Other tags can be extracted by computer vision tools to acquire semantic information about depicted objects, actors, and context condi-

tions (e.g., time of the day, light level, symbolic location). Hence, by reasoning with activity labels and contextual tags, it is theoretically possible to build a comprehensive set of activity models, without the need of sensor-based training sets and without manual modeling effort. To the best of our knowledge, this is the first work that addresses this research direction, except for our preliminary investigation reported in [16]. Several research issues are involved in this work, including search of the most representative images or videos, computer vision algorithms for tag extraction, and data mining methods to construct the activity models. For the sake of this paper, we concentrate on activity model extraction from still images, where images may include frames extracted from videos. However, the approach can be extended to mine more sophisticated models, considering temporal relationships captured from activities recorded in videos.

We point out that several previous works tried to recognize activities based on images or videos [17, 18, 19, 20]. However, the goal of those works was vision-based activity recognition; i.e., the model extracted from a training set of images or videos was used to recognize the activity depicted in other images or videos. In this paper, we investigate a completely different approach: mining activity models from pictures and videos, and using them to recognize activities based on firing of sensor events. Thus, in our work, the domain of mined data (visual contents) is disjoint from the one of data used for activity recognition (sensor data). A preliminary investigation of this approach was presented in [16], where we introduced the use of computer vision tools to extract object information from images depicting activity execution. In this paper, we extend our previous work with (i) activity mining from videos, (ii) a novel activity recognition technique, (iii) use of additional computer vision tools, and (iv) extensive experiments with additional datasets.

The main contributions of this work are the following:

- We introduce a novel approach to unsupervised sensor-based recognition of human activities, which exploits tagged visual contents shared on the

Web, and computer vision tools.

- We propose a probabilistic-based method to extract activity models from pictures and videos.
- We present the results of extensive experiments with two large datasets of real-world ADLs and different computer vision tools, showing the effectiveness and practicality of our approach.

The rest of the paper is structured as follows. Section 2 discusses related work. Section 3 introduces our methodology. Section 4 presents our algorithms. Section 5 illustrates the methods to recognize activities based on the extracted models. Section 6 presents the experimental evaluation. Section 7 discusses strong and weak points of the approach. Section 8 concludes the paper.

2. Related work

Activity recognition (AR) systems can be broadly divided into vision-based [17] and sensor-based [5] ones. Vision-based AR systems make use of cameras and scene recognition algorithms to recognize the situation, including the activity carried out by people appearing in the video stream. Those systems are used in application domains such as video surveillance, security, entertainment, and rehabilitation. However, their applicability is restricted to confined environments, and the usage of cameras determines relevant issues in terms of privacy and enforcement of regulations. On the contrary, sensor-based AR systems rely on events fired by different kinds of sensors, which are wore by people or embedded in the environment. Those sensors allow monitoring basic human actions such as gestures and interactions with furniture and appliances, as well as contextual conditions (e.g., power consumption, light level, presence of gases) determined by the execution of specific activities. Since more and more sensors are invisibly integrated in everyday objects, those systems are generally perceived as less intrusive than vision-based ones [21]. For this reason, in this paper, we pursue the sensor-based approach.

Sensor-based AR techniques can be classified in two main categories: data-driven and knowledge-driven ones [22]. While the former rely on datasets of activities to derive the activity model, the latter rely on manual definition of those models by means of formal ontologies, rules, or other logical formalisms. Most data-driven AR systems adopt supervised learning methods to infer activity models based on a training set of labeled activities and sensor data. Early AR systems are based on multiple accelerometers worn at different body locations [23, 24]. Of course, those systems are quite obtrusive; hence, later efforts were devised to recognize simple activities based on a single accelerometer, possibly integrated in a smartphone [25]. An indoor device-free activity recognition system based on passive RFID tags was presented in [26]. The system detects the user's position through the analysis of RSSI patterns of signals emitted by RFID tags on the wall. The recognition of transitions between different positions is achieved by a HMM-based approach. This method enables the early detection of abnormal actions, and thus an early intervention by raising immediate alarms. A supervised learning method was also proposed by Zheng et al. to recognize high-level activities based on traces of GPS locations [8, 9]. Huang et al. proposed a framework to support reasoning on the edge that included streaming-based activity recognition [27]. In order to recognize complex activities, other supervised systems proposed the use of additional data, such as noise level, temperature, and objects usage [28, 29, 30, 31, 32].

Even though multiuser activities constitute large part of people's daily lives, most of the literature focuses the attention on single-user activities. Indeed, recognising multi-user activities in pervasive environments introduces additional challenges. In [33] the authors propose the epMAR system, which is able to recognise both single and multi-user activities using wireless sensors worn by the users. Their solution is based on Emerging Patterns describing significant differences between two classes of data, and on a variable-length sliding window technique. For the sake of this work, we concentrate on the recognition of single-user activities. However, our image-mining approach could be extended to support recognition of multi-user activities. However, when complex ADLs

are considered, the supervised approach incurs high costs in terms of training set acquisition. Indeed, a comprehensive training set should include ADLs executed by a large set of heterogeneous individuals in different real-world situations. The acquisition of such datasets is expensive, time-consuming, and may violate the individuals privacy, since an external observer must annotate the start and end time of activities. For these reasons, semi-supervised learning approaches have been proposed in the last years. In [34], the authors propose a semi-supervised method for activity recognition that uses $\ell_{2,1}$ minimisation for outlier detection, and a graph-based label propagation method for categorizing unlabelled data. Their objective is the minimisation of the intra-class variability, which is due to the different ways in which the same persons can perform an activity. To this aim, the technique searches for a subspace of the original feature space that can be used as a signature of each activity.

Due to the shortcomings of supervised approaches, different researchers investigated unsupervised techniques. Unsupervised AR methods based on data are mainly devoted to recognize high-level variations in the normal pattern of actions in order to early detect health issues, as proposed by Rashidi and Cook in [12]. However, those approaches disregard the semantics of activities, which is an important factor to consider for several applications. In order to consider semantics, other works were based on ontologies expressed through description logic languages, possibly extended with rules, to model ADLs based on their constituting simpler actions [35, 36, 37]. An unsupervised method for activity segmentation was proposed in [38]. The method is based on an ontological model inspired by Semiotic theory, which captures generic knowledge and individual-specific features about ADLs execution. In ontology-based systems, activity recognition is based on the observation of temporal sequences of sensor events that match the definition of actions defining a given activity. However, that approach has limits in the rigidity of activity descriptions and burden of manually defining the ontological axioms.

A less investigated approach consists in mining activity models from the Web. A first attempt in this sense was due to Perkowitz et al. in [39] and

refined in later works [40, 41]. The input is an activity label such as "cleaning kitchen". The activity label is used as a query for a Web search engine, to find pages related to that activity. The textual content of the top k pages is passed to an object identification module, which exploits a lexical database to extract the key objects related to the activity. A statistical method is used to obtain the set of top-j related objects, weighted considering their frequency in the Web pages. Finally, for each object $o \in O$ (O being the set of objects) with weight w, and each activity $a \in A$ (A being the set of activities), the probability of using o during a is computed. The probability distribution $p(O|A)$ is used by a generative model to reconstruct the most probable sequence of activities given an observed sequence of used objects.

A similar approach was used by Pentney et al. in [42], exploiting user-contributed common sense acquired by the Open Mind Indoor Common Sense project [43]. Gu et al. presented a different method to extract activity models from the text of Web pages [44]. In that work, object-use fingerprints are extracted in terms of *contrast patterns*, which describe statistically significant differences in object-use patterns among any couple of activities. A similar method was used by Palmes et al. in [45] for both activity recognition and segmentation. Ihianle et al. proposed a method to mine the textual content of Web pages for identifying the most probable activity given a temporal sequence of used objects [46]. The most probable activities are inferred based on a combination of statistical and ontological reasoning. Differently from the above mentioned works, Nakatani et al. apply the Web mining approach to recognize activities based on video streams, not from sensor data [47]. Their method consists in recognizing the objects used by a subject based on egocentric cameras, and mining the Web to derive associations among objects and activities. The activity with highest correlation values is the predicted activity.

Even though they adopt different techniques for inferring the relevance among objects and activities, to the best of our knowledge, all existing Web-based activity mining methods rely on textual content only. Instead, in this work we investigate a different approach: exploiting visual contents available on the Web

Figure 1: Pictures related to "preparing a birthday gift" retrieved by an image search engine

Figure 2: Pictures related to "cleaning kitchen" retrieved by an image search engine

to automatically build activity models. The intuition of our work is that visual data provide much more compact and expressive information than text found on Web pages. Moreover, temporal information in video streams may enable the inference of activity models considering temporal constraints about occurred actions and events, while existing Web-based activity mining methods disregard temporal information.

3. Methodology and algorithms

In this section, at first we illustrate the rationale of our methodology and the overall architecture. Then, we explain our methods and algorithms for extracting activity-related features from visual resources and for building activity models.

