Integrated Computer-Aided Engineering 1 (2019) 1–5
IOS Press

Semantic Visual Recognition in a Cognitive Architecture for Social Robots

Francisco Martin-Rico [a,*], Francisco Gomez-Donoso [b], Felix Escalona [b], Jose Garcia-Rodriguez [b] and Miguel Cazorla [b]

[a] *Departamento de teoria de la señal y sistemas telematicos y computacion, University Rey Juan Carlos, Madrid, Spain*
E-mail: francisco.rico@urjc.es
[b] *Institute for Computing Research. University of Alicante, P.O.Box 99. E03080 Alicante, Spain*
E-mails: fgomez@ua.es, felix.escalona@ua.es, jgr@ua.es, miguel.cazorla@ua.es

Abstract. Cognitive architectures allow robots to perform their operations by drawing on a process that aims to simulate human reasoning. This paper presents an integrated semantic artificial memory system in cognitive architecture based on symbolic reasoning and a connective representation of the knowledge. This memory system attempts to simulate how humans learn to distinguish instances of particular objects within their class using a convolutional network to detect the relevant elements of an image. We use a vector with the extracted features to learn to discriminate an instance of another element from the same class. A novel feature of our approach is its autonomous learning process during the operation of the robot, integrating a deep learning embedding with a statistical classifier. The usefulness and robustness of this method are demonstrated by applying it to a social robot that learns to differentiate people. Finally, experiments are carried out to validate our approach, comparing the detection results with several alternative methods.

Keywords: cognitive architectures, people recognition, pose estimation, social robotics

1. Introduction

The learning process of robots is still far from the learning process of humans. Recent advances in convolutional neural networks have revolutionized the ability of machines to visually recognize classes of elements in images. Despite its high effectiveness, the learning process is offline, slow, and requires much computation. Humans, in contrast, learn by accumulating knowledge from experience.

Distinguishing a person is a desirable capacity of a social robot. Robots meet different people during their operation. In many tasks, they are required to distinguish them from others. Tasks such as serving drinks in a bar, following a person in a crowd, greeting the people around them by name, among others, are commons.

This paper presents the results of an investigation that mixes deep learning (DL) techniques and statistical classifiers to achieve a robust system of online learning and recognition of people. The proposed approach has different operating modes. In the learning mode, the robot learns to distinguish the person with whom it is interacting. The robot extracts the person's name from its interaction with the person. The aim is that, after several seconds of communication, the robot will be able to distinguish it from the rest of the people. In normal mode, the robot can identify all the people from whom it has learned to carry out tasks entrusted to it. This approach is not only

*Corresponding author. E-mail: francisco.rico@urjc.es.

valid for people, but can also be used to identify any visual elements.

In addition, the integration of this system into a cognitive architecture designed for social robots has been performed. This architecture provides mechanisms to control the changes in the operation mode. It is also essential to see how the recognition system's results are made available to the rest of the robotic system. This description will be useful to understand how the learning and recognition system is used.

The proposed approach has several advantages over current methods. The most obvious is that it does not detect classes of elements, but distinguishes particular instances within the same category. While works like [26] identify instances of a given object, we, however, are able to track those instances over time. The second advantage is its operation mode. Current DL approaches would require an offline process of labeling, training, and deployment on the robot. The proposed approach can alternate training phases with operation phases without turning off the robot. Moreover, this learned knowledge between robot operations is preserved. This technique is similar to online learning methods such as [8] and [5], as the model can be adapted to new knowledge received in execution time. Last but not least, learning a person requires less than 15 seconds of interaction (this is the maximum time that some competitions establish, as explained in the next paragraph). Such training times are unimaginable in current methods that use only DL.

Robotic competitions also provide a useful situation in which to test the system. Competitions, such as RoboCup, present a problem and a common scenario where research groups from around the world apply and contrast their research. The authors of the present work participated in the RoboCup SSPL@Home, in which a social robot is required to carry out a series of missions in a domestic environment to help a dependent person in their daily life. In one of the tests, the robot must follow a person out of the house to help her unload the car. The robot must learn to follow a person (the referee of the test) in less than 15 seconds without confusing them with any person it might detect during the follow-up. The proposed system will be able to address this situation without trouble, since the system is not based on recognizing a face, but an entire appearance, since the face is not visible when it is on the side during tracking.

Specifically, the main contributions of this work are:

- A person identification memory system (PIMS).
- An exhaustive experimentation of different classifiers focused on processing time and accuracy for the PIMS.
- An approach to integrate PIMS in an existing cognitive architecture for driving social robots.

The proposed approach combines DL architectures trained with a new dataset with feature extraction techniques. Some previous works also created a framework to combine different visual features [29] with a similar approach to the proposed one, but this approach uses DL embeddings instead of handcrafted features.

This work follows up our previous work [36] by, on the one hand, integrating the person identification system into a cognitive architecture for social robotics and, on the other, improving the person identification system and allowing it to detect when the appearance of the subject is not already learned.

The rest of the paper is organized as follows. Firstly, Section 2 analyzes the state of the art of person identification. Next, Section 3 presents a description of the cognitive architecture. Section 4 provides an in-depth description of the proposal. Subsequently, Section 5 is devoted to the experimentation carried out with the proposed PIMS module (only people recognition has been tested) with some available DL pre-trained networks and to the corresponding discussion. The

limitations of the system are addressed in Section 5.4. Finally, Section 6 draws conclusions and present possible future research directions.

