INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 13(6), 988-1006, December 2018.

Indoor Localisation through Probabilistic Ontologies

I. Mocanu, G. Scarlat, L. Rusu, I. Pandelica, B. Cramariuc

Irina Mocanu*
Computer Science Department, University Politehnica of Bucharest
Romania, RO-060042 Bucharest, Splaiul Independentei, 313
*Corresponding author: irina.mocanu@cs.pub.ro

Georgiana Scarlat
Computer Science Department, University Politehnica of Bucharest
Romania, RO-060042 Bucharest, Splaiul Independentei, 313
georgiana.scarlat@cti.pub.ro

Lucia Rusu
Faculty of Economics and Business, Babes Bolyai University of Cluj-Napoca
Romania, 400591, Cluj-Napoca, Teodor Mihali, 58-60
lucia.rusu@econ.ubbcluj.ro

Ionut Pandelica
Agora University of Oradea
Romania, 410526, Oradea, Piata Tineretului, 8
ionut.pandelica@univagora.ro

Bogdan Cramariuc
IT Center for Science and Technology
Romania, 11702, Bucharest, Av. Radu Beller, 25
bogdan.cramariuc@citst.ro

Abstract: For elderly people that are living alone in their homes there is a need to permanently monitor them. One of this aspect consist in knowing their indoor position and motion behavioural status, in real time. One possibility for indoor positioning of an user consists in understanding the images provided by supervising cameras. In this case the main aspect is represented by recognition of objects from these images. Thus, object recognition plays an essential part in understanding the environment and adding meaning to it. This paper presents a method for indoor localisation based on identifying the user's context. The user's context is computed based on object recognition and using a probabilistic ontology. The key element is represented by the probabilistic ontology that describes objects, scenes and relations between them. This ontology contains probabilistic relations that are learned using a large database. Results show that given a set of object detectors with high detection rate and low false positive rate, the system can recognize the user's context with high accuracy.

Keywords: object recognition, detection rate, probabilistic ontology, context identification.

1 Introduction

The high number of elderly who are living alone, or who are spending too much of the day without supervision of specialized people is increasing exponentially. With the development of Active and Assisted Living (AAL) technologies the localisation of persons is easier.

Indoor positioning/localisation systems (IPS) is similar with GPS and can be used successfully to locate people or objects inside buildings via mobile devices (smartphones or tablets). IPS relies on cameras mounted on walls or ceiling that work together to recognize user's context or objects. Results are in highly accurate position. Like GPS, IPS systems can detect the direction of movement, and it can predict the path based on this information so that accurate position remains as the space is displaced [20].

Hospitals and medical centers can benefit from this indoor localisation systems for staff, patients and managerial purposes. IPS staff includes: rapid location of colleagues in the building, finding records and mobile devices, notifications when and where patients are checked. IPS for patients that are seek can achieve the below benefits: automatic checking of the building's entrance, turn-by-turn dynamic way finding meetings/appointments with relevant information based on location, way finding back to the parked car.

For people with Alzheimer, eHealth and mHealth solutions includes complex IPS technologies, which are essential for the pursuit of both indoor and outdoor patients [1]. Therapy Acceptance and Commitment (ACT) encourages seniors (patients in general) in two directions: (1) acceptance of thoughts and emotions, difficult and undesirable, and (2) adoption and simultaneous promotion, of actions and behaviors in daily practice, which is a consistent of individual values. ACT includes mindfulness exercises that promotes contact with the present [1].

The goal of the system proposed in this paper is to offer a reliable context detection of a user in his home. The user's context will be represented by surrounding objects. Thus the main problem that must be solved consist in object recognition. Object recognition has evolved very much in the past decade but the current state of the art solutions are still far from what the human brain can do. Moreover, training object detectors involves a big amount of resources such as computational power and large image training set. Hence, object recognition is the main part of the proposed system - especially extracting the meaning from the object configurations found and detect the type of the scene in which they are in. For this scope, we propose a probabilistic ontology that describes objects, scenes and relations between them.

The system is described by a generic probabilistic ontology which can be instantiated according to the set of objects and scenes that need to be recognized. The ontology contains probabilistic relations that are learned using a large database containing thousands of images (LabelMe image database [15]) of annotated images with object and scene information. The results of analyzing the co-occurrence and spatial relations between objects are then used to improve the object recognition process.

For implementing a semantic information based solution for scene recognition, a large number of object detectors are needed because each scene can have so many different object configurations. Training so many object detectors is one problem, but it can be solved given the necessary resources. Another important problem is that using so many object detectors to scan through an image can be time consuming. The solution proposed in this paper tries to reduce the number of image scans to a minimum. This can be achieved if the algorithm would somehow "knows" what objects to search for. This would be possible in scenarios in which some objects are already detected and the next searched objects would be the ones that are relevant to the object context found so far. By doing this, the algorithm could recognize the scene after only a small number of object scans.

The rest of the paper is organized as follows: Section 2 presents some existing methods for object recognition. Section 3 contains the proposed method for improving scene recognition by using semantic information organized as a probabilistic ontology model. Section 4 presents the evaluation of the proposed system. Conclusions and future work are given in Section 5.

2 Related work

The IPS systems have made significant progress in the last years. The utility of those types of systems on GPS Tracking Devices is appreciated both for seniors, and also for health care professionals (HCPs) and Caregivers. Bellow are described some of the most popular systems and applications for HCPs and caregivers [11].

Balance is an application geared specifically for Alzheimer's Caregivers, which works on iPhone and iPad. The balance features are: Alzheimers disease references and information, Alzheimer's caregiving and advice, advanced medication management features (refill date, start date and dosage), native scheduling features, adds categories, relevant to caregivers, "Doctor diary" for logging symptoms and taking notes, news about Alzheimer's.

Mobicare is a simple, straightforward and free iPhone and Web application. Their features are: profile of loved ones who are receiving personal information, including birth date, gender, basic insurance information, the contact information for one physician, basic symptoms, tracking based on 15 preset choices (i.e. insomnia, wandering, etc.), basic medication, tracking but with some limitations.

Dementia Caregiver Solutions is an informational application for dementia caregivers. Its features are: perform advices for addressing the difficult behaviors associated with Alzheimer's and other types of dementia. They also have bookmark or "star" articles you wish to read in the future.

Object recognition is a vastly studied topic for which many systems were developed, however none of them are even close to the performance with which human brain can recognize objects, even though there are many variations in light, shape and color. The process through which the human brain accomplishes object recognition with very high speed and accuracy has been intensely studied by neurologists. Based on [2]: "The ability to rapidly recognize objects despite substantial appearance variation, is solved in the brain via a cascade of rflexive, largely feed forward computations that culminate in a powerful neuronal representation in the inferior temporal cortex. However, the algorithm that produces this solution remains poorly understood". In [3] suggests that "object recognition does not only involve physical properties (such as shape, color and texture) of the objects but also semantic information which includes the understanding of its use, previous experience with the object and how it relates to others".