3.1. Rationale and methodology

Our work starts from the intuition that visual contents publicly available on the Web provide concise and diverse information about activity execution. Consider for instance the pictures illustrated in Figures 1 and 2. Those pictures

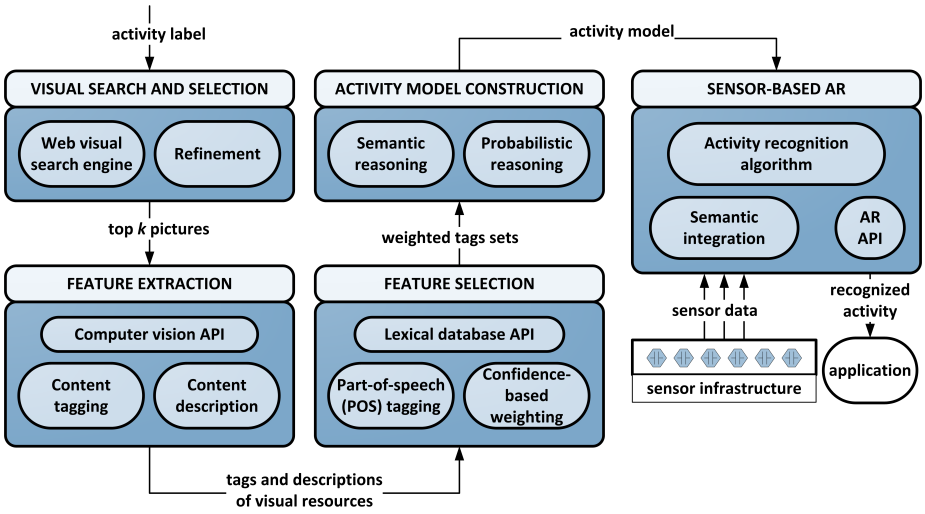


Figure 3: The overall architecture to implement our methodology.

were retrieved querying an image search engine, issuing queries "preparing a birthday gift" (Figure 1) and "cleaning kitchen" (Figure 2). Those pictures include several objects typically used to perform the searched activity. Indeed, pictures regarding "preparing a birthday gift" include gift boxes, scissors, ribbons, scotch tape, while pictures regarding "cleaning kitchen" include broom, rag, bucket, sink. Pictures also depict actions taken to perform the activity, such as "cut", "write", "sweep". Moreover, those pictures provide insights about the context of execution of depicted activities, including light level, indoor/outdoor location, age and gender of actors, and posture. It is worth noting that pictures capture different ways to perform the same activity. Moreover, frequently those pictures concentrate on the key objects and actors regarding the activity, disregarding irrelevant details.

Our goal is to mine Web visual contents for automatically building activity models to be used for sensor-based AR. Our methodology is composed of five main elements:

1. search and refinement of activity-related pictures and videos;
2. feature extraction from retrieved resources;

3. feature selection and weighting;
4. activity model construction;
5. sensor-based activity recognition.

3.2. Overall architecture

The overall architecture to implement our methodology is illustrated in Figure 3.

3.2.1. Visual search and refinement

Given an activity label (e.g., "preparing a birthday gift"), the first module (VISUAL SEARCH AND REFINEMENT) queries a Web search engine to find the top k visual resources that match the label. Those visual resources are either images directly retrieved from a Web image search engine, or frames extracted from videos retrieved from a video search engine. For the sake of simplicity, in the following of the paper we refer to both images and frames using the term *pictures*. The query result is refined considering a set of *filtering directives*. Those directives ensure that retrieved pictures satisfy requirements concerning size and resolution, in order to provide sufficient information to the computer vision software.

3.2.2. Feature extraction

Remaining pictures are given to the FEATURE EXTRACTION module, which queries a Computer vision API to extract a description and tags of elements identified in those pictures. The description briefly summarizes the picture content (e.g., "a person wrapping up a gift"). Picture tags refer to objects, actions, and other contextual information found in the picture. For instance, tags found for the first picture in Figure 1 include "gift box", "writing", "indoor", "brown". Each tag is associated to a confidence value, which measures the probability that the element actually appears in the picture or represents its content.

3.2.3. Feature selection

Pictures' tags and descriptions are passed to the FEATURE SELECTION module, which applies part-of-speech (POS) tagging to keep only those terms that are useful to describe the execution of the activity (actions and objects), as well as relevant context conditions. In particular, we retain only *nouns* that represent:

- physical objects and furniture (e.g., "scissors", "table"),
- symbolic locations (e.g., "indoor", "dining room"),
- time of day or light conditions (e.g., "morning", "dim light"),

or *verbs* that represent actions or postures (e.g., "cutting", "lying").

For each term found in a picture's tag, we keep track of its confidence value, provided by the computer vision software. Terms extracted from the description are assigned confidence 1; indeed, the picture description is considered accurate.

3.2.4. Activity model construction

The procedure explained above is executed for each activity a belonging to the set A of considered activities. Then, for each considered activity a and each tag $t \in T$ (T being the set of tags), this module probabilistically estimates $p(a|t)$; i.e., the conditional probability that the current activity is a given the observation of t, where t can correspond to either an action, or to the usage of an object. The algorithm to compute $p(A|T)$ is described in Section 4.2.

3.2.5. Sensor-based activity recognition

The sensor-based activity recognition module is in charge of acquiring data from the sensor infrastructure, executing the AR algorithm using the computed model, and communicating the recognized activities to external applications through an API.

This module includes a SEMANTIC INTEGRATION component, which maps occurred sensor events to tags extracted from pictures. This mapping is necessary, since the extracted model is based on tags, and it does not directly consider

sensor events. Essentially, that component implements a function $M : S \to T$, where S is the set of sensor events that can be captured in the smart environment, and T is the set of tags extracted from pictures. The component exploits a very simple OWL 2 ontology [48], named ArOnt, modeling a generic set of tags, sensors, objects, and actions. The ontology is available on the Web[1]. The generic ontology is instantiated considering the current environment and tags extracted from pictures.

Example 1. *Consider the CASAS smart home, which we used as one of the environments for our experimental evaluation reported in Section 6. The smart home is equipped with several sensors, including sensors attached to doors, items, and forniture. For each sensor, a corresponding instance is added to the ontology. For instance, the following axiom:*

$$d11 : MedCabinetDoorSensor$$

states that the apartment includes a sensor $d11$ *that is an instance of* MedCabinetDoorSensor. *The latter is a subclass of* CabinetSensor*, and represents a sensor attached to a medicine cabinet door. Similarly, the axiom:*

$$medicines : Tag$$

states that medicines *is an instance of the class* Tag.

While the set of sensors must be manually specified, possibly using user-friendly interfaces, the set of tags is automatically retrieved from computer vision API calls and added to the ontology.

The instantiated ontology specifies which sensors are triggered by the use of sensorized objects or by the execution of basic actions, through the datatype property *triggers*.

Example 2. *Continuing Example 1, the following axiom states that the action* openMedCabinet *triggers the sensor* $d11$*:*

$$\langle openMedCabinet, d11 \rangle : triggers,$$

[1] `http://sites.unica.it/domusafe/aront/`

where OPENMEDCABINET *is an instance of the ontological class* ACTION.

The ontology also maps each action or object to the corresponding tag through the datatype property *correspondsTo*.

Example 3. *Continuing the above example, the following axiom states that the action* OPENMEDCABINET *corresponds to the tag medicines:*

$$\langle openMedCabinet, medicines \rangle : correspondsTo.$$

Axioms defining correspondences among actions/objects and tags are environment-independent, and can be pre-defined in the ontology.

The function M is implemented in our ontology using the property composition operator $\circ$ of OWL 2:

$$M\_function \subseteq triggers^{-} \circ correspondsTo,$$

where $triggers^{-}$ denotes the inverse of the *triggers* property. Hence, for each sensor $s \in S$, the module performs ontological reasoning to find the set T_s of tags that are related to the activation of s.

Example 4. *Continuing the previous example, ontological reasoning derives through the property composition operator that the sensor* $d11$ *is related to the tag* medicines *according to the* $M\_function$*. Note that each tag is automatically augmented with its synonyms; hence, they do not need to be manually added to the ontology.*

Then, the temporal sequence of sensor activations is mapped to a temporal sequence of tags, which is used by the AR algorithm to detect the current activity. Ontological reasoning is performed offline in order to avoid computational overhead at activity recognition time.

The extracted activity model can be used to apply different activity recognition methods. In the experimental evaluation, we use the model with two methods. The first one relies on a sliding window to consider the last n sensor events, and applies a temporal smoothing function to give more confidence

to the more recent events. The second one relies on the Markov Logic Networks (MLN) probabilistic logic. However, the model can be applied, with minor modifications, to several other AR algorithms, including those presented in [44, 45, 49, 50, 51].

4. Algorithms

In this section, we formally describe the algorithms used to implement the modules illustrated above.