2. Related Works

Most technical approaches to memory implementations are centered representations of the world where knowledge about the elements of the environment is maintained, together with values that indicate its reliability or accuracy. In [11] an anchoring system is presented, where each element has a reliability value associated with it, which depends on when an element has been perceived. In [51], reliability of the knowledge of an object is also associated with the time when this perception is obtained, but adding semantic relations between the elements of memory. A similar approach is presented in [38], although focused on the learning of elements in memory.

To maintain a memory, biologically inspired approaches attempt to emulate totally or partially the mental processes that are attributed to human beings. The separation between long-term, short-term, or episodic memories [13] is common in these approaches. In [22] the short-term memory focuses on the stimuli relevant to the current task, while the long-term memory contains episodic events that are derived from the interaction between a robot and a human. These types of long-term memories that remember episodes are discussed in depth in [50] and [14]. The concept of Working Memory [39, 49] is applied to robots in an attempt to provide a robot with biologically inspired cognitive abilities. The proposed approach is among the short-term memories, being its biological approach inspired by the process of information acquisition, based on neural networks.

The rest of this section presents a brief analysis of the state-of-the-art methods on person re-identification based on its activity.

People re-identification in videos is a challenging problem but it also promises a huge potential for a wide range of applications mainly related with security and surveillance and health care or human-machine interaction [17].

An automated re-identification mechanism takes as input either tracks or bounding boxes containing segmented images of an individual person, as generated by a localized tracking or detection process of a visual surveillance system. To automatically match people at different locations over time, a re-identification process typically takes the following steps: 1. Extracting imagery features; 2. Constructing a descriptor or representation capable of both describing and discriminating individuals; 3. Matching specified probe images or tracks against a gallery of people in another camera view by measuring the similarity between the images.

A classic taxonomy classifies recognition methods as either single-shot when only one image pair is used, or multi-shot when two sets of images are employed. With regard to the learning approach, it is categorized as a supervised method if, prior to application, it exploits labeled samples for tuning model parameters. Otherwise a method is considered as an unsupervised approach and no training data is used to train the system.

In recent years, deep learning (DL) techniques have surpassed the classic methods in most computer vision challenges [28, 52, 56]. Furthermore, in [40] a multiscale re-identification system is proposed.

However, these models suffer from a lack of training data samples. This is because most of the available datasets provide only two images per individual [18], which makes the model fail at test time due to overfitting. In this line, a number of new datasets have been proposed to solve this problem. Some of these are based on images: Market1501 [57], CUHK03 [31], DukeMTMC-reID [59], while other are based on video: MARS [58], iLIDS-VID [54] or PRID2011 [21].

Some recent works using DL models include [4], which proposes a deep convolutional architecture with layers specially designed to address the problem of re-identification. In [10],

the authors learn multi-scale person appearance features using Convolutional Neural Networks (CNN) by aiming to jointly learn discriminative scale-specific features and maximize multi-scale feature fusion selections in image pyramid input. In [30], a Tracklet Association Unsupervised deep learning (TAUDL) framework is proposed. It is characterized by jointly learning per-camera (within-camera) tracklet association (labeling) and cross-camera tracklet correlation by maximizing the discovery of most likely tracklet relationships across camera views. Some approaches employ graph deep neural networks like [47], which proposes a novel DL framework called Similarity-Guided Graph Neural Network (SGGNN). Given a probe image and several gallery images, SGGNN creates a graph to represent the pairwise relationships between probe gallery pairs (nodes) and utilizes such relationships to update the probe gallery relation features in an end-to-end manner.

A comprehensive and exhaustive survey on person re-identification can be found in [53]. In this analysis, it is concluded that the main drawback of deep learning-based approaches is that they cannot assure high accuracy and low computation cost because they need constant retraining. Nonetheless, our proposal mixes the accuracy of the deep learning models with the low computation cost of traditional machine learning methods.

Regarding the approaches used by RoboCup teams (in this link[1]) to solve the problem of re-identification, few teams use an elaborated method. In 2018, the team AUPAIR participating in the Social Standard Platform League (SSPL) used an improved Siamese [24] convolutional neural network architecture [3]. Using the score generated by these networks, they tag images of people for future re-identifications. A similar approach was used in 2019 by the CATIE team in the Open Platform League (OPL), but applied only to distinguish person already identified. The Team Lions (2019, OPL) used a specially trained Single Shot MultiBox Detector [34], a Kalman Filter, and a global nearest neighbor data assignment for following people.

3. A Cognitive Architecture for Social Robots

In this section, the cognitive architecture is described. The main contribution of this work, PIMS, is described in the next section. The description presented in this section is important to understand how it is integrated and applied in a real use case. The discussion and experimentation of this architecture can be found in dedicated works such as [44][35].

The cognitive architecture is designed in the form of layers. Each layer is called Tier N, where N is a number that indicates the level of abstraction of each one. Symbolic concepts are handled in Tier 1, while Tier 4 implements the skills that a robot must have (object detection, navigation, dialogue, etc.). These capabilities directly use the information from the sensors or send commands to actuators found in Tier 5. Figure 1 shows this concentric layer scheme. At the bottom of the figure is the knowledge graph, whereby the internal and external knowledge of the robot spreads between layers.

For the implementation of the proposed architecture, Behavior-based Iterative Component Architecture (BICA) [1] has been chosen, which is a toolbox to create software architectures for robots. Virtually all the elements of the design are BICA components that perform different functions. A BICA component is an independent process that can declare that it depends on other BICA components. When a BICA component is activated, it automatically activates all its dependencies. When all components that enable a dependency are deactivated, the dependence is deactivated. This mechanism is a simple way to save computation time when the results of certain computations are not being used.