Some approaches in object recognition were purely based on visual properties of the objects. For example, paper [5] tries to detect general classes of objects, not just specific ones by using part-based modelling and recognition of objects. The pictorial structure models were first introduced in [4]. They describe how a set of object parts arranged in a flexible configuration are used to model an object. The object parts represent their localized visual properties. The flexible object parts configuration is represented by pair-wise object connections. This approach is suitable for generic recognition problems because of the complex description of object visual features.

Another machine learning approach is presented in [18]. Their solution is capable of processing images extremely fast and achieves high detection rates. Integral images are used to speed up computations. Also an important aspect is the fact that they train extremely efficient classifiers using an algorithm based on AdaBoost, but that are used only a small number of visual features which are selected as being critical features. Many regions of the image than on background areas are discarded in the early stages of the algorithm. This is accomplished by combining increasingly more complex classifiers in a cascade. A major advantage of this solution is that it can run in real-time applications at 15 frames per second.

Both of the previously described systems are very promising but they lack semantic information about objects, which is crucial in recognizing objects with large variations in shape, such as furniture objects and others.

A synergy between Google and Toronto University research team obtain a remarkable result called MultiModel, which solve multiple translation tasks, image captioning with COCO dataset, a speech recognition corpus, and an English parsing task. This model can caption images, categorize them, then translate to French and German and construct parse trees, by spanning multimple domains. MultiModel used encoder-decoder architectures, and applied to neural machine translation Extended Neural. GPU is another model which used a recurrent stack of gated convolutional layers and ByteNet used left-padded convolutions in the decoder. Compared to Extended Neural GPU and ByteNet, MultiModel idea improves efficiency.and obtained good results in image classification [7].

The Inception deep convolutional architecture was called GoogLeNet or Inception-v1, than the was refined by the introduction of batch normalization in Inception-v2, or by additional factorization ideas in the third iteration - Inception-v3, later Inception-v4 with similarly expensive hybrid Inception-ResNet versions for both residual and non-residual Inception networks [17].

Another results was offered by CapsNet, which used a 3 layer architecture for convolutional neural networks (CNNs) for translated replicas of learned feature detectors. The primary capsules are the lowest level of multi-dimensional entities and corresponds to inverting the rendering process. The second layer (PrimaryCapsules) is a convolutional capsule layer with 32 channels of convolutional 8D capsules. The third Layer (DigitCaps) has one 16D capsule per digit class and each of these capsules receives input from all the capsules in the layer below.The implementation is in TensorFlow using the Adam optimizer [16].

There are, however, approaches that use semantic information concerning objects. One such approach is presented in paper [12, 13]. Object recognition and scene understanding is strongly influenced by semantic and context-based information from a psychological point of view. Therefore, they use the context information to improve object recognition based on visual properties. Their approach presents how to extract context probability maps from images. Also, based on these maps, they learn specific configurations for a set of object classes. The final goal is to filter out false positives.

There are new methods based on deep learning for object detection and recognition. In the context of object detection, several network architectures have been proposed [8], [9] and found to outperform methods based on traditional hand-crafted features. Most of these models were trained on RGB datasets such as PASCAL VOC to predict the bounding boxes of objects from images.

The main difference between the solutions presented in paper [12] and the solutions presented in this paper is that, here the context is used not only to remove objects that don't fit the context, but also for inferring what other objects can be found in the same context. For example, if a keyboard was detected at a previous step, then there is a high probability that a mouse may also be present in the same scene. This reasoning is used to make the system converge faster in order to recognize the scene by running the most relevant object detectors for the current context. For example, if a TV was detected, the system shouldn't run an object detector for finding a car, but should run an object detector for finding a coach. Also, this system is designed in a generic way, thus allowing the use of any objects and scenes. This means that entities modeled by the ontology are abstract and can be mapped to any set of objects and scenes as long as these have the needed correlations between them (objects are relevant for the chosen set of scenes).

3 System description

The system uses as stored data both a pre-trained probabilistic ontology model containing object and scene entities and relations among them and a set of object recognizers mapped on the object entities contained by the probabilistic ontology model, as given in Figure 1.

The object recognition module can run, at request, a specific object recognizer and provide the result as a pair of object confidence and object bounding box. ***The inference module*** represents the main reasoning algorithm and, at each iteration, sends a request to ***the object***

recognition module for finding a certain object. Based on the found/not found object results, the ***inference module*** uses the ***probabilistic ontology*** to filter out false positives, to determine what object to inquire about next and to determine if a scene can be recognized.

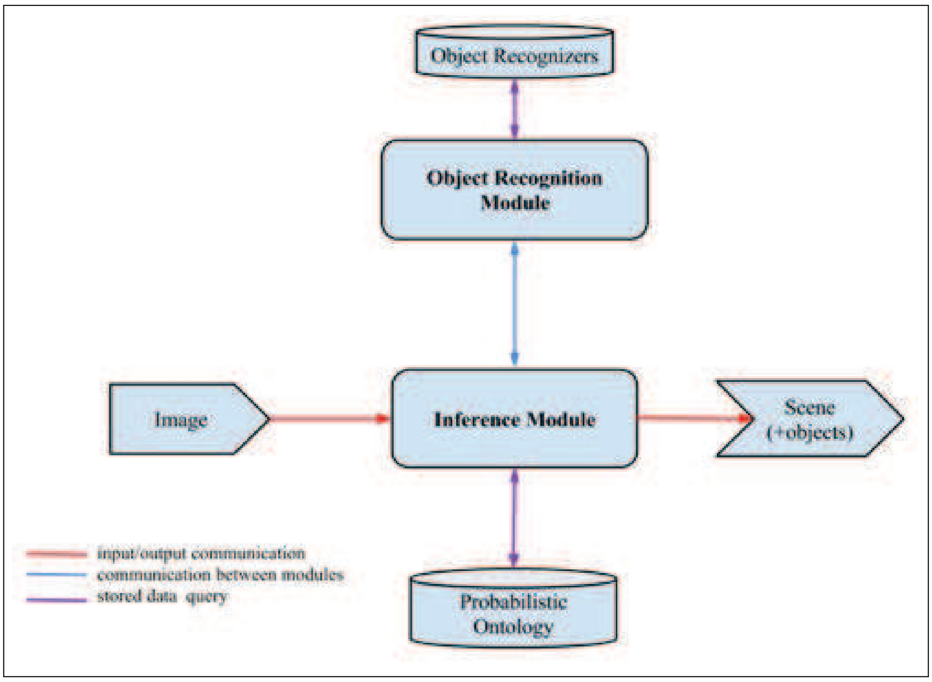

Figure 1: System architecture

In order to implement a reliable system for scene recognition, we consider the following steps:

- Selecting the ontology's structure: finding the set of relevant scenes and objects and mapping the meaningful relations among them into ontology relation types with associated attributes that reveal their semantic understanding.
- Choosing a large training data set that contains diverse context configurations for the chosen set of objects.
- Implementing a component that scans through the training dataset to compute object probabilities, scene probabilities and relation probabilities and aggregate all this information into a ***probabilistic ontology*** model.
- Implementing a object recognition method that allows testing the system and analyzing the influence on performance that object detection rate has on it.
- Doing ontology-based reasoning using already found objects to determine if a new found object belongs or not to the current context, thus filtering object recognition false positives.
- Doing ontology-based reasoning using previous object search results to determine what other objects can appear in the current scene in order to make the number of object recognition scans needed to recognize a scene much lower.
- Doing ontology-based reasoning to recognize a scene based on the previously found objects.