4.1. Retrieving weighted visual features

The algorithm *VisualExtraction* implements the modules for VISUAL SEARCH AND FILTERING, FEATURE EXTRACTION, and FEATURE SELECTION shown in Fig. 3. The algorithm pseudo-code is shown in Fig. 4. It takes as input an activity label a and the number k of visual objects to be used to identify the tags related to the execution of a. Each tag is essentially a feature that represents a characteristic of the activity execution. The output is the set of identified tags, together with their respective weights.

4.1.1. Visual search and refinement

After initializing a set P of pictures to the empty set, the algorithm queries a visual Web search engine to get the top k visual objects that respond to the query 'a'. Those objects (i.e., images, or frames sampled from video clips) are added to P. For each picture, the algorithm checks whether it fulfills the filtering directives D regarding size and resolution. Any image not satisfying D is removed from P. After filtering, if the size of P is less than k, the algorithm queries the search engine to download other k visual objects, and applies filtering on them. This process is repeated until P contains at least k pictures (lines 2 to 5). Then, the algorithm selects the top k pictures according to the relevance of their visual objects, computed by the search engine, and removes the other ones from P (line 6).

Algorithm ***VisualExtraction***(a, k, D):

Input: Activity label a; number k of visual objects; filtering directives D

Output: Weighted tags sets for activity a

1: $P \leftarrow \emptyset$
2: **repeat**
3: $\quad P \leftarrow P \cup VisualWebSearch(a, k)$
4: $\quad P \leftarrow Filtering(P, D)$
5: **until** $|P| \geq k$
6: $P \leftarrow top\text{-}k(P)$
7: **for each** $p_i \in P$ **do**
8: $\quad < tags_i, desc_i > \leftarrow VisualAnalysis(p_i)$
9: $\quad$ **for each** $tag \in tags_i$ **do**
10: $\quad\quad$ **if** $POS\text{-}tag(tag.label) \notin \{`act', `object', `artifact', ...\}$ **then**
11: $\quad\quad\quad tags_i \leftarrow tags_i - \{tag\}$
12: $\quad\quad$ **end if**
13: $\quad$ **end for**
14: $\quad terms \leftarrow POS\text{-}extraction(desc_i)$
15: $\quad$ **for each** $term \in terms$ **do**
16: $\quad\quad$ **if** $POS\text{-}extr(term) \in \{`act', `object', `artifact', ...\}$ **then**
17: $\quad\quad\quad$ **if** $\exists\, t \in tags_i \, such\, that : t.label = term$ **then**
18: $\quad\quad\quad\quad t.conf \leftarrow 1$
19: $\quad\quad\quad$ **else**
20: $\quad\quad\quad\quad tag.label \leftarrow term$
21: $\quad\quad\quad\quad tag.conf \leftarrow 1$
22: $\quad\quad\quad\quad tags_i \leftarrow tags_i \cup \{tag\}$
23: $\quad\quad\quad$ **end if**
24: $\quad\quad$ **end if**
25: $\quad$ **end for**
26: **end for**
27: **return** $T_a = \{tags_1, \ldots, tags_k\}$

Figure 4: Algorithm *VisualExtraction*

4.1.2. Feature extraction and selection

For each image $p_i \in P$, the algorithm queries a visual analysis tool to get its set of tags, as well as a textual description of the picture's content (line 8). Each tag is associated to a confidence value, ranging from 0 to 1, which represents the

probability of that tag to actually be representative of p_i's content according to the computer vision algorithm. For each tag, the algorithm queries a POS tagger to get its lexicographic category (object, plant, animal, etc.); those tags that do not refer to categories of interest (enumerated in Section 3.2.3) are removed from the set of tags (lines 9 to 13), because they do not represent useful features to characterize the activity execution. Then, the POS engine is queried to extract additional terms from the picture description (line 14). Once again, terms not referring to categories of interest are discarded. For each remaining term, the algorithm checks whether it appears as the label of any tag of the image. If so, the confidence of that tag for the image is set to 1. Otherwise, a new tag with that label is created for that picture, and its confidence value is set to 1 (lines 15 to 25). Tags extracted from descriptions are given weight 1 because descriptions are assumed to be reasonably accurate. Finally, the algorithm returns the set T_a of weighted tags sets $\{tags_1, \ldots, tags_k\}$ for activity a.

4.2. Computing activity models

The algorithm *ModelExtraction* implements the module for ACTIVITY MODEL CONSTRUCTION shown in Fig. 3. The algorithm pseudo-code is shown in Fig. 5. It takes as input the set of considered activity labels $a_1, \ldots, a_n$ and the number k of images per activity. The output is the set of conditional probabilities $p(a_j|t_i)$, for each activity a_j and tag t_i. At first, for each a_j, the algorithm executes *VisualExtraction*(a_j, k) to get the set $T_{a_j} = \{tags_1, \ldots, tags_k\}$ of weighted tags sets associated to a_j (line 2). For each *tag* of each tags set $tags_i$, the algorithm assigns the tag's confidence $tag.conf$ to $p(tag.label, a_j, i)$. The latter is the probability of observing the object corresponding to the tag's label in the i^{th} image of a_j (lines 3 to 7). Then, for each activity a_j and each tag t_i, the algorithm computes the conditional probability $p(a_j|t_i)$ according to the following formula:

$$p(a_j|t_i) = \frac{\sum_l p(t_i, a_j, l)}{\sum_{m,l} p(t_i, a_m, l)} \cdot$$

Finally, the algorithm returns the conditional probability distribution $P(A|T)$,

Algorithm ***ModelExtraction***$(a_1, \ldots, a_n, k)$:
Input: Activity labels $a_1, \ldots, a_n$; number k of pictures per activity
Output: Conditional probabilities $p(a_j|t_i)$

for each activity label a_j **do**
  $T_{a_j} \leftarrow VisualExtraction(a_j, k)$
  for each $tags_i \in T_{a_j}$ **do**
    for each $tag \in tags_i$ **do**
      $p(tag.label, a_j, i) = tag.conf$
    end for
  end for
end for
for each activity label a_j **do**
  for each tag t_i **do**
    $$p(a_j|t_i) \leftarrow \frac{\sum_l p(t_i, a_j, l)}{\sum_{m,l} p(t_i, a_m, l)}$$
  end for
end for
return $P(A|T)$

Figure 5: Algorithm *ModelExtraction*

where A is the set of activities and T is the set of tags. As explained before, this simple model can be applied to several state-of-the-art activity recognition methods. More sophisticated models could be obtained by considering additional features, such as temporal information or sound mined from video clips, but we leave this aspect to future work. Anyway, as shown in Section 6, the model computed by the *ModelExtraction* algorithm is sufficiently accurate to provide satisfactory recognition rates.

5. Unsupervised activity recognition

Formally, given a temporal sequence of activations of sensors $\langle s_1, s_2, \ldots, s_n \rangle$ occurred at timestamps $\langle \tau_1, \tau_2, \ldots, \tau_n \rangle$, respectively, the objective of activity recognition is to reconstruct the current activity at each τ_i. As explained in Section 3.2.5, we use the M function to translate the temporal sequence of

sensor activations in a temporal sequence of corresponding tags:

$$\langle t_1 = M(s_1), t_2 = M(s_2), \ldots, t_n = M(s_n) \rangle.$$

In the following of this section, we present two unsupervised methods for recognizing human activities based on the mined model, considering the sequence of tags derived from sensor activations acquired in a smart environment. The experimental evaluation of those methods is reported in Section 6.

5.1. Temporally smoothed algorithm (TempS)

The first activity recognition algorithm is rather simple: in order to guess the current activity, it considers the conditional probabilities $p(A|T)$ and a fixed-length sliding window of the n most recent observations $\langle s_j, s_{j-1}, \ldots, s_{j-n+1} \rangle$. As explained above, the algorithm transforms those observations into the corresponding temporal sequence of tags $\langle t_j, t_{j-1}, \ldots, t_{j-n+1} \rangle$. The contribution of the observations to the prediction is smoothed according to their temporal order: more recent observations contribute more than less recent ones.

Technically, for each timestamp τ_j ($j \geq n$) and for each activity $a \in A$, the algorithm computes the weight $w(a, \tau_j)$, which is the temporally smoothed product of the conditional probability of a being the current activity at τ_j, at $\tau_{j-1}, \ldots$, and at τ_{j-n+1}. Formally:

$$w(a, \tau_j) = \prod_{k=j-n+1 \ldots j} p(a|t_j) \cdot c^{j-k},$$

where $c \in (0, 1]$ is the temporal smoothing factor, used to give more relative weight to the recent events. Given the weights computed for each activity $a \in A$ at τ_j, the predicted current activities $pred\_act(\tau_j)$ at τ_j are the ones that maximize the weight:

$$pred\_act(\tau_j) = \arg\max_{a \in A} w(a, \tau_j).$$

Normally, the cardinality of $pred\_act(\tau_j)$ is 1; hence, the predicted activity is unique. However, it is possible that more activities achieve the maximum value for $w(a, \tau_j)$. In this case, since we assume that the current activity is unique (even though multiple activities can be executed in an interleaved fashion), the algorithm predicts one activity at random from $pred\_act(\tau_j)$. It is also possible that all activities achieve score 0 for $w(a, \tau_j)$; this case happens when all recent sensor events do not correspond to tags in the model. In this case, the algorithm predicts one activity at random.