[1] https://github.com/RoboCupAtHome/AtHomeCommunityWiki/wiki/Team-Description-Papers

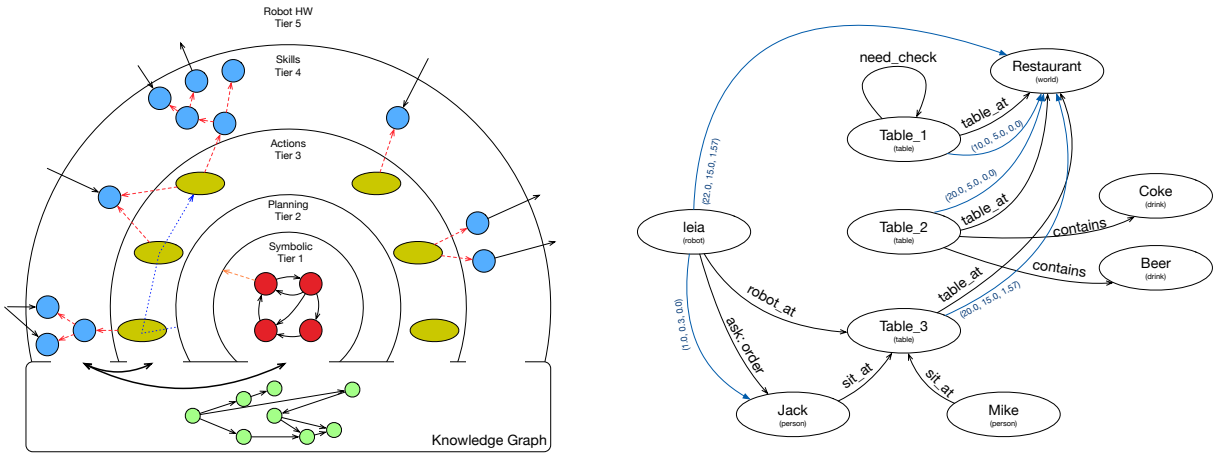


Fig. 1. Layered cognitive architecture (right) and an example of a Knowledge Graph (left). The cognitive architecture is composed of tiers, and generates the robot behavior from inner to outer tiers. The Knowledge Graph represents the internal and external knowledge of the robot. Ellipses represent nodes with an ID and a type. Lines are text and geometric arcs.

Tier 1 is the Mission Layer. It implements the operating modes or phases into which the robot's operation is divided. Its relationship with Tier 2, the planning layer, is very close. Both layers know the domain that defines the symbolic elements of the problem on which the robot establishes its action plans. Tier 1 can add instances in the problem and consult predicates, while its primary function is to set goals for the plans generated by Tier 2. In Figure 1, the red circles represent each of the states of a state machine, with its transitions. Each state machine is implemented within a BICA component, facilitating the work of the behavior developer.

Tier 2 implements a symbolic planner based on Planning Domain Definition Language (PDDL) [37]. This planner receives the goal established in any state of Tier 1 (orange arrow in Figure 1) and calculates a plan to achieve it. This sequence of actions is called a plan.

The planner is responsible for activating the implementation of the actions of which the plan is composed. These actions are implemented in Tier 3 and separate the symbolic world (Tier 1 and 2) from the sub-symbolic (Tier 4 and 4). Each action is executed one after the other in Tier 3. Each time an action is successful, the planner inserts the effects of the action into the knowledge base. When an action fails, the plan execution stops and the planner informs Tier 1, which can trigger a replanning.

In **Tier 3**, each of the actions defined symbolically in the PDDL domain maps to the implementation that carries it out. Actions receive their parameters as symbols (instances of a type). The action must typically translate symbols into specific data. For example, a move action could receive *kitchen* as a parameter. Then, the action must obtain the metric coordinate corresponding to the *kitchen* symbol and send it to the navigation module.

The actions are implemented as BICA components, declaring as execution dependencies skills in Tier 4, also implemented as BICA components. The actions may not require another BICA component but communicate directly with other modules (navigation, Human-Robot Interaction (HRI), etc.) or with the robot's sensors and actuators, which are in Tier 5. Actions can take a long time to finish their work, informing the planner when they end and if everything went correctly.

Actions do not usually implement all the functionality to carry out a task. These functionalities are implemented separately in skills, in **Tier 4**. An advantage is that one action could use several skills. In this layer, the robot may implement

perceptual abilities (detect people, read a text) or act (pay attention to an element or position, grab an object). Each one is implemented as a BICA component, which allows them to be activated from Tier 3, as dependencies of actions. In addition, some skills may be composed of the cascade execution of more than one BICA component, favoring the reuse of processes.

The knowledge graph stores the information relevant to the operation of the robot. This shared representation of data has been designed to disengage certain components from each other, especially between different layers. An action in Tier 3 uses the result of computing a skill in Tier 4 by reading it from the knowledge graph.

The elements of the graph are nodes and arcs. The nodes represent instances of a specific type. The arcs can contain a text, or can provide a geometric transformation.

The architecture described in this section is modular. For each application, the user defines which modules will be activated. Evidently, Tier 1 must be fully aware of the existing modules, as it must orchestrate them. Some modules may depend on other modules, so all components of both modules must be executed.

A module can contain:

- A PDDL domain, which provides new actions, types, and predicates to consider.
- The implementation (in Tier 2) for all the actions that this module provides in its PDDL model.
- All the skills needed by actions in Tier 3.

4. Description of Person Identification Memory System

The goal of the PIMS module is to identify people rapidly and accurately and learn and memorize new subjects on the fly. The techniques used to perform this task are a combination of deep learning architectures with traditional classifiers.