The system can run in two modes. The first one is the mode in which the system is used for running tests batches and computing system statistics. The tests are run on a big test database (approximately 3000 images) and the computed statistics consist of scene recognition accuracy, mean number of iterations needed to converge to recognise a scene and total number of removed false positives from the entire test set. This data is used to evaluate the system and analyze how different changes and improvements can affect its performance.

The second mode is used for viewing the analyzed image and obtaining the label associated to the image. An object is searched into the image - if the object is found then the most probable scene's name will appear. After that, correlated objects with already found are searched in the image. The rectangles for founded objects are displayed on the input image and the recognized scene's name is provided.

3.1 Object recognition

The scene recognition problem can also be approached using only image processing algorithms that analyze low level information about color, shapes and texture. This approach has the advantage that the classification model is easier to learn and design, but the disadvantage is that this method cannot distinguish very well between different indoor scenes such as living room, bedroom and so on. This is due to the fact that indoor scenes usually have similar colors, shapes and textures. Such scenes with similar visual features can only be distinguished using more top level information about the image such as objects and correlations between them.

Moreover, neurological studies [20] show that even the human brain cannot distinguish well objects and scenes unless there is some semantic meaning attached to them. As a result, scene recognition is best approached using semantic information regarding the scene.

Using object recognition for solving a scene recognition problem has the advantage that once the objects are found, recognizing the scene becomes a much simpler problem. However, there is a huge disadvantage to this approach. Object detectors can sometimes generate false positives. An object that is "alien" to the current object context can affect the final recognized scene very much, especially if there are only a few objects in the scene. The approach presented in this paper shows how contextual information can also be used to filter out false positives, thus increasing overall scene recognition accuracy.

Both ideas presented above about reducing the number of object scans and filtering out false positives are based on the relations that exist between objects. Some objects are strongly related among them (for example: keyboard and mouse, table and chair, etc.) and others are hardly related and can't be seen together in the same scene very often (for example: tree and TV, refrigerator and car, etc.). However, these relations are not always applicable (for example: keyboard can appear in a scene without a mouse). Therefore, they should be modeled stochastically. In this paper, the chosen model for representing the relations between objects is a probabilistic ontology.

The ontology used in this paper contains as entities objects and scenes and as relations the stochastic relations among objects and among scenes and objects. Every ontology entity has an apriori probability and every relation is also described by the probability of it to be true in a scene. The object to object relations from the ontology are described by an attribute called *Interpretation* that helps distinguish between positive and negative relations. Positive relations are those for which the objects are semantically connected to each other, meaning that this makes it more probable for them to appear in the same scene. The negative relations are exactly the opposite and they mean that the two objects are less likely to be found together.

For filtering out false positives obtained as a result of object detection, the current approach uses the previously found objects in the scene and the relations that connect them to the newly found object. If these relations are positive ones, this will increase the detection confidence of the checked object, but if the relations have a negative interpretation then this will decrease the detection confidence. If the newly found object's detection confidence has decreased considerably after this update, then it is considered to be a false positive and it will be eliminated from the found objects list.

If the first detected object is a false positives there are not afterward corrections applied. However, a prevention method is used in order to decrease the chances of this to happen. This measures is to use the false positive rate of each object detector as an influencing factor in choosing what object to search for in the case when no objects were found so far. This means that objects whose object detectors have lower false positive rates are preferred in the initial steps of the algorithm.

Choosing what objects to search for so that a scene can be recognised faster uses the positive and negative relations between previously searched objects and current objects that are candidates for search. This means that candidate objects that are in positive relations with previously found objects with high detection confidence are more suitable to be searched for next than candidate objects that are in negative relations with previously found objects.

Also, this reasoning is applied not only for the searched and found objects, but also for the searched and not found objects. If an object was not found after scanning the current scene, then candidate objects that are in a negative relation with it are more suitable for the next search than the ones that are in a positive relation with it. This reasoning is applied so that the next object choosing criteria can be more complex even if no objects were found yet.

The weight of the influence the not found objects have on choosing a new candidate is smaller than the weight for the found objects. This is justified by the fact that even though an object is not found in a scene, it does not necessarily mean that the current context is not suitable for it. For example, in a kitchen scene there might not be a stove object found, but this does not mean that other objects related to it cannot exist in that scene. However, if there are two equally suitable objects candidates according to their relations with already found objects in the scene, a tie breaker between them can be the same criteria based on not found objects after scanning the image. This reasoning is useful if the algorithm is in a state when all the previous searched objects are not found and it should try to search objects from different contexts than the ones searched before so that the chances of finding a new object increase.

All the criteria for choosing the next object to scan the image described earlier are combined into a fitness formula and the candidate object with the highest fitness value is chosen. The fitness value can also take into account the object's associated object detector false positive rate in case no objects were found yet in order to avoid the situation when the first found object is a false positive. In order to recognize a scene after some objects were found, the proposed solution uses the semantic information stored as relations between scenes and objects inside the ***probabilistic ontology***. The relation ***inScene*** is described by the probability of an object to be in a certain scene. At each step of the algorithm, the set of already found objects in the scene is used together with the relations between them and the entire set of scenes to compute the a fitness value for each scene.

The number of objects needed to recognize a scene can vary very much depending on the objects and how correlated they are with a certain scene. In some cases, only a few objects can be enough to know the scene as long as their detection confidence is big enough. For example, if a stove object was found and it has a very high detection confidence, then the most probable scene by far is kitchen.

Every time a new object is found, the fitness value is computed for each scene and if the scene with the highest fitness value is much beyond average then it is returned as the recognized scene and the object recognition process ends. The probabilities contained in the ontology are computed based on a large annotated image dataset, such that it can be applicable in general cases. The entities and the topology of the ontology can vary according to the use case it is designed for, but it is important that they reflect meaningful semantic information about objects and how they relate to one another and scenes and what objects are most probable to be contained by them.

Probabilistic ontology

The probabilistic ontology can be modeled using any topology structure and any entity set as long as these are relevant to the current use case. However, there are some restrictions on how to build the ***probabilistic ontology***. More precisely, the entity set must contain both scenes and objects. The object set has to be identical to the object set used by the ***object recognition module***, and the objects must be relevant for the scene set.