5.2. Markov Logic Networks algorithm (MarLoN)

Markov Logic Networks (MLN) [52] is a probabilistic logic that was successfully applied to sensor-based AR, due to its natural support for reasoning with uncertain information [53, 37]. Formally, a MLN $\mathcal{M}$ is a finite set of pairs $(F_i, w_i), 1 \leq i \leq n$, where each F_i is an axiom in function-free first-order logic, and $w_i \in \mathbb{R}$ is a weight representing the confidence in F_i's truth. Together with a finite set of constants $C = \{c_1, ..., c_n\}$ it defines the *ground* MLN $\mathcal{M}_C$. This comprises one binary variable for each grounding of F_i with weight w_i. A MLN defines a log-linear probability distribution over Herbrand interpretations (possible worlds):

$$P(\mathbf{x}) = \frac{1}{Z} \exp\left(\sum_i w_i n_i(\mathbf{x})\right)$$

where $n_i(\mathbf{x})$ is the number of satisfied groundings of F_i in the possible world $\mathbf{x}$ and Z is a normalization constant. Maximum a posteriori (MAP) inference is the task of finding the most probable world given some observations. Given the observed variables $E = e$, MAP finds an assignment of all hidden variables $X = x$ such that:

$$\mathbf{I} = \arg\max_x \ P(X = x \mid E = e),$$

where $\mathbf{I}$ is the assignment of x which leads P to be maximal. We have used the RockIt[2] MLN framework to formulate the knowledge base. In Fig. 6 we

[2] `http://executor.informatik.uni-mannheim.de/systems/rockit/`

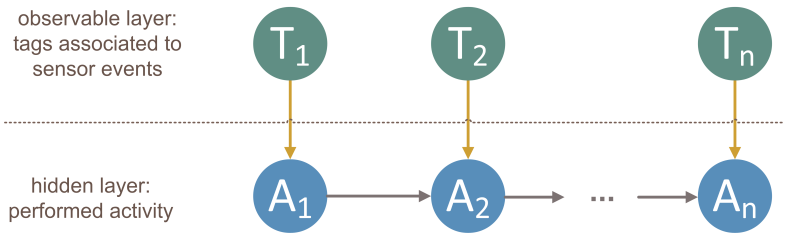


Figure 6: Formulation of the AR problem as an MLN. T_i is the tag derived from the sensor event observed at τ_i. A_i is the (hidden) current activity at τ_i. A_i depends on both T_i and on the previous activity A_{i-1}.

illustrate the MLN model, which is composed of:

- a set of *hidden states* $\{A_1, \ldots, A_n\}$: axioms representing the activities executed at $\{\tau_1, \ldots, \tau_n\}$, respectively;
- a set of *observations* $\{T_1, \ldots, T_n\}$: axioms representing the temporal sequence of tags obtained through the application of the M function (Section 3.2.5) to the temporal sequence of sensor events;
- *action probabilities*: axioms representing the probability that the current activity is a when the current tag is t;
- *transition probabilities*: axioms representing the probability that the current activity is a given that the previous activity was a'.

The MLN components are represented using the language of the RockIt framework, as shown in Figure 7.

As mentioned before, in MLN the probability of an axiom to be true is expressed through a weight w, as in the last line of the example above. Hence, we transform probability values into weights by applying the well-known *logit* function:

$$logit(p) = ln\left(\frac{p}{1-p}\right) \ , \ p \in (0;1).$$

We also declared the following axioms to state that there is exactly one activity and one tag at a time:

```
// The tag t corresponds to the sensor event
// occurred at time τ (second parameter).
Observe(t, int_)
// The unknown activity a is executed at τ
// (second parameter).
State(a, int_)
// Probability (third parameter) that the
// activity is a given the tag t.
PriorProb(a, t, float_)
// Transition probabilities (w = logit(p),
// where p is the probability value).
w !State(a1, τi) v State(a2, τi+1)
```

Figure 7: Main definitions of the MLN model using the RockIt framework

```
|a| state(a, τ) = 1
|t| observe(t, τ) = 1
```

In order to recognize activities, we instantiate the OBSERVE axioms with the temporal list of tags. Probabilistic PRIORPROB axioms are instantiated based on the $P(A|T)$ probability distribution of our inferred model. Transition probabilities are chosen based on common sense, or based on statistics. Finally, we use the RockIt reasoner to execute MAP inference for computing the most probable grounding of STATE axioms, which provide us with the most probable current activity performed at each τ. Note that, in the definition of our MLN model, we assume that exactly one activity is executed at a time. Hence, when multiple activities have the highest probability at tau, the reasoner predicts one activity at random among the most probable ones.

6. Experimental evaluation

We have performed extensive experiments with two real-world datasets of ADLs executed in different smart homes, using our methodology and the activity recognition algorithms presented in Section 4. Since our methods are unsupervised, we made a comparison with another unsupervised technique. We performed the comparison using an existing Web-based activity mining technique, which relies on Web page search and lexical analysis [49].

6.1. Datasets

The datasets were acquired in two smart homes instrumented with several kinds of sensors, and include the execution of variegate activities of daily living.

6.1.1. CASAS dataset

The first dataset that we used is the well-known dataset acquired at Washington State University by Cook et al. within the CASAS project [54, 55]. This dataset includes both interleaved and sequential ADLs executed in a smart-home by 21 subjects[3]. Sequential activities are pre-segmented, while interleaved activities are not. Since, in the real world, people perform activities in an interleaved fashion, we limited our attention to the recognition of interleaved ADLs. Sensors collected data about movement, presence in specific home locations, temperature, use of water, interaction with objects, doors, phone; 70 sensors were used in total. For the sake of this work, we considered only 24 out of 70 sensors; indeed, the other sensors (mostly presence sensors) were not associated to actions, or to the use of objects or furniture. Used sensors are reported in Table 1. The dataset considers eight activities, whose labels are reported in Table 2.

The order and time taken to perform the activities were up to the subject. Activities were executed naturalistically by a single subject at a time. In the dataset, each sensor activation (e.g., "fridge opened", "cup moved") is labeled

[3] `http://ailab.wsu.edu/casas/datasets/adlinterweave.zip`

Table 1: CASAS sensors

id	sensor	id	sensor
$I04$	medicine box	$I01$	pan
$I06$	cabinet	$I02$	stove
$I03$	remote control	$D12$	wardrobe
$I05$	television	$P01$	cellphone
$AD1$-B	sink	$M19$	vacuum
$AD1$-C	sink	$D08$	freezer
$D11$	watering can	$M02$	sofa
$I08$	scissors	$M24$	medicine cabinet
$I09$	stationery	$M23$	outfit cabinet
$D07$	kitchen door	$M04$	vacuum
$D09$	fridge	$M05$	duster
$D10$	cutlery	$M13$	envelope

Table 2: CASAS activities

activity	label
a_1	fill medicine cabinet
a_2	watch tv
a_3	watering plants
a_4	answering the phone
a_5	diy birthday card
a_6	prepare soup
a_7	hoovering
a_8	choosing outfit

with the timestamp of the event, and with the current activity executed at that timestamp. Given the temporal sequence of sensor activations, the goal of the activity recognition system is to reconstruct the current activity at each activation.

6.1.2. IELAB dataset

The second dataset was collected in the IELAB facility of the University of Auckland [56]. The smart environment includes lounge, toilet, kitchen, and dining room. The environment is instrumented with several sensors, including

presence sensors and door sensors to track movement and actions within the home. Other sensors are attached to objects and furniture, including chairs, pots, plates, burner. As with the CASAS dataset, we disregarded data acquired from presence sensors. The dataset[4] considers 7 activities of daily living executed by 20 subjects, which perform the activities in an arbitrary order. The list of sensors and the activities are reported in Table 3.

Table 3: IELAB sensors and activities

id	**sensor**
$M03$	toilet
$I02$	television
$I03$	sofa
$I04$	pot
$I05$	plate
$D02$	kitchen cabinet
$W01$	kitchen sink
$W02$	bathroom sink
$C01$	toilet seat
$C03$	dining chair
$BURNER$	burner
$FLUSH$	toilet flush

activity	**label**
a_1	cooking meal
a_2	having meal
a_3	cleaning dishes
a_4	toileting 1
a_5	toileting 2
a_6	napping
a_7	watching TV

6.2. Experimental setup

We have implemented all the algorithms used in our experimental evaluation in Python. In order to provide the possibility to replicate the experiments, we have published our code online[5].