The deep learning architectures are used in order to generate the embeddings and features to recognize people, which perform much better than traditional approaches. However, they require a lot of time in the training stage, so they are trained offline and their weights will remain frozen on the live learning stage. In the case of traditional classifiers, their accuracy is poorer compared with the previous ones, but they require little time to be trained. Consequently, they can be retrained live and adjust their parameters to the new recognition requirements. The combination of both methods leverages their strengths and offsets their weaknesses.

The architecture of the proposal is shown in Figure 2. In the training stage the system receives images of the subject to be learnt, which moves in front of the camera with different postures. For every frame, the Detector locates the person with a bounding box inside the image. The Detector consists of a region-based Convolutional Neural Network capable of predicting the location of subjects in an image and returns the Area Of Interest (AOI) of every person. In this case, the architecture used is YOLO v3 [42]. Subsequently, the AOI is served as input to a modified Resnet50 [19]. This architecture is a state-of-the-art Convolutional Neural Network for classification tasks. In order to take advantage of the generated features, the fully connected layer at the end of the network, which is used for classification tasks, has been removed, so this network acts as a feature extractor. Therefore, the output of the network is a feature vector of 2048 values which is labeled with the person ID and sent to the classifier's model to be learnt. The label is known as this is performed at training time. Although YOLO extracted features could be processed directly, instead of applying another network's output, previous experiments show that using Resnet as feature extractor outperforms YOLO.

In the prediction stage, for every frame the Detector calculates the location of the people in the scene and generates the AOI. These AOI are then sent to the Feature Extractor, which calculates

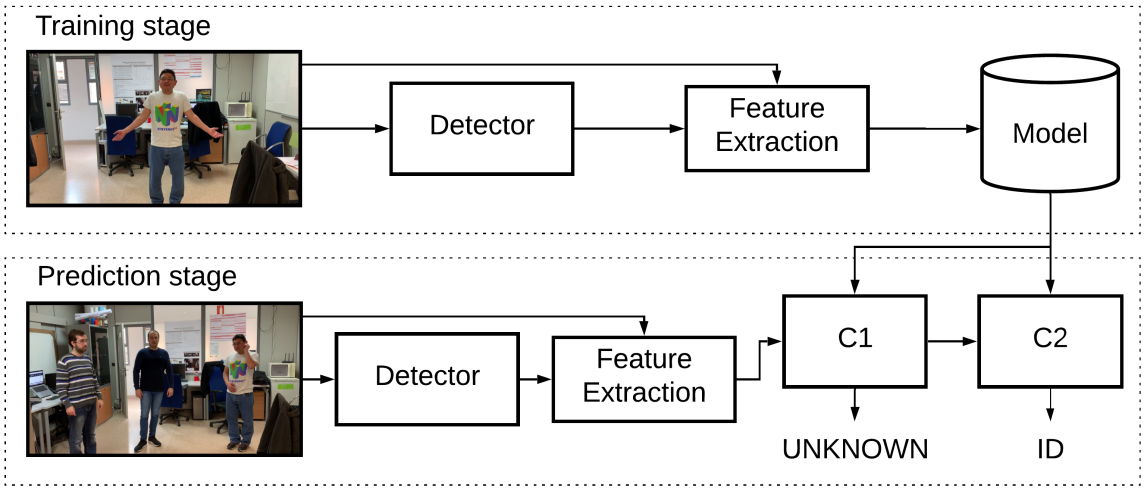


Fig. 2. Diagram of the proposal. At the training stage, the detector calculates the bounding box around the person, a deep learning network extracts their features and the vector representation is sent to the classifier's model, which is created by gathering labeled data. At the prediction stage, the detector and the feature extractor work as on the training stage, the C1 classifier rejects the subjects that are unknown and the C2 classifier distinguishes people's identities.

a vector representation for every subject. These features are forwarded to the C1 classifier, which distinguishes between known and unknown people. Finally, if the person has been classified as known, their features are sent to the C2 classifier, which recognizes their identity.

The need for a separate classifier that performs the differentiation between known and unknown people is a result of the difficulties in setting a proper distance threshold inside a class-instance classifier for this purpose. In the literature, there exists a family of methods, known as anomaly detectors [7], that carry out the differentiation between "normal" and "abnormal" data (in this case, normal data is known persons and abnormal data is unknown persons). In the semi-supervised case, they are capable of building their models with only "normal" data, which is perfectly suitable for this problem, as there is only data of known people. The hyperparameters of these classifiers can be optimized via automatic hyperparameter optimization such as random or grid search, both used in the machine learning suite *Scikit-learn* [25].

With this previous step, the final class-instance classifier will always receive a known subject and can perform the classification without applying any kind of filtering threshold. If the received person is unknown, they will be rejected in the first step.

As a result of the combination of deep learning and traditional classifiers, the PIMS approach trains rapidly as the training samples are inserted in traditional classifier models and there is no need to retrain the deep learning models which calculate the features. Moreover, its accuracy is high as it takes advantage of deep learning architectures for detection and feature extraction. Finally, the system can also learn new classes (unforeseen person IDs) without any architectural modifications. In contrast, a pure deep learning approach would require modifications on the last layer and a retraining process.

4.1. Integration in the Cognitive Architecture

In the proposed cognitive architecture, PIMS is a skill in Tier 4. However, integrating PIMS is not only determining what layer it is in but also,

from the rest of the elements of the architecture, how to use it.

Figure 3 shows how a robot learns and follows a person using PIMS integrated into the proposed architecture. It also shows a real example of one of the tests that a robot performs in the RoboCup competition. The robot begins in front of a person who acts as a guide. It has 10-20 seconds to learn the appearance of the guide. Once the learning phase is over, the robot must follow the guide in an environment where there are more people.