The general structure of the ***probabilistic ontology*** is given in Figure 2. The possible relations are: *inScene*, *is-a*, *Object-Object relation*, *hasVisualFeature*.

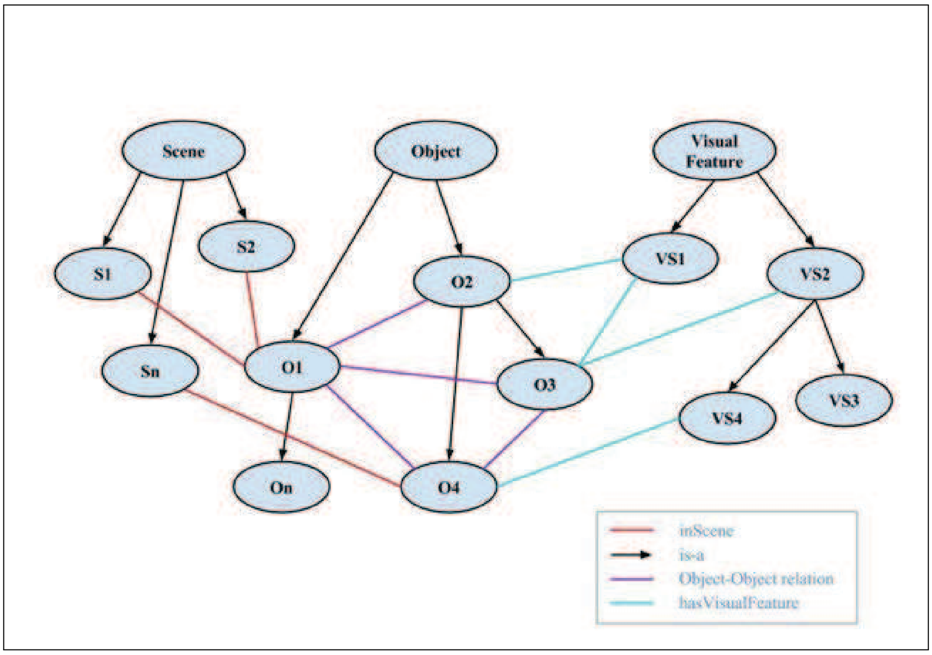

Figure 2: Ontology structure

There is no restriction regarding the stochastic relations between objects, except the fact that the relations have to be either positive or negative and weights have to be provided for each relation. The relation between objects and scenes is restricted to be of the type **inScene**, which reflects how probable it is for an object to be in a scene. Also, this relation should fully connect all the objects and scenes. Another restriction regarding the ontology building is that there should be no 0 value probabilities (in the case there are, they should be replaced with a very small value, close to 0).

An example of a part of the ***probabilistic ontology*** is given in Figure 3.

Each object to object relation contained by the ***probabilistic ontology*** has a fixed format. We consider the following relations properties:

- **Name**: A string that uniquely identifies each relation
- **Interpretation**: Represents how the relation should be interpreted during reasoning (POSITIVE, NEGATIVE).

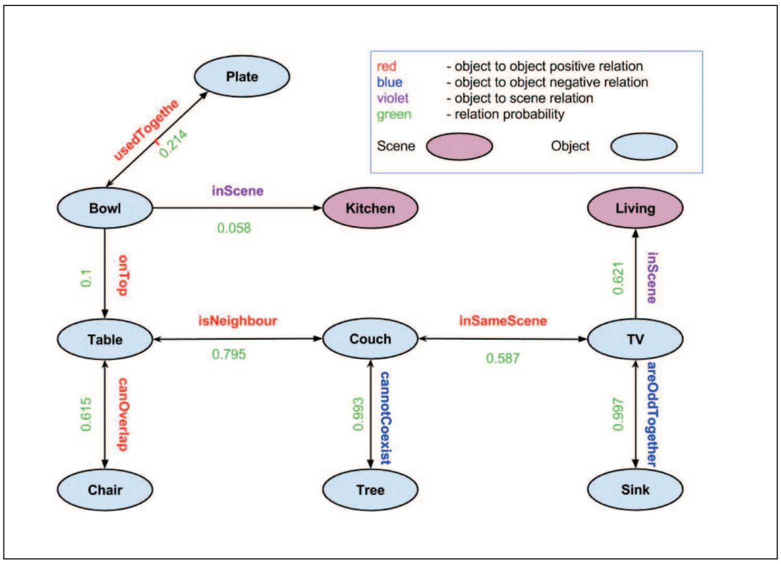

Figure 3: Ontology example

- **Check Weight**: Represents how important is this relation for checking newly detected objects and filtering out false positives.

- **Next Weight**: Represents how important this relation is for determining the next object to query the Object *Recognition Module.*

- **Check Rule**: Some relations are based on spatial requirements. This rule is used to check if the spatial requirements are met. For the relations that have no spatial meaning attached to them, this rule should be NONE.

This ontology contains as entities objects from a house and the following scenes: kitchen, living room, street, office.

We pointed relationship description between objects:

- **usedTogether**: Two objects are often used together. The object pairs that are in this relation are given by the user according to common sense information about the objects in the object set.

- **inSameScene**: Two objects often appear together in the same scene. The object pairs are determined by the object recognition algorithm according to the probability of the co-occurrence of each pair of objects.

- **cannotCoexist**: Two objects can't both be present in the same scene. The object pairs are determined by the object recognition algorithm according to the probability of the co-occurrence of each pair of objects.

- **areOddTogether** Two objects may appear together in the same scene, but their combination seems unnatural. The object pairs that are in this relation are given by the user according to common sense information about the objects in the object set.

- **onTop**: Most of the cases when the two objects appear together, one of them is placed on top of the other. The object pairs that are in this relation are given by the user according to common sense information about the objects in the object set.

- **canOverlap**: Most of the cases when the two objects appear together, their areas overlap. The object pairs that are in this relation are given by the user according to common sense information about the objects in the object set.
- **isNeighbour**: Most of the cases when two objects appear together, they are very close to one another. The object pairs that are in this relation are given by the user according to common sense information about the objects in the object set.