We evaluated the prediction's quality in terms of the standard measures of precision, recall and F_1 score; the latter is the harmonic mean of precision and recall. While precision, recall and F_1 score were computed for each individual activity, we used the micro-F_1 score [57] to measure the global accuracy of the

[4] `http://halim.readismed.com/datasets/`

[5] `http://sites.unica.it/domusafe/arcode/`

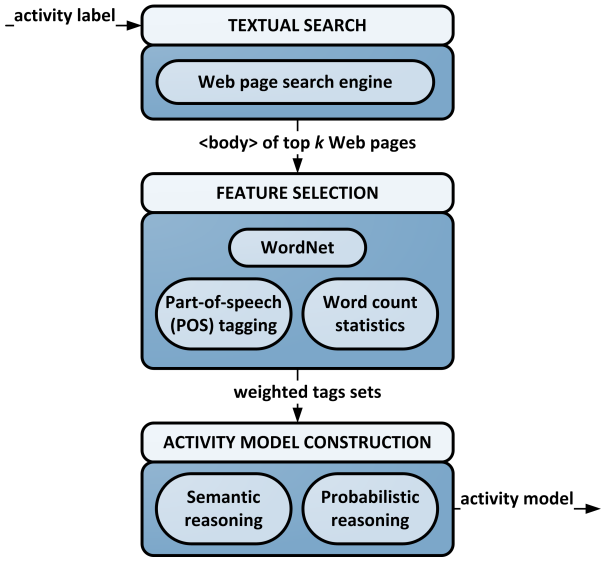


Figure 8: Text-based activity mining method (*TextAM*).

activity recognition technique on the whole set of activities. Indeed, in our experiments, each sample (i.e., activity instance) belongs to one and only one class, and the algorithms assign each sample to exactly one class. Hence, in our case the micro-F_1 score corresponds to the ratio of correctly classified instances over the total number of instances. Since the methods are unsupervised, we did not use cross-validation.

6.2.1. Implementation of Text-based activity mining

In order to compare our approach with a Web-based activity mining one, we have implemented the method based on Web page search and lexical analysis illustrated in Figure 8 and explained below. In the rest of the paper, we refer to this method as *TextAM*. That method is analogous to the ones presented in [40] and [41].

Essentially, given an activity label, TextAM queries a search engine to find a list of Web pages related to that activity; then, it applies text processing and lexical analysis on the text of the Web page to compute the relatedness between activities and terms. Relatedness is expressed through a conditional probability

distribution $P(A|T)$, where A is the set of activities and T is the set of terms (or *tags*). In particular, for each activity, TextAM downloads the first k Web pages found by Google Search. The value k is fixed to 15, because we experimentally found that in general it provides the highest accuracy with the considered activities. For each Web page, TextAM considers the content of the ⟨body⟩ element, removing stopwords. For each word in the body, the method queries the Natural Language Toolkit APIs of WordNet [58], a well-known lexical database of English nouns, verbs, adjectives and adverbs, to retrieve its POS. Only those words that belong to lexicographic categories of interest (enumerated in Section 3.2.3) are considered. For each of those words and its synonyms, TextAM computes the number of occurrences and the weight. The weight is computed as the probability of that word to actually be a noun (# of senses of that word that are noun / total # of senses of that word). Finally, TextAM computes the conditional probability distribution $P(A|T)$ using the same method used in Algorithm *ModelExtraction* (Fig. 5), but considering weighted tag sets extracted from Web page text instead of pictures. Note that, differently from the technique presented in [40], TextAM computes the distribution of $p(activity|tag)$, since in the experiments we use discriminative activity recognition algorithms.

6.2.2. Implementation of picture-based activity mining

At first, our Python software queries the Google Image Search APIs to retrieve the top k pictures for each activity label, excluding cliparts or drawings, and searching for photos of medium dimension. Also in our case, k is experimentally set to 15. However, we experimentally found that slight modifications to the k value produce little change to the overall recognition rates of our method.

In order to analyze the retrieved images for extracting tags and descriptions (Algorithm VisualExtraction, shown in Fig. 4), our software exploits two Computer Vision services: Microsoft Cognitive Services[6] and Clarifai Image &

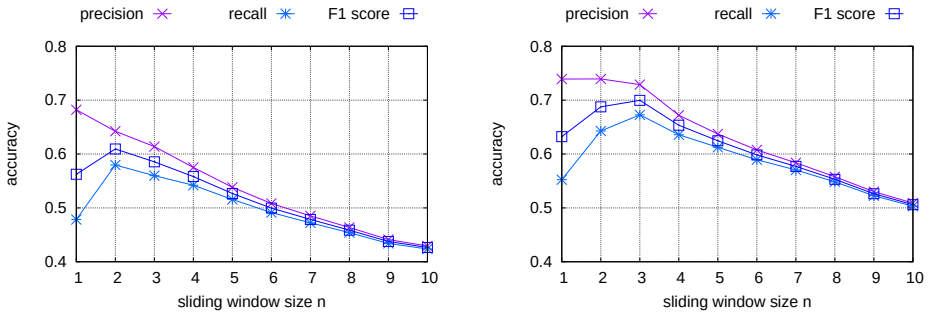
[6] `https://www.microsoft.com/cognitive-services/en-us/computer-vision-api`

Video Recognition[7]. Those services take a picture as input, and return a JSON document storing several information about the picture, including tags with associated confidence, and a textual description. Thanks to recent advances in artificial intelligence methods, new graphics-processing-unit (GPU)-based hardware, and increasing volume of labeled data, those Computer Vision services provide impressively accurate results [59]. Several ad-hoc models are available for recognizing particular classes of objects, such as food, logo, animals; in our implementation, we use a generic model. Our software takes advantage of the MongoDB NoSQL DBMS for storing the data retrieved from the remote Computer Vision APIs. For each term appearing in tags and descriptions, the program queries the Natural Language Toolkit APIs of WordNet to retrieve the lexicographic category of terms extracted from images, in order to select only those terms belonging to the categories of interest. In our current system, synsets are automatically extracted from the Natural Language Toolkit of WordNet, and it is possible that in some cases they do not fully correspond to the terminology of each computer vision service. For the sake of this work, we do not have considered this problem. However, the problem could be addressed by applying a technique for word sense alignment [60]. Finally, for every activity a and tag t, the program computes the conditional probability $P(a|t)$ according to Algorithm ModelExtraction (shown in Fig. 5).

6.2.3. Implementation of video-based activity mining

We also conducted an experiment using short videos illustrating the execution of the activities of interest. We manually chose two videos for each activity from YouTube, and queried the Video Recognition APIs of Clarifai to retrieve related tags. The mechanism to select and use tags for computing the activity models is the same used for pictures.

[7] `https://www.clarifai.com/developer`



(a) TextAM. (b) PictAM(MS).

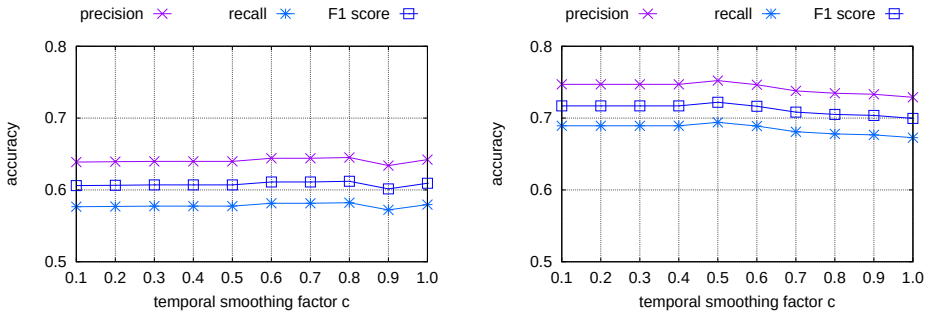
Figure 9: TempS algorithm and CASAS dataset: accuracy obtained varying the size n of the sliding window. No temporal smoothing ($c = 1$).

6.3. Results

In the following, we present the results of our experiments on the two datasets. We compared the text-based activity mining method (TextAM) with our methods to extract activity models from pictures and videos. As explained before, we used different computer vision APIs to identify features of interest. We denote by PictAM(MS) the method using pictures and Microsoft Cognitive Services; PictAM(CF Image) is the one using pictures and Clarifai Image Recognition; PictAM(CF Video) uses videos and Clarifai Video Recognition.

6.3.1. TempS algorithm and CASAS dataset

As explained in Section 5.1, the TempS algorithm has two parameters: the temporal smoothing factor $c \in (0, 1]$, and the length $n \geq 1$ of the sliding window of sensor events. Hence, the first experiment was aimed at finding the most effective values for c and n. Initially, we set the temporal smoothing factor c to 1; i.e., no temporal smoothing was applied. We varied the size n of the sliding window from 1 to 10. Results for TextAM and PictAM(MS) are shown in Fig. 9. Results using PictAM(CF Image) and PictAM(CF Video) have a similar trend; they are omitted for lack of space. The best results are obtained using relatively small values of n. With no temporal smoothing, the PictAM(MS) method significantly outperformed TextAM (micro-$F_1 = 0.6996$ vs micro-$F_1 = 0.6093$).