First, PIMS must be a BICA component, so it is active if there is any action in Tier 3 that activates it. The PIMS component must use the knowledge graph to interact with the other levels, especially with Tier 3.

If there is a self arc in the robot node that begins with the text `"learn_person:"` the module enters in learning mode. The module learns the features of the person in the center of the image. PIMS labels the person with the identifier of the rest of the arc text.

Once the `"learn_person"` arc disappears, it is in detection mode. If it detects the learned person again, it adds a person node with the identifier specified in the learning mode. It adds one arc, indicating that it sees the person. It also adds another arc with the position of the person with respect to the robot. If it detects another person, it adds a node with a generic identifier, and the corresponding arcs.

Two actions are defined in Tier 3: `learn_person` and `follow_person`. Both actions indicate that they require PIMS to be carried out.

The `learn_person action` receives a parameter, which is the identifier of the person to learn. Its only effect is to create the self arc `"learn_person:"` with the ID it receives as a parameter. Then PIMS enters in learning mode, as explained above.

The `follow_person` action receives a parameter, which is the identification of the person to follow. A successful detection creates a `"sees"` arc from the robot to a node with the specified identifier. Then, the action sends the commands to the motors to follow the person. If the "sees" arc does not exist, it can wait or turn to search for it.

In Tier 1, a state of the one finite state machine can be set as a goal `"person_learned ?p"` or `"person_followed ?p"`, which triggers the execution of a plan that includes the actions described above.

The complete PIMS module would include: 1) the PIMS component in Tier 4, 2) the two actions described above, and 3) the portion of PDDL that provides for both actions, the predicates that it requires and the person type.

5. Experimentation

In this section, the experimentation carried out to evaluate and validate the proposed approach is described. In addition, the details of the dataset used in the experiments are also reported.

The experiments were carried out using the following setup: Intel Core i5-3570 with 16 GiB of Kingston HyperX 1600 MHz and CL10 DDR3 RAM on an Asus P8H77-M PRO motherboard (Intel H77 chipset). The system also included an Nvidia GTX1080Ti, which was used for DL model inference. The framework of choice was Keras 1.2.0 with Tensor Flow 1.8 as the backend, running on Ubuntu 16.04. CUDA 9.0 and cuDNN v7.1 were also used to accelerate the computations. All the reported time measurements were made on this hardware.

5.1. Dataset

A custom dataset was recorded in order to test the proposed approach. This dataset was divided into two sets: training and test videos. The training set involved individuals standing in front of the camera and turning 360 degrees for 10 seconds. The test set consisted of different videos where the previous individuals moved around the

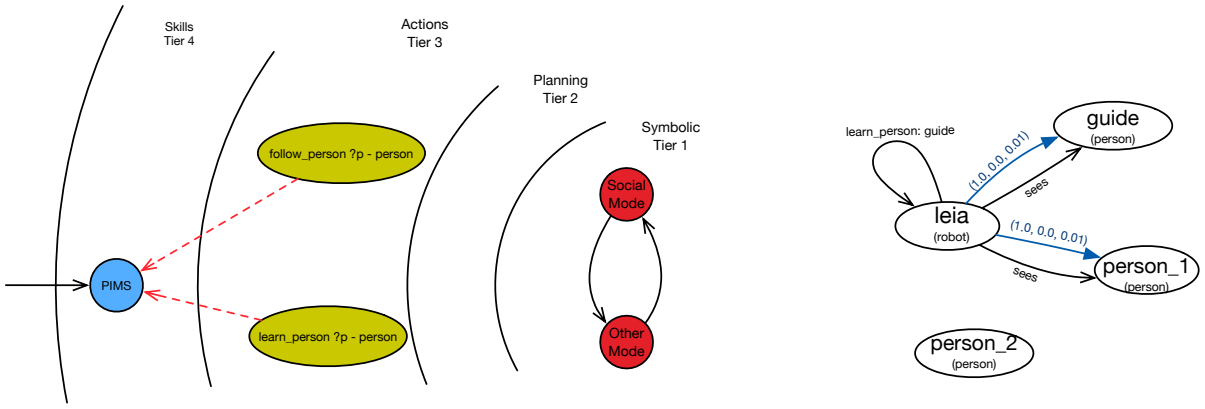


Fig. 3. Integration of PIMS in the cognitive architecture.

scene freely for 20 seconds. Finally, an additional video was recorded with three of the subjects walking around the room and interacting with one another. The last video was used for qualitative evaluation and the rest for quantitative benchmarking. The total size of the dataset was 9 videos, recorded by a 12 MPx color camera at 1080p resolution and 30 fps.

In the experiments described in the following sections, all four training videos were used to build the recognition models. Then, these models were used to perform inference on the test videos. As the test videos only showed one person, the system performance could be evaluated directly.

5.2. *RoboCup challenge*

The proposed problem to be solved was *Carry my Luggage [Party host]* from the Robucup@Home 2019 competition [43]. In this challenge, the robot must help the operator to carry some luggage outdoors.

First, the target person stands in front of the robot. The robot can give orders to move (turn round, move closer...) and takes pictures to recognize the person at that moment. Once the learning stage has concluded (20 seconds as maximum), the operator turns around and starts walking in different directions and crossing paths with other people. The robot must be able to follow its target even with occlusions and unknown people around the environment. Figure 4 shows a recreation of the challenge captured from a Pepper Robot.