Properties of these relationship between objects are syntetize in Table 1. The rules are the following:

$$\textbf{OnTopRule} : Center(object1).y > Center(object2).y$$

$$\textbf{IsNeighbourRule} : Distance(object1, object2) \leq \frac{Diagonal(object1) + Diagonal(object2)}{2}$$

$$\begin{aligned}\textbf{CanOverlapRule} : Left(object1).x < Left(object2).x < Right(object1).x \\ AND \\ Left(object1).y < Left(object2).y < Right(object1).y\end{aligned}$$

Table 1: Properties of relationship between objects

Relation Name	Interpretation	CheckWeight	NextWeight	CheckRule
usedTogether	POSITIVE	0.8	0.95	NONE
inSameScene	POSITIVE	0.6	0.9	NONE
cannotCoexist	NEGATIVE	0.98	0.98	NONE
areOddTogether	NEGATIVE	0.75	0.85	NONE
onTop	POSITIVE	0.8	0.7	OnTopRule
canOverlap	POSITIVE	0.8	0.65	CanOverlapRule
isNeighbour	POSITIVE	0.8	0.7	IsNeighbourRule
inScene	BELONGING	0	0	NONE

The values for the relation *CheckWeight* and *NextWeight* attributes are chosen manually so that they reflect how reliable a relation is for checking for false positives (*CheckWeight*) and how reliable a relation is for predicting the presence of one of its referred objects when the other one is already found in the scene (*NextWeight*). For example, the relation **usedTogether** describes a much stronger bond between objects than the relation **inSameScene**, therefore its *CheckWeight* is much bigger (0.8 vs. 0.6). However, both usedTogether and inSameScene are relations between objects that are frequently found together, therefore their *NextWeight* is very big (0.95 and 0.9).

These weights are used so that not all relations influence in the same way false positive filtering and next object choosing, because each of them has a different semantic interpretation and should affect the reasoning process in its own custom way. The *CheckWeight* and *NextWeight* attributes have the role of quantifying semantic relation attributes as numbers in the [0, 1] interval, that can be used as components in the **inference module** reasoning formulas.

The current ***probabilistic ontology*** contains the relation between objects and scenes: **inScene** describes how probable is for an object to appear in a scene.

The probabilities of each object, each scene and each relation were computed based on the LabelMe database [15] that currently contains 78840 annotated images.

3.2 Object recognition

The **object recognition module** has access to a set of predefined object detectors which are run at the request of the **inference module** on the input image or on a region of the input image. The result that is supplied by this module is represented by a tuple of the form (*FoundObject*, *BoundingBox*, *Confidence*). The bounding box represents the area in the image where the found object is placed. This information is used by the ***inference module*** to check ontology rules that have spatial requirements. The confidence of object detection is also used by the ***inference engine*** as a measure of how much the detected object influences reasoning regarding other related objects.

An important aspect worth mentioning is that not all the object detectors are necessarily run in order to recognize a scene, The **inference module** decides to inquire about a certain object which it believes it is more probable to appear in the image based on previous detections, and when it has enough information to infer the scene with a high probability, the system returns and no more detectors are run. This means that, given a powerful **inference module** that obtains reliable information from all the other components, the final result can be obtained very fast, avoiding the costly image scans that the object detectors apply.

The **object recognition module** is responsible for running object detectors on request. Also, this component has access to information regarding each object recognizer that is relevant to the **inference module**. More precisely, the **object recognition module** can provide information about each object detector's detection rate and false positive rate. This information is meaningful for the **inference module** because it influences what object to inquire about.

3.3 Inference module

The **inference module** represents the system's component that implements the main algorithm for scene recognition. A scene can be inferred based on the objects recognized using the **object recognition module**. The information regarding found or not found objects is used to interrogate the **probabilistic ontology**. This ontology contains information about the stochastic relations between scenes and objects and between objects. The algorithm starts by choosing a first object to interrogate the **object recognition module** about. The first object is chosen based on false positive rate. Therefore, the first interrogated object is the one with the lowest false positive rate. This is done because at the initialization of the system, there is no information available about the input image, therefore it is important not to start with invalid information that can compromise future reasoning. After the initialization step when the first object is chosen, the algorithm enters a repetitive loop. At each iteration, the **object recognition module** is queried about the existence of a certain object inside the input image. If the object is found, then its confidence and bounding box are available. Next, a set of inference rules are used. The **first inference rule** has the role of filtering out false positives. The information used to update an object's detection confidence is represented by the positive and negative relations between the current object and the previously found objects. If the newly found object is in a positive relation with a previously found object then its detection confidence will increase. On the other hand, if the current object is in a negative relation with a previously found object, then its detection confidence will decrease. A visual description of the **first inference rule** is given in Figure 4.

The **first inference rule** are the following rules (R_i), i=1,7:

- *(R1)* Compute the list of previously found objects rejected by the current found object.
- *(R2)* Compute rejection factor of the found object REJ_{FOUND} according to each rejected object relation.

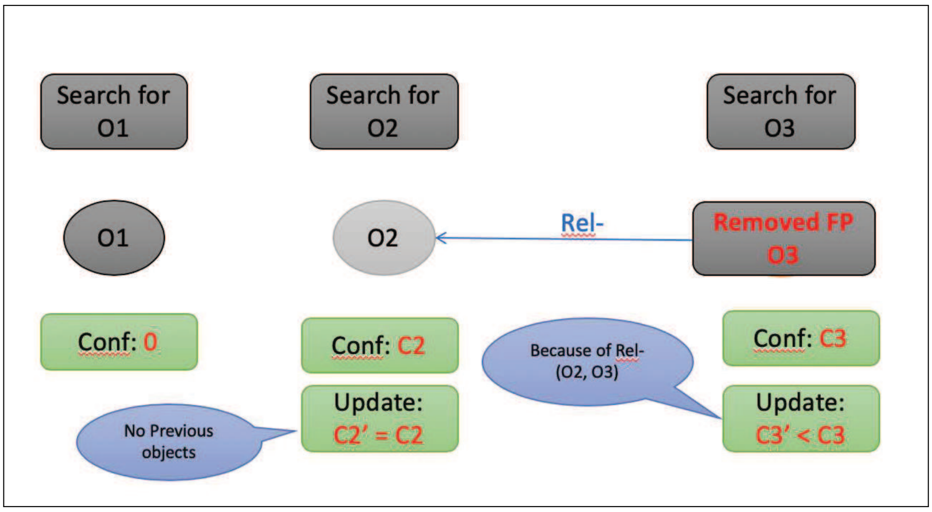

Figure 4: Description of the first inference rule

- *(R3)* Compute the list of previously found objects attracted by the current candidate object.
- *(R4)* Compute the found object attraction factor. $ATTR_{FOUND}$ according to each attracted object relation
- *(R5)* Update the current found object's detection $C(O)$ rate according to the rejection and attraction factors already computed: $ATTR_{FOUND}$, REJ_{FOUND}

$$C(O) = C(O) * (1 - \frac{REJ_{FOUND}}{|Objects_{REJECTED}|} + \frac{ATTR_{FOUND}}{|Objects_{Attracted}|})$$

- *(R6)* If the found object detection confidence $C(O)$ has decreased more than 20%, then it is filtered out.

The **second inference rule** has the role of determining what the next searched object should be. This rule is important because it helps the algorithm converge faster to the recognized scene and it avoids running all the object recognizers on the input image. In order to determine what object to search for next, information about the positive and negative relations from the **probabilistic ontology** is used against the set of previously found and not found object sets. Therefore, the next object to inquire the **object recognition module** about is the one that best fits the current context. More precisely, not searched objects that are in a positive relation with previously found objects will have a bigger chance at being next and not searched objects that are in a negative relation with previously found objects will have a smaller chance at being next. A visual description of the **second inference rule** is given in Figure 5.