(a) TextAM. The sliding window size is $n = 2$.

(b) PictAM(MS). The sliding window size is $n = 3$.

Figure 10: TempS algorithm and CASAS dataset: accuracy obtained varying the temporal smoothing factor c. The size of the sliding window is fixed.

Then, for each method, we fixed the size n of the sliding window, and tried different values of temporal smoothing factor c from 0.1 to 1. Fig. 10 reports the results. With TextAM, we obtained the best results fixing $n = 2$. The highest F_1 score 0.612 was obtained with $c = 0.8$. With our image-based method PictAM(MS), the highest F_1 score 0.722 was obtained with $n = 3$ and $c = 0.5$. Once again, we omitted the results of PictAM(CF Image) and PictAM(CF Video) for lack of space, but they showed a similar trend. In general, with all methods, the influence of temporal smoothing was limited.

Table 4 shows the F_1 score achieved by the different methods to extract activity models, applying the TempS algorithm on the CASAS dataset. Overall, the Pict(AM) method achieved the best results (micro-$F_1 = 0.722$). The accuracy of the PictAM(CF Image) method was significantly lower (micro-$F_1 = 0.6431$). Since the only difference between the two methods is the type of computer vision API used, this results indicates that the accuracy of the algorithm used to identify tags in pictures is an essential point for the effectiveness of our approach. The PictAM(CF Video) method achieves a micro-F_1 score of 0.6943, which is close to the best one achieved by PictAM(MS). Even though the experiments with videos were preliminary, this result indicates that extraction of activity models from videos has a good potential. The TextAM method achieved the

Table 4: TempS algorithm and CASAS dataset: F_1 score of activity mining methods for the different activities. In the last row we report the micro-F_1 score over all activities.

Activity	**TextAM**	**PictAM (MS)**	**PictAM (CF Image)**	**PictAM (CF Video)**
a_1	**0.8184**	0.7917	0.6065	0.7394
a_2	0.5370	**0.8691**	0.8140	0.8131
a_3	**0.5336**	0.4283	0.0566	0.5017
a_4	0.3524	0.3162	0.0615	**0.5367**
a_5	**0.8909**	0.8788	0.7657	0.8183
a_6	0.7094	**0.7305**	0.6063	0.6522
a_7	0.4654	**0.6405**	0.6030	0.5699
a_8	**0.7871**	0.5876	0.7330	0.7417
micro-F_1	0.6120	**0.7220**	0.6431	0.6943

lowest micro-F_1 score (0.612).

6.3.2. MarLoN algorithm and CASAS dataset

Table 5: MarLoN algorithm and CASAS dataset: F_1 score of activity mining methods for the different activities. In the last row we report the micro-F_1 score over all activities.

Activity	**TextAM**	**PictAM (MS)**	**PictAM (CF Image)**	**PictAM (CF Video)**
a_1	0.3089	0.4337	0.4051	**0.6547**
a_2	0.7841	0.8026	0.8161	**0.8247**
a_3	0.4146	0.3205	0.0759	**0.4431**
a_4	0.3724	**0.5749**	0.4744	0.4406
a_5	**0.8530**	0.8275	0.7549	0.7974
a_6	0.6000	**0.6935**	0.5559	0.5132
a_7	**0.5485**	0.4762	0.5000	0.5218
a_8	0.3218	**0.7634**	0.7117	0.7522
micro-F_1	0.5814	**0.6502**	0.6092	0.6483

We executed the same experiments on the CASAS dataset, using the MarLoN algorithm instead of the TempS one. Results are shown in Table 5. The outcome of experiments was in line with the results obtained using the TempS

algorithm. Indeed, the PictAM(MS) method achieved the best results, closely followed by PictAM(CF Video). The micro-F_1 score of PictAM(CF Image) was significantly lower. The TextAM method achieved the lowest micro-F_1 score.

In general, we observe that for most activities the MarLon algorithm achieved lower accuracy than the TempS one on the CASAS dataset. The only activity for which MarLoN achieved higher accuracy than TempS was a_4 (answering the phone). A possible explanation for the low accuracy of MarLoN is that the information about probable activity sequences (e.g., "cooking is usually followed by having meal") cannot be exploited with the CASAS dataset, because activities were carried out by subjects in random order. Hence, in the MarLoN model we set uniform transition probabilities for activity changes; we set a slightly higher probability value (0.2) for transitions between the same activity. However, in a real-world situation, the exploitation of non-uniform transition probabilities should increase the accuracy of the activity recognition algorithm.

6.3.3. TempS algorithm and IELAB dataset

Table 6: TempS algorithm and IELAB dataset: F_1 score of activity mining methods for the different activities. In the last row we report the micro-F_1 score over all activities.

Activity	**TextAM**	**PictAM (MS)**	**PictAM (CF Image)**	**PictAM (CF Video)**
a_1	0.2838	0.6642	0.7333	**0.7564**
a_2	0.8039	**0.8057**	**0.8057**	0.8042
a_3	**0.5098**	0.3265	0.0801	0.2794
a_4	**0.5592**	0.4602	0.4602	0.1389
a_5	0.4800	0.5560	0.5560	**0.5625**
a_6	**0.9058**	0.8966	0.8966	0.9016
a_7	**0.4279**	0.4033	0.4033	0.4164
micro-F_1	0.6455	0.6724	**0.6803**	0.6654

In this set of experiments, we used the TempS algorithm for recognizing the activities carried out within the IELAB dataset. Table 6 shows the outcome of these experiments.

Since the IELAB dataset considered different activities with respect to the CASAS dataset, the set of pictures retrieved by the visual search engine was different from the one retrieved for CASAS. In these experiments, the highest micro-F_1 score was achieved using the PictAM(CF Image) method, while the most effective method for the CASAS dataset was PictAM(MS). An explanation for this result is that the Clarifai Image computer vision tools were more effective in recognizing relevant objects within those pictures than Microsoft Cognitive Services ones, while Microsoft tools were more effective than Clarifai ones on the pictures of the CASAS dataset. These results confirm that the effectiveness of our methods strongly depends on the accuracy of the underlying computer vision mechanism. This is a positive indication, since the recognition capabilities of computer vision tools are continuously improving thanks to the use of deep learning and to the availability of increasingly large datasets of pictures and videos.

6.3.4. MarLoN algorithm and IELAB dataset

Table 7: MarLoN algorithm and IELAB dataset: F_1 score of activity mining methods for the different activities. In the last row we report the micro-F_1 score over all activities.

Activity	**TextAM**	**PictAM (MS)**	**PictAM (CF Image)**	**PictAM (CF Video)**
a_1	0.4147	0.3631	0.3183	**0.7564**
a_2	0.8039	0.7695	0.7127	**0.8042**
a_3	**0.3437**	0.3131	0.2287	0.2794
a_4	0.5140	0.4917	**0.5513**	0.3727
a_5	0.1728	0.4471	**0.6538**	0.4564
a_6	0.7980	0.8867	**0.9038**	0.9016
a_7	**0.4279**	0.3794	0.4224	0.4164
micro-F_1	0.5748	0.6192	0.6492	**0.6688**

In the last set of experiments, we used the MarLoN algorithm for recognizing the activities carried out within the IELAB dataset. Results are shown in Table 7.

Table 8: Micro-F_1 score of activity mining methods considering the both datasets as a whole.

Activity mining method	**TextAM**	**PictAM (MS)**	**PictAM (CF Image)**	**PictAM (CF Video)**
TempS	0.6235	**0.7049**	0.6559	0.6843
MarLoN	0.5791	0.6395	0.6230	**0.6554**

In general, the results achieved by MarLoN were less accurate than the ones achieved by TempS. However, the results achieved by PictAM(CF Video) using MarLoN were slightly better than the ones achieved by the same method using TempS. In these experiments, the highest accuracy was indeed achieved by PictAM(CF Video). This result confirms the potential of mining activity models from videos for sensor-based activity recognition. The accuracy achieved by the other methods was significantly lower. Also in these experiments, TextAM achieved the lowest micro-F_1 score.

6.3.5. Overall results

Table 8 provides an overview of results, showing the micro-F_1 score of activity mining methods considering the two datasets as a whole. With TempS, the most effective method was PictAM(MS), which achieved a score of 0.7049. The performance of PictAM(CF Video) was also good, while the one of PictAM(CF Image) was lower. With this activity recognition method, the score of TextAM was the lowest.

With MarLoN, the highest accuracy was achieved by PictAM(CF Video), while PictAM(MS) and PictAM(CF Image) achieved a slightly lower score. Even with this activity recognition algorithm, the score of TextAM was the lowest.

7. Discussion

Overall, experimental results indicate that visual media can be effectively used to mine activity models, even when sporadic activities are considered.

Moreover, activity models extracted from pictures and videos outperform models extracted from text in terms of activity recognition accuracy.