5.3. *Person Identification System Experiments*

In this set of experiments, full body person identification was benchmarked. The person detector, which was based on YOLO, ran a model trained on the COCO MS [32] dataset as provided by the original author. This model was able to predict AOIs of different objects but as people were the subject of interest, the other predictions were ignored. In addition to the detector network, the ResNet50 trained on ImageNet was used as the AOIs feature extractor. VGG16[48] and MobileNet V2[46] were also tested as feature extractors. The ResNet50, VGG16 and MobileNet were trained on the ImageNet [12] dataset as provided by the Keras framework.

For the first experiment, the goal was to identify the best classifier that distinguishes between known people instances. The performance of different classifiers was tested using the features obtained by the previous network, considering that every sample was known by the system. The training video consisted of the individuals completely showing themselves. By sampling them at a fixed frame skipping value, the model was able to capture the features for each pose, thus leading to high accuracy rates and reducing the pro-

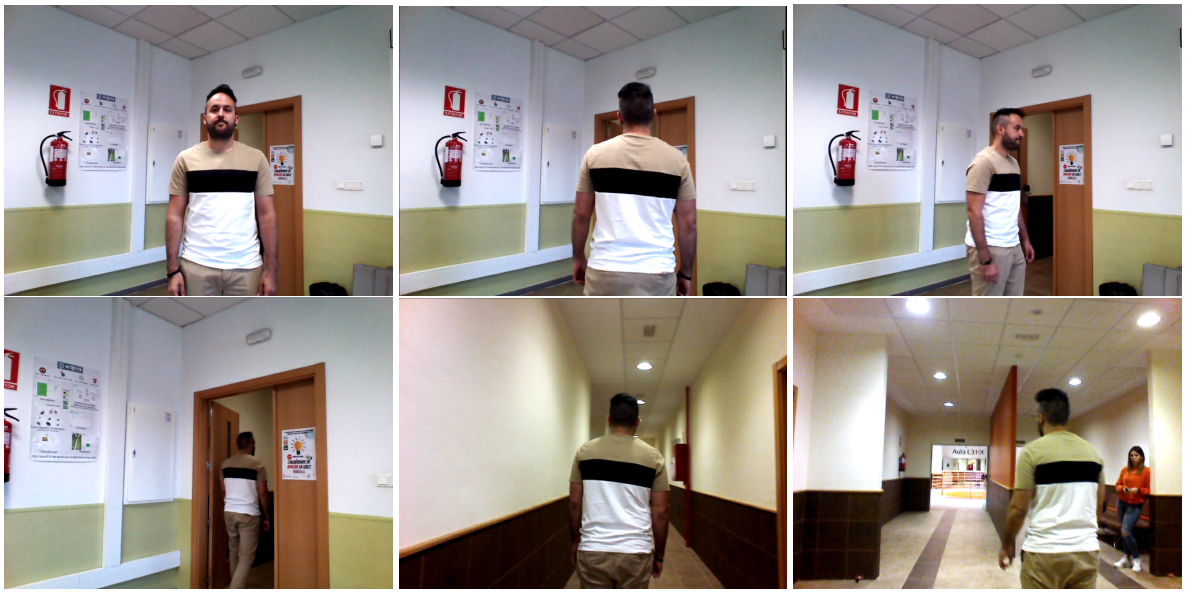
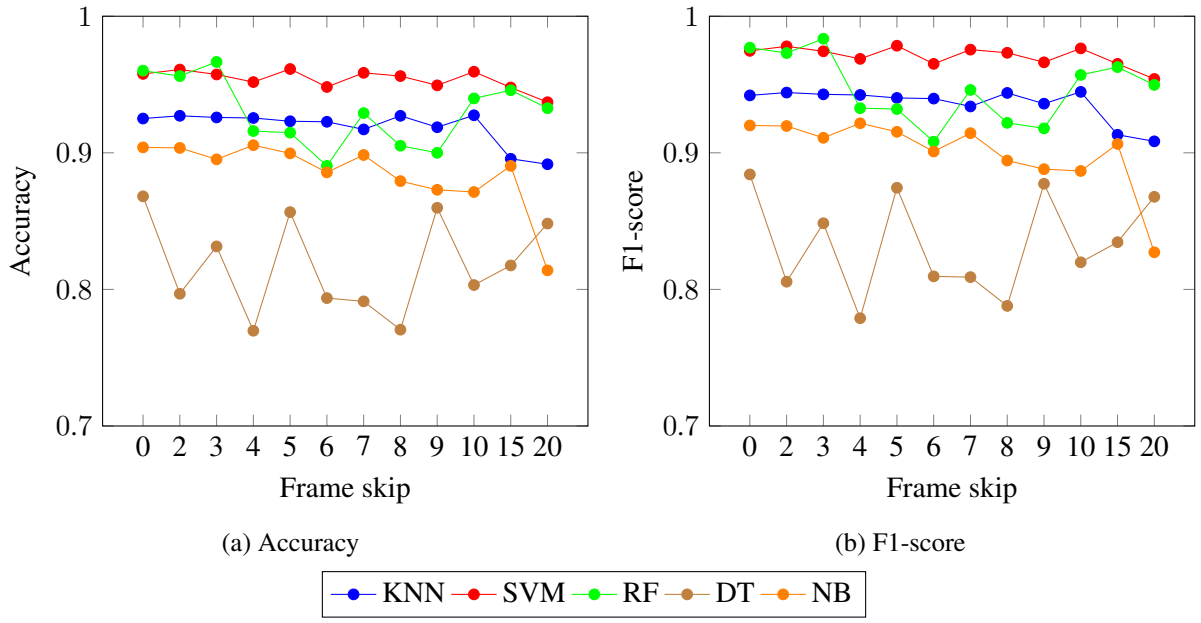