Similar reasoning is used also in the case of previously not found objects (objects that were searched for in the input image but were not found), but in this case the positive relations have a negative impact on the new object's chance at being next and the negative relations have a positive impact on the new object's chance at being next. However, the reasoning based on previously found objects has a bigger weight in influencing the detection rate than the one on previously not found objects. Treating these two cases differently is justified by the fact that, even though an object is not found in the input image, this does not necessarily mean that the object cannot belong to the current context.

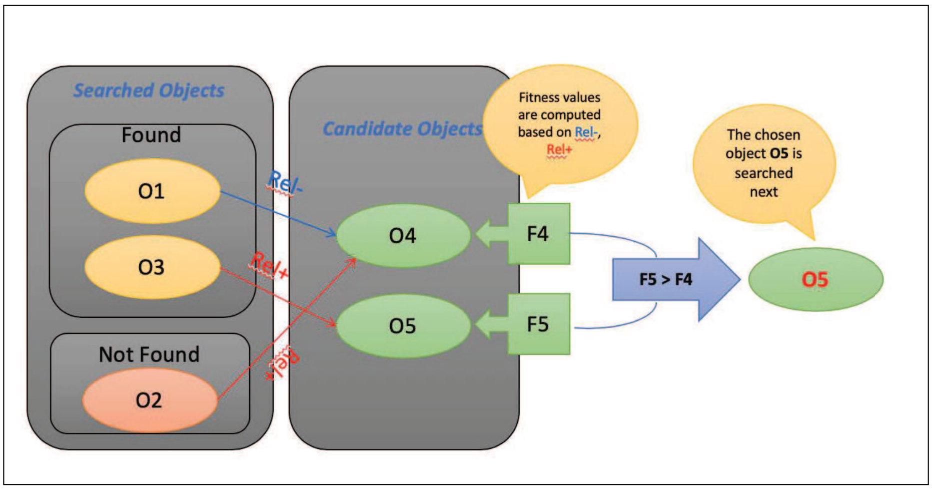
Searched Objects
Found
O1
O3
Not Found
O2
Candidate Objects
Rel-
Rel+
Rel+
O4
O5
F4
F5
Fitness values are computed based on Rel-, Rel+
The chosen object O5 is searched next
F5 > F4
O5

Figure 5: Description of the second inference rule

The scene recognition algorithm reasoning can be compromised if the first found object is a false positive. This can lead to filtering out other following correct detection as false positives and the final result becomes meaningless. Therefore, when choosing the next object to search for, the false positive rate of the object detector is also taken into consideration if no objects have been found so far. This reduces the risk of starting the algorithm with a false positive detection.

- *(R1)* Obtain the list of previously found objects rejected by the current candidate object $Objects_{REJECTED}$
- (R2) Compute the found object rejection factor REJ_{FOUND} according to each rejected object relation
- *(R3)* Obtain the list of previously found objects attracted by the current candidate object $Objects_{ATTRACTED}$
- *(R4)* Compute the found object attraction factor $ATTR_{FOUND}$ according to each attracted object relation
- *(R5)* Repeat the steps above for previously not found objects and compute $REJ_{NOT-FOUND}$ and $ATTR_{NOT-FOUND}$ factors.
- *(R6)* Compute the fitness of the current candidate object:

  $$F(O_{CAND}) = (ATTR_{FOUND} - REJ_{FOUND}) + \alpha * (REJ_{NOT-FOUND}) - ATTR_{NOT-FOUND})$$

  where α represent a sub-unit weight for decreasing the influence of the not found objects.

  If there was no object found, the fitness value is also influenced by the false positive rate (FP): FP($Detector_{O_{CAND}}$) of the candidate objects:

  $$F(O_{CAND}) = (1 - FP(Detector_{O_{CAND}}) * F(O_{CAND})$$

- *(R7)* Choose the next object to search for from the list of candidate objects by finding the one with the biggest fitness value:

  $$O_{NEXT} = argmax(F(O_{CAND}))$$

The **third inference rule** has the role of determining the fitness value of each scene for the current image. If the scene with the highest fitness value is bigger than a threshold proportional to the average scene fitness, then the algorithm returns the scene and the object searching process is ended. For computing the fitness value of each scene the Inference Module uses the set of objects found so far to interrogate the **probabilistic ontology** about the probability of having each found object in the current candidate scene. The ontology relation that contains information about the probability of an object to appear in a scene is called **inScene**. This is a mandatory relation that should exist in any instance of the **probabilistic ontology**, no matter what object set and scene set is used or what other relations are used between objects. A visual description of the **third inference rule** is given in Figure 6.

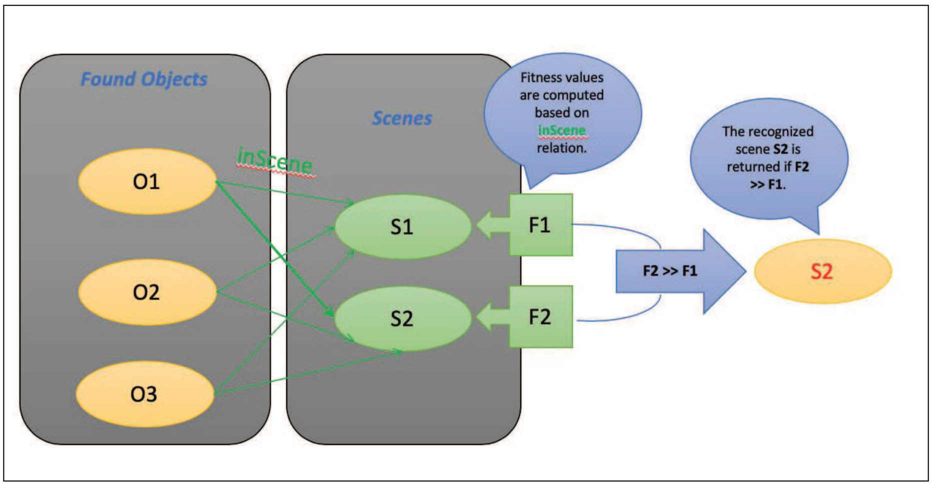
Found Objects
Scenes
inScene
O1
O2
O3
S1
S2
F1
F2
Fitness values are computed based on inScene relation.
The recognized scene S2 is returned if F2 >> F1.
F2 >> F1
S2

Figure 6: Description of the second inference rule

The fitness value for a scene S is computed as:

$$Fitness(S) = \sum_{O \in Objects_{FOUND}} log \frac{P(inScene(O,S)}{P(O)}$$

where $Objects_{FOUND}$ represents the set of the previously found objects, *P(inScene(O,S))* represents the probability of the ***inScene*** relation between object O and scene S and *P(O)* represents the apriori probability of object O (obtained from the ***probabilistic ontology***).