Inspecting individual activities, we found some activities (especially, "watering plants", "answering the phone", and "hoovering") hard to recognize, mainly due to the lack of sufficient information about objects usage in the dataset for them. In particular, the recognition rate of "watering plants" was low. Indeed, as shown by the confusion matrix in Table 9, that activity was frequently confused with "preparing soup". Those errors happened because both activities involve the use of the same or similar tools (e.g., sinks and water containers). We claim that this is an intrinsic limit of object based-activity recognition; it is not due to the specific method used to define object-based activity models. This problem can be mitigated by considering additional smart objects that can better characterize the activities.

Table 9: TempS algorithm and CASAS dataset ($n = 3$, $c = 0.5$): confusion matrix.

classified as →	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8
a_1	**98**	9	6	2	3	12	3	1
a_2	0	**409**	5	0	8	3	17	0
a_3	0	8	**75**	0	0	68	25	1
a_4	4	9	3	**10**	1	7	7	1
a_5	0	5	0	0	**106**	1	0	0
a_6	10	11	13	9	5	**289**	18	5
a_7	0	32	67	0	3	2	**165**	0
a_8	2	16	4	0	3	47	11	**65**

A comparison with other activity recognition techniques using the CASAS dataset indicates that our results are positive. The Hidden Markov Model method used in [54] on the CASAS dataset achieved an average recognition accuracy of 0.700; our method achieved a higher score, having the advantage of being unsupervised (i.e., no training set must be acquired). The unsupervised method presented in [51], based on a hybrid combination of ontological and probabilistic reasoning, achieved higher accuracy on the CASAS dataset; i.e., average accuracy of 0.781. However, that method adopts an advanced activity recognition technique, while the two methods used in this paper were rather

simplistic. Moreover, that hybrid method strongly relies on manual activity modeling through ontology engineering, which is an expensive task. On the contrary, our approach has the advantage of being fully automatic, apart from the straightforward ontological definition of the sensor infrastructure.

Despite our ArOnt ontology having a rather simple structure, the effort of filling the ontology with sensor instances is not negligible. We believe that the effort can be reduced by adopting user-friendly interfaces to specify the semantics and positioning of sensors in the smart home, possibly with the help of plug-and-play mechanisms. Moreover, the effort can be reduced by re-using and extending existing ontologies. Indeed, while the set of sensors is environment-dependent, the correspondences among actions/objects and tags is generic and can be shared across different smart homes.

In a recent work, it was proposed to reduce the burden of ontology engineering while retaining the advantages of unsupervised activity recognition; that approach relies on a generic ontological model, interactively refined by users' feedback and active learning [61]. The accuracy of that method, applied without active learning, was 0.73 on the CASAS dataset. Our method using pictures achieved essentially the same accuracy, without the need of manually defining a generic ontology of activities and smart environments. At the time of writing, we cannot compare the effectiveness of our methods on the IELAB dataset, since the experimental setup used in the original paper [56] was different (different set of activities, different recognition task) and we could not find any other publication using the IELAB dataset.

8. Conclusion and future work

In this paper, we introduced a new approach to unsupervised activity mining, which relies on visual information. We presented a technique for extracting relevant activity images and videos from the Web, identifying key information through computer vision tools, and computing activity models. A detailed experimental comparison with related works shows the effectiveness of our ap-

proach.

This work can be extended in several directions. The technique to select relevant pictures and videos could be improved by analyzing the context in which they are published. While we used general-purpose computer vision APIs, object recognition could be improved adopting specific methods to recognize human-object interaction [62]. Moreover, temporal information could be extracted by mining activity data from videos in order to compute more accurate activity models.

Currently, our method mainly relies on sensors attached to objects and furniture. Our method could be extended in order to take advantage of indoor positioning systems, which are more and more available in smart homes. To this aim, we could exploit image-based place recognition APIs, and match the typical location of activities with the user's current position.

Acknowledgments

The authors would like to thank the anonymous reviewers for their insightful comments and suggestions.

This work was partially supported by the "DomuSafe" project, funded by Sardinia regional government (CRP 69, L.R. 7 agosto 2007, n.7), and by the EU's Marie Curie training network PhilHumans - Personal Health Interfaces Leveraging HUman-MAchine Natural interactionS (grant number 812882).

References

[1] N. Davies, D. P. Siewiorek, R. Sukthankar, Activity-based computing, IEEE Pervasive Computing 7 (2) (2008) 20–21.

[2] P. Rashidi, A. Mihailidis, A survey on ambient-assisted living tools for older adults, IEEE J. Biomedical and Health Informatics 17 (3) (2013) 579–590.

[3] T. Zhao, H. Ni, X. Zhou, L. Qiang, D. Zhang, Z. Yu, Detecting abnormal patterns of daily activities for the elderly living alone, in: Health Informa-

tion Science, Vol. 8423 of Lecture Notes in Computer Science, Springer, 2014, pp. 95–108.

[4] L. Bao, S. S. Intille, Activity recognition from user-annotated acceleration data, in: Proceedings of Pervasive Conference, Springer, 2004, pp. 1–17.

[5] A. Bulling, U. Blanke, B. Schiele, A tutorial on human activity recognition using body-worn inertial sensors, ACM Computing Surveys (CSUR) 46 (3) (2014) 33:1–33:33.

[6] M. M. Hassan, M. Z. Uddin, A. Mohamed, A. Almogren, A robust human activity recognition system using smartphone sensors and deep learning, Future Generation Comp. Syst. 81 (2018) 307–313.

[7] L. Sun, D. Zhang, B. Li, B. Guo, S. Li, Activity recognition on an accelerometer embedded mobile phone with varying positions and orientations, in: Ubiquitous Intelligence and Computing, Vol. 6406 of Lecture Notes in Computer Science, Springer, 2010, pp. 548–562.

[8] Y. Zheng, Q. Li, Y. Chen, X. Xie, W. Ma, Understanding mobility based on GPS data, in: UbiComp, Vol. 344 of ACM International Conference Proceeding Series, ACM, 2008, pp. 312–321.

[9] V. W. Zheng, Y. Zheng, Q. Yang, Joint learning user's activities and profiles from GPS data, in: Proc. of the 2009 International Workshop on Location Based Social Networks, ACM, 2009, pp. 17–20.

[10] A. Calatroni, D. Roggen, G. Tröster, Collection and curation of a large reference dataset for activity recognition, in: Proceedings of Systems, Man, and Cybernetics Conference, IEEE, 2011, pp. 30–35.

[11] A. Reiss, D. Stricker, Creating and benchmarking a new dataset for physical activity monitoring, in: Proceedings of the PETRA Conference, ACM, 2012, pp. 1–8.

[12] P. Rashidi, D. J. Cook, COM: A method for mining and monitoring human activity patterns in home-based health monitoring systems, ACM Transactions on Intelligent Systems and Technology 4 (4) (2013) 64:1–64:20.

[13] D. Riboni, C. Bettini, OWL 2 modeling and reasoning with complex human activities, Pervasive and Mobile Computing 7 (3) (2011) 379–395.

[14] I. Bae, An ontology-based approach to ADL recognition in smart homes, Future Generation Comp. Syst. 33 (2014) 32–41.

[15] G. Okeyo, L. Chen, H. Wang, Combining ontological and temporal formalisms for composite activity modelling and recognition in smart homes, Future Generation Comp. Syst. 39 (2014) 29–43.

[16] D. Riboni, M. Murtas, Web mining & computer vision: New partners for object-based activity recognition, in: Proceedings of the 26th IEEE International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), IEEE Computer Society, 2017, pp. 158–163.

[17] P. K. Turaga, R. Chellappa, V. S. Subrahmanian, O. Udrea, Machine recognition of human activities: A survey, IEEE Trans. Circuits Syst. Video Techn. 18 (11) (2008) 1473–1488.

[18] S. Vishwakarma, A. Agrawal, A survey on activity recognition and behavior understanding in video surveillance, The Visual Computer 29 (10) (2013) 983–1009.

[19] L. Onofri, P. Soda, M. Pechenizkiy, G. Iannello, A survey on using domain and contextual knowledge for human activity recognition in video streams, Expert Syst. Appl. 63 (2016) 97–111.

[20] Q. Li, Z. Qiu, T. Yao, T. Mei, Y. Rui, J. Luo, Learning hierarchical video representation for action recognition, International Journal of Multimedia Information Retrieval 6 (1) (2017) 85–98.

[21] D. Christin, A. Reinhardt, S. S. Kanhere, M. Hollick, A survey on privacy in mobile participatory sensing applications, Journal of Systems and Software 84 (11) (2011) 1928–1946.

[22] J. Ye, S. Dobson, S. McKeever, Situation identification techniques in pervasive computing: A review, Pervasive and Mobile Computing 8 (1) (2012) 36–66.

[23] A. R. Golding, N. Lesh, Indoor navigation using a diverse set of cheap, wearable sensors, in: Proceedings of the Third International Symposium on Wearable Computers (ISWC'99), IEEE Computer Society, 1999, pp. 29–36.