Fig. 4. Different moments of a recreation for the proposed challenge. The first 3 images correspond to the training stage and the last 3 correspond to the test stage.

cessing time. The proposed approach was tested ranging the frame skipping parameter from 0 (all frames are used) to 20 (around 65 frames remained from each training video). The classifiers trained are K-Nearest Neighbors (KNN), whose backbone yields 10 trees. At inference time, only the nearest neighbor is used; Support Vector Machine (SVM) with radial basis function and linear functions; and Random Forest (RF). As setting the correct parameters directly impacts the accuracy, C and *gamma* for the SVM were automatically computed by using Grid Search. The values we used were $C = [0.1, 1, 10, 100]$, $gamma = [1, 0.1, 0.01, 0.001, 0.00001, 10]$. The number of trees for the RF classifier were also automatically computed by Grid Search. In this case, $n = [20, 50, 100, 200, 500, 1000]$. The depth of each tree is computed by expanding all nodes until all leaves are pure or until all leaves contain less than 2 samples. Finally, Naive Bayes (NB) and Decision Trees (DT) were also involved as classifiers. These algorithms have no configurable parameters. We applied Grid Search when necessary because the parameters are dependent of the data, and different datasets could require different parameters to perform properly. Manually setting these parameters is not feasible because the robot should be able to set them in an unattended fashion. All the methods were set to multiclass classification. The accuracy rates are as shown in Figures 5a, 6a, 7a, 8a, 9a and 10a. In addition, F1-score for each experiment are also shown in Figures 5b, 6b, 7b, 8b, 9b and 10b.

Regarding our dataset, as depicted in Figure 11a, the SVM classifier outperformed the rest and the accuracy was maintained around 93% regardless of the frame skipping parameter. However, KNN and RF performed well, with an average precision of 90%. Despite the model containing fewer samples as the frame skipping increased, the accuracy was sufficiently high. This is because the model has enough semantic information in every case. It is worth noting that the model generated for frame skipping equal to 20 only contained 65 samples, around 16 for each individual. Random samples with the predictions superimposed are shown in Figure 12 for qualitative evaluation.

In addition, the proposal was tested with the Kinect Action Recognition Dataset (KARD)



(a) Accuracy (b) F1-score

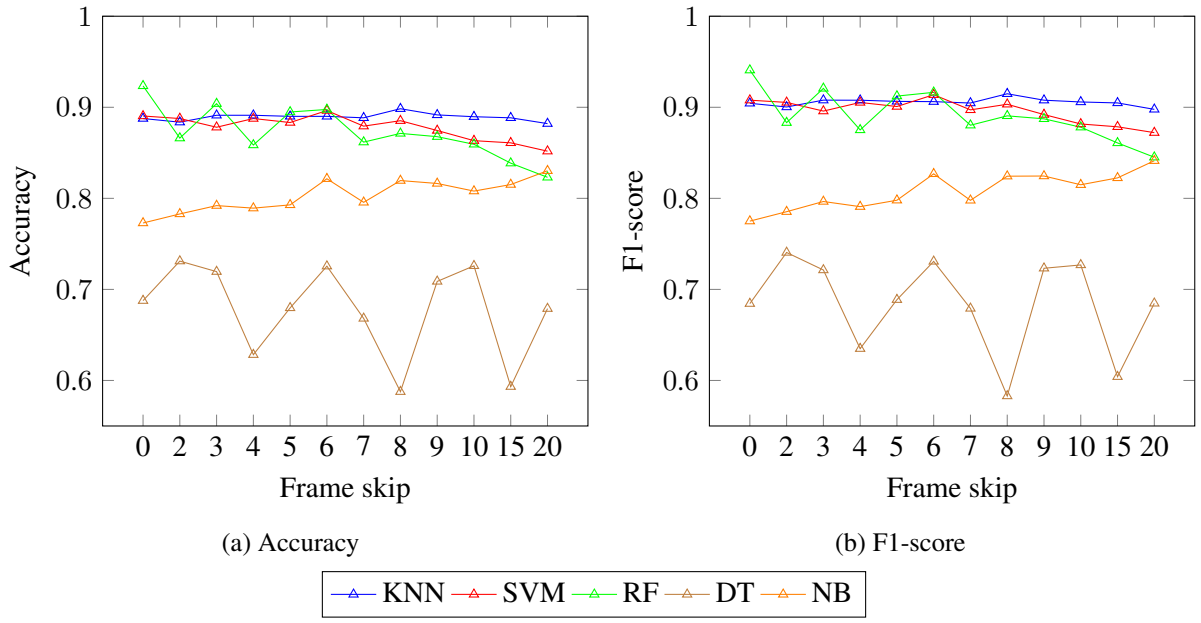
Fig. 5. Accuracy and F1-score results for OUR's testing videos using ResNet50 with frame skip from 0 to 20.

[15]. This dataset comprised 18 activities. Each activity was performed 3 times by 10 different subjects. In total, there were 540 videos. Despite this dataset being intended for action recognition tasks, it could be used to test the proposed approach as it had each person labeled independently. As there were a vast number of videos, only 5% of them were taken for training. The number of training frames was thus approximately the same as in the last experiment for each frameskip parameter, and so the results can be compared. The remaining 95% of the videos were used for testing purposes. It is worth noting that the range of different poses in the training set was limited as each video depicts just one action.