The best candidate scene is chosen by:

$$Scene_{BEST} = argmax(Fitness(S))$$

The best candidate scene is returned as the final recognized scene if it's fitness is bigger than a threshold proportional to the average scene fitness:

$$Fitness(Scene_{BEST}) > \delta * \frac{1}{|Scenes|} * \sum_{s \in Scenes} Fitness(S) \tag{1}$$

If no scene is recognized at the current iteration, then the object searching process continues and the previous three inferences are applied again at each iteration until a scene can be inferred or all the objects have been searched for. In the latter situation, the returned scene is the one with the biggest fitness value, without taking into consideration the equation 1.

4 System evaluation

The system is implemented to be flexible regarding replacing any of the components implementation. Every module communicates with the others through an interface, making the components loosely coupled. Therefore, in order to replace the current implementation of a module all that is needed is to implement the module interface. The scene recognition application is written in Java language using Eclipse as IDE. For object recognition algorithm we use the YOLO network [19]. The scene recognition application was tested using images from LabelMe database [15]. This database includes images for many indoor and outdoor scene types. For testing the system, only images with relevant scene types were used: kitchen, office, living room and street. The images are annotated with scene information under the attribute ***scenedescription***. After the ***probabilistic ontology*** probabilities are computed based on object and scenes co-occurrences inside the training image set, the model is stored inside an XML file. The XML ontology is then parsed by the application and mapped into Java objects that are used by the ***inference module***.

The pseudocode given in algorithm 1 that describes the reasoning algorithm implemented in the ***inference module***:

Algorithm 1 Reasoning algorithm

```
procedure reasoningAlgorithm($Imageimage$)
    $searchObject \leftarrow getObjectWithLowestFP()$
    $foundObjectsSoFar \Leftarrow []$
    $notFoundObjectsSoFar \Leftarrow []$
    while true do
        $(confidence, boundBox) \Leftarrow findObject(searchObject, image)$
        $updatedConfidence \Leftarrow checkFoundObject(searchObject, confidence, foundObjectsSoFar)$
        if $updatedConfidence - confidence > threshold$ then
            $foundObjectsSoFar \Leftarrow foundObjectsSoFar \cup searchObject$
        else
            $notFoundObjectsSoFar \Leftarrow notFoundObjectsSoFar \cup searchObject$
            if $[] \neq foundObjectsSoFar$ then
                $sceneFitnessList \Leftarrow computeAllSceneProbabilities()$
                $bestScene \Leftarrow maxFitnessScene(sceneFitnessList)$
                $meanFitness \Leftarrow meanFitnessValue(sceneFitnessList)$
                if $bestScene.fitness > \epsilon * meanFitness$ then
                    return $bestScene$
                end if
            end if
        end if
        $searchObject = getNextMostProbableObj(foundObjSoFar, notFoundObjSoFar)$
    end while
end procedure
```

The ***object recognition module*** has access to a set of object detectors and to information regarding their performance: detection rate and false positive rate. We use YOLO (You Only Look Once) network [19]. It is a very robust method, which is almost invariant to position and lightning. It simultaneously predicts multiple bounding boxes and class probabilities for those boxes. YOLO trains on full images and directly optimizes detection performance. YOLO reasons globally about the image when making predictions. Unlike sliding window and region proposal-based techniques, YOLO sees the entire image during training and test time so it implicitly encodes contextual information about classes as well as their appearance. And also YOLO learns generalizable representations of objects.

For testing the system, only images with relevant scene types were used: kitchen, office, living room and street. The images are annotated with scene information under the attribute *scenedescription*. The database contains many object types that appear in many combinations. The images in the database are annotated with object information. For the current system, the relevant annotated object information is represented by the object outline polygon and the "verified": flag. The object polygon is used to obtain the object's bounding box for the object recognition stubs and the "verified" flag is used to select the object's confidence. If an object is verified, then it was annotated correctly and the object recognition will give it a high confidence, otherwise a lower confidence value is assigned. Some examples of LabelMe annotated images can be seen in Figure 7.

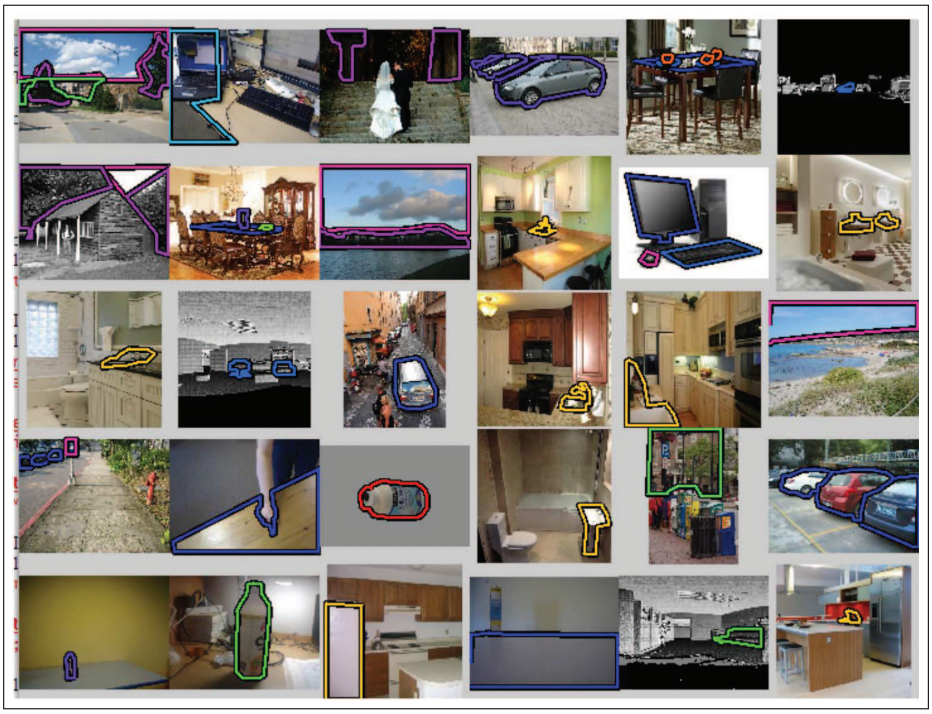

Figure 7: LabelMe annotated images example

Figure 8 shows some examples of recognising kitchen, office, living room. The current system was evaluated on a test set containing a total of 3381 annotated images from LabelMe database [15]. The aggregated evaluation results obtained after running the application on the test database can be seen in Table 2.