[24] N. Kern, B. Schiele, A. Schmidt, Multi-sensor activity context detection for wearable computing, in: Proceedings of the First European Symposium on Ambient Intelligence (EUSAI 2003), Vol. 2875 of Lecture Notes in Computer Science, Springer, 2003, pp. 220–232.

[25] J. L. Reyes-Ortiz, L. Oneto, A. Samà, X. Parra, D. Anguita, Transition-aware human activity recognition using smartphones, Neurocomputing 171 (2016) 754–767.

[26] L. Yao, Q. Z. Sheng, W. Ruan, T. Gu, X. Li, N. Falkner, Z. Yang, Rf-care: Device-free posture recognition for elderly people using a passive RFID tag array, ICST Trans. Ambient Systems 2 (6) (2015) e2.

[27] Z. Huang, K. Lin, B. Tsai, S. Yan, C. Shih, Building edge intelligence for online activity recognition in service-oriented iot systems, Future Generation Comp. Syst. 87 (2018) 557–567.

[28] J. Lester, T. Choudhury, N. Kern, G. Borriello, B. Hannaford, A hybrid discriminative/generative approach for modeling human activities, in: Proceedings of the 19th international joint conference on Artificial intelligence, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2005, pp. 766–772.

[29] T. Gu, Z. Wu, X. Tao, H. K. Pung, J. Lu, epSICAR: An emerging patterns based approach to sequential, interleaved and concurrent activity recognition, in: Proceedings of the Seventh Annual IEEE International Conference on Pervasive Computing and Communications (PerCom), IEEE Computer Society, Washington, D.C., 2009, pp. 1–9.

[30] X. Hong, C. D. Nugent, M. D. Mulvenna, S. I. McClean, B. W. Scotney, S. Devlin, Evidential fusion of sensor data for activity recognition in smart homes, Pervasive and Mobile Computing 5 (3) (2009) 236–252.

[31] M. Hardegger, D. Roggen, A. Calatroni, G. Tröster, S-SMART: A unified bayesian framework for simultaneous semantic mapping, activity recognition, and tracking, ACM Transactions on Intelligent Systems and Technology 7 (3) (2016) 34:1–34:28.

[32] M. Rawashdeh, M. G. Al Zamil, S. Samarah, M. S. Hossain, G. Muhammad, A knowledge-driven approach for activity recognition in smart homes based on activity profiling, Future Generation Computer Systems, in Press.

[33] T. Gu, L. Wang, H. Chen, X. Tao, J. Lu, Recognizing multiuser activities using wireless body sensor networks, IEEE Trans. Mob. Comput. 10 (11) (2011) 1618–1631.

[34] L. Yao, F. Nie, Q. Z. Sheng, T. Gu, X. Li, S. Wang, Learning from less for better: semi-supervised activity recognition via shared structure discovery, in: UbiComp, ACM, 2016, pp. 13–24.

[35] L. Chen, C. Nugent, Ontology-based activity recognition in intelligent pervasive environments, International Journal of Web Information Systems 5 (4) (2009) 410–430.

[36] D. Riboni, L. Pareschi, L. Radaelli, C. Bettini, Is ontology-based activity recognition really effective?, in: Proceedings of the Ninth Annual IEEE International Conference on Pervasive Computing and Communications Workshops, IEEE, 2011, pp. 427–431.

[37] R. Helaoui, D. Riboni, H. Stuckenschmidt, A probabilistic ontological framework for the recognition of multilevel human activities, in: Proceedings of ACM UbiComp, ACM, 2013, pp. 345–354.

[38] D. Triboan, L. Chen, F. Chen, Z. Wang, A semantics-based approach to sensor data segmentation in real-time activity recognition, Future Generation Comp. Syst. 93 (2019) 224–236.

[39] M. Perkowitz, M. Philipose, K. P. Fishkin, D. J. Patterson, Mining models of human activities from the web, in: Proceedings of WWW Conference, ACM, 2004, pp. 573–582.

[40] D. Wyatt, M. Philipose, T. Choudhury, Unsupervised activity recognition using automatically mined common sense, in: Proceedings AAAI, AAAI Press / The MIT Press, 2005, pp. 21–27.

[41] E. M. Tapia, T. Choudhury, M. Philipose, Building reliable activity models using hierarchical shrinkage and mined ontology, in: Proceedings of Pervasive, Vol. 3968 of LNCS, Springer, 2006, pp. 17–32.

[42] W. Pentney, A. Popescu, S. Wang, H. A. Kautz, M. Philipose, Sensor-based understanding of daily life via large-scale use of common sense, in: Proceedings of AAAI, AAAI Press, 2006, pp. 906–912.

[43] R. Gupta, M. J. Kochenderfer, Common sense data acquisition for indoor mobile robots, in: Proceedings of AAAI, AAAI Press / The MIT Press, 2004, pp. 605–610.

[44] T. Gu, S. Chen, X. Tao, J. Lu, An unsupervised approach to activity recognition and segmentation based on object-use fingerprints, Data Knowl. Eng. 69 (6) (2010) 533–544.

[45] P. P. Palmes, H. K. Pung, T. Gu, W. Xue, S. Chen, Object relevance weight pattern mining for activity recognition and segmentation, Pervasive and Mobile Computing 6 (1) (2010) 43–57.

[46] I. K. Ihianle, U. Naeem, A. H. Tawil, M. A. Azam, Recognizing activities of daily living from patterns and extraction of web knowledge, in: UbiComp Adjunct Proceedings, ACM, 2016, pp. 1255–1262.

[47] T. Nakatani, R. Kuga, T. Maekawa, Preliminary investigation of object-based activity recognition using egocentric video based on web knowledge, in: Proc. of the 17th International Conference on Mobile and Ubiquitous Multimedia, ACM, 2018, pp. 375–381.

[48] B. C. Grau, I. Horrocks, B. Motik, B. Parsia, P. F. Patel-Schneider, U. Sattler, OWL 2: The next step for OWL, Journal of Web Semantics 6 (4) (2008) 309–322.

[49] D. Wyatt, M. Philipose, T. Choudhury, Unsupervised activity recognition using automatically mined common sense, in: Proceedings of the 20th National Conference on Artificial Intelligence, Vol. 1, AAAI Press, California, USA, 2005, pp. 21–27.

[50] S. Wang, W. Pentney, A.-M. Popescu, T. Choudhury, M. Philipose, Common sense based joint training of human activity recognizers, in: Proceedings of IJCAI 2007, 2007, pp. 2237–2242.

[51] D. Riboni, T. Sztyler, G. Civitarese, H. Stuckenschmidt, Unsupervised recognition of interleaved activities of daily living through ontological and probabilistic reasoning, in: Proceedings of ACM UbiComp, ACM, 2016, pp. 1–12.

[52] M. Richardson, P. Domingos, Markov logic networks, Machine learning 62 (1) (2006) 107–136.

[53] R. Helaoui, M. Niepert, H. Stuckenschmidt, Recognizing interleaved and concurrent activities using qualitative and quantitative temporal relationships, Pervasive Mob Comput 7 (6) (2011) 660 – 670.

[54] G. Singla, D. J. Cook, M. Schmitter-Edgecombe, Tracking activities in complex settings using smart environment technologies, Int. J. Biosci. Psychiatr. Technol. 1 (1) (2009) 25–35.

[55] D. J. Cook, A. S. Crandall, B. L. Thomas, N. C. Krishnan, CASAS: A smart home in a box, Computer 46 (7) (2013) 62–69.

[56] Z. S. M. H. M. Noor, K. I.-K. Wang, Enhancing ontological reasoning with uncertainty handling for activity recognition, in: Knowledge-Based Systems, vol. 114, 2016, pp. 47–60.

[57] Y. Yang, An evaluation of statistical approaches to text categorization, Inf. Retr. 1 (1-2) (1999) 69–90.

[58] G. A. Miller, Wordnet: A lexical database for english, Commun. ACM 38 (11) (1995) 39–41.

[59] J. Lemley, S. Bazrafkan, P. Corcoran, Deep learning for consumer devices and services: Pushing the limits for machine learning, artificial intelligence, and computer vision, IEEE Consumer Electronics Magazine 6 (2) (2017) 48–56.

[60] M. Matuschek, I. Gurevych, Dijkstra-wsa: A graph-based approach to word sense alignment, Trans Assoc Comput Linguist 1 (2013) 151–164.

[61] G. Civitarese, C. Bettini, T. Sztyler, D. Riboni, H. Stuckenschmidt, Nectar: Knowledge-based collaborative active learning for activity recognition, in: Proceedings of IEEE PerCom, IEEE Comp. Soc., 2018, pp. 1–12.

[62] B. Yao, F. Li, Recognizing human-object interactions in still images by modeling the mutual context of objects and human poses, IEEE Trans. Pattern Anal. Mach. Intell. 34 (9) (2012) 1691–1703.