Table 2: System evaluation results using stubs

Scene Name	Accuracy	Mean Iterations	Removed FP	Test Images
kitchen	89%	7	16	651
office	90%	9	35	914
living	91%	8	30	775

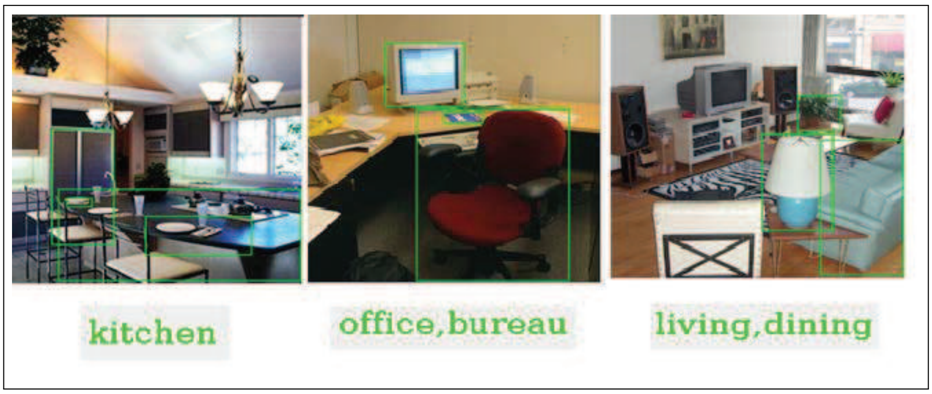

Figure 8: Example of scene recognition

5 Conclusions and future work

This paper presents a method for context detection of a user in an indoor space. The proposed method is based on the results of an object recognition process. In order to be able to recognize a wide range of scenes, the number of objects that need to be recognized can become very big. Running a large number of object detectors can be time consuming; therefore the current approach uses semantic information about objects and scenes to speed up the scene recognition process and to eliminate false positives that can have a negative impact on the final result. This semantic information is organized as an object and scene probabilistic ontology model. Reasoning is done using the stochastic relations between objects and between objects and scenes and its outcome can influence in which order objects are searched for or if a newly detected object is eliminated for being a false positive. Scene recognition is influenced by the semantic relations between scenes and objects, and the object recognition process ends as soon as there are enough objects found to determine the scene. Results show that given a set of object detectors with high detection rate and low false positive rate, the system can recognize a scene with high accuracy and in a small number of iterations.

As future work, the ontology model can be extended to meet domain-specific requirements because it is easily adaptable to different domains.

Acknowledgment

This research and paper was co-founded by both the Executive Unit for Financing Higher Education, Research and Development and Innovation through Partnership Program, project "Mobility pattern assistant for elderly people", project number PN-II-PT-PCCA-2013-4-2241, by University Politehnica of Bucharest, through the "Excellence Research Grants" Program, UPB - GEX. Identifier: UPB-EXCELENTA-2016 "Optimizarea Activitatilor Zilnice Folosind Deep Learning Implementata pe Sisteme Reconfigurabile" / Daily Activities Using Deep Learning Implemented on FPGA, Contract number 8 / 26.09.2016 (code 341) and by the grant of the Romanian National Authority for Scientific Research and Innovation, CCCDI - UEFISCDI and of the AAL Programme with co-funding from the European Union's Horizon 2020 research and innovation programme project "IONIS - Improving the quality of life of people with dementia and disabled persons", project number AAL2017-AAL-2016-074-IONIS (Contracts 52/20017 and 53/2017).

Bibliography

[1] Burm, C. (2015). Dementia and Elderly GPS Tracking Devices, http://www.aplaceformom.com/blog/4-29-15-dementia-and-elderly-gps-tracking-devices/, last accessed October 2018.

[2] DiCarlo, J.; Zoccolan, D.; Rust, N. C.(2012). How does the brain solve visual object recognition, *Neuron*, 73(3), 415–434, 2012.

[3] Enns, J. T.; (2004). *The Thinking Eye, The Seeing Brain: Explorations in Visual Cognition*, W. W. Norton Company, ISBN: 0393977218, 2004

[4] Fischler, M.A. ; Elschlager, R.A., (1973). The Representation and Matching of Pictorial Structures, *IEEE Transactions on Computer*, 22(1), 67–92, 1973.

[5] Felzenszwalb, P.F, Huttenlocher, D.P.; (2005), Pictorial Structures for Object Recognition, *International Journal of Computer Vision*, 61(1):55–79, 2005.

[6] Gupta, S.; Girshick, R.; Arbelaez, P.; Malik, J. (2014). Learning Rich Features from RGB-D Images for Object Detection and Segmentation, *ECCV*, 345–360, 2014.

[7] Kaiser, L.; Gomez, A. N.; Shazeer, N.; Vaswani, A.; Parmar, N.; Jones,l.; Uszkoreit,J (2017). One Model To Learn Them All, http://arxiv.org/abs/1706.05137, last accessed October 2018.

[8] Li, B.; Wu, T.; Shuai1, S.; Zhang, L.; Chu, R., (2017). Object Detection via Aspect Ratio and Context Aware Region-based Convolutional Networks, arXiv:1612.00534v2 , https://arxiv.org.

[9] Leal-Taixe, L. (2016). Multiple Object Tracking with Context Awareness, http://arxiv.org/abs/1411.7935, last accessed October 2018.

[10] Maturana, D.; Scherer., S. (2015). Voxnet: A 3d Convolutional Neural Network for Realtime Object Recognition, *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 922–928, 2015.

[11] Napoletan, A. (2015). 10 Best (and Worst) Apps for Caregivers, https://www.aplaceformom.com/blog/best-and-worst-apps-for-caregivers-07-03-2013/, last accessed October 2018.

[12] Perko, R.; Leonardis, A., (2010). Context Awareness for Object Detection, *Computer Vision and Image Understanding*, 114(6), 700–711, 2010.

[13] Rehman, Z.; Kifor C.K. (2016). An Ontology to Support Semantic Management of FMEA Knowledge, *International Journal of Computers Communications & Control*, 11(4), 507-521, 2016.

[14] Ren, S.; He, K.; Girshick, R.B.; Sun, J. (2015). Faster R-CNN: towards real-time object detection with region proposal networks, *Advances in Neural Information Processing Systems*, 91–99, 2015.

[15] Russell, B. C.; Torralba, A.; Murphy, K. P.; Freeman, W. T. (2008). LabelMe: a Database and Web-Based Tool for Image Annotation, *International Journal of Computer Vision*, 77(1-3), 157–173, 2008.

[16] Sabour, S.; Frosst, N.; Hinton, G. E. (2017). Dynamic Routing Between Capsules, Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA,3859–3869, https://arxiv.org/pdf/1710.09829.pdf, last accessed October 2018.

[17] Szegedy, C.; Ioffe, S.; Vanhoucke, V. (2016). Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning, http://arxiv.org/abs/1602.07261, last accessed October 2018.

[18] Viola, P. A.; Jones, M. J. (2001). Rapid Object Detection using a Boosted Cascade of Simple Features, *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1-9, 2001.

[19] YOLO network, https://pjreddie.com/darknet/yolo/, last accessed October 2018.

[20] https://senion.com/indoor-positioning-system/, last accessed October 2018.