As the results show, the behavior of the classifiers is similar to that in the previous experiment. In this case, the overall accuracy increased slightly in every case despite this experiment having 10 different categories and the previous one only 4. The best performer in this case was also the SVM, which outperformed the KNN and the RF for every frameskip setting. Overall, it could be appreciated that the accuracy of RF decreased as the number of samples also decreased. This behaviour was also exhibited by the KNN, but the drop was not so considerable. The SVM performs similarly across all the experiments. DT and NB are far behind in terms of accuracy.

These conclusions can be extrapolated to the experiments that involved VGG16 as the backbone, as shown in Figure 11b. In this case, the trends remained the same but with a lower overall accuracy. This is for two main reasons: on the one hand, the feature vector it provides has 25088 parameters. Despite some works concluding that the number of parameters may not impact on the accuracy of the classifiers, in this case it definitely did. On the other hand, VGG16 provided lower classification accuracy than ResNet50 when it comes to the full convolutional network including the last fully connected layer, so the features were also likely poorer.

The experiments that used MobileNetV2 as the backbone also show the same trend. The overall accuracy was better than that of the VGG16 but poorer than the ResNet50 approach. This is because MobileNetV2 was purposely designed to be very fast to predict with, so the classification accuracy was lower than that of both the other



(a) Accuracy (b) F1-score

Fig. 6. Accuracy and F1-score results for OUR's testing videos using VGG16 with frame skip from 0 to 20.

mentioned networks. Nonetheless, as the number of features in its feature map was significantly smaller that the VGG16, its accuracy was better.

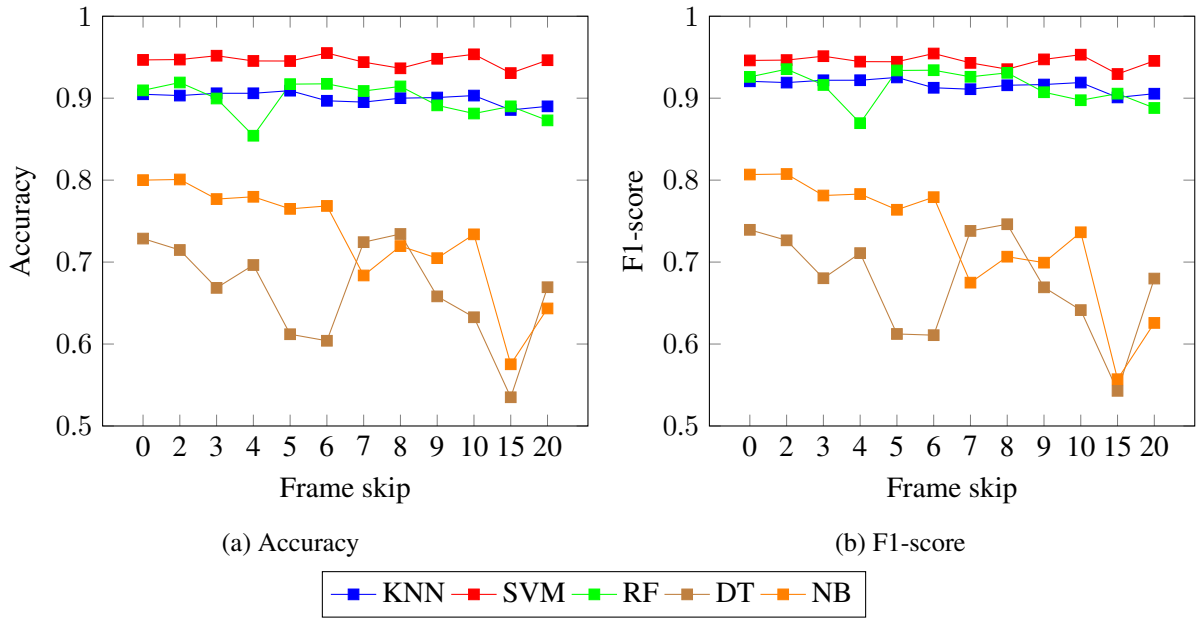
Figures 5b, 6b, 7b, 8b, 9b and 10b show the F1-score for each experiment. As can be seen, there is no bias towards a certain category.

To enable the system to perform in real environments, it should learn as fast as possible. With the goal of benchmarking this, the total time consumed to generate each model with the different number of frames was calculated. The results are shown in Figures 13a and 13b.

Although SVM and RF had better precision performances, they consumed a lot of time of training compared to KNN, which is almost immediate in every case (KNN is considered lazy learning indeed). 300 seconds for training (or 64 in the case of RF) takes as much time for a real application problem as for only 1200 frames. As expected, the approaches that involved VGG16 as the backbone took a vast amount of time to train because of the number of features. Regarding ResNet50 and MobileNetV2, the training times were similar, but MobileNetV2 was slightly faster. In addition, the training time grew as the problem became more complex. For instance, all the experiments that involved the KARD dataset took longer than the experiments on the proposed one. This was also expected as the KARD dataset features 10 different classes and the proposed one only 4.

In view of these results, it can be stated that the most suitable setting for online learning purposes involves ResNet50 or MobileNetV2 as the feature extractor and KNN as the final classifier. This is the fastest and most accurate setup. As the ResNet50 is slightly more accurate, it was selected for the following experiment.

For the second experiment, the ability of the proposed system to distinguish between known and unknown subjects was tested. To do this, a set of different classifiers (C1) to separate between these two classes was tested. If the person was classified as known, the KNN classifier (C2) from the previous experiment was used to perform the recognition. This was an important feature because the robot was likely to find new persons that have not as yet been considered by its model. The system should, thus, recognize when



(a) Accuracy (b) F1-score

Fig. 7. Accuracy and F1-score results for OUR's testing videos using MobileNetV2 with frame skip from 0 to 20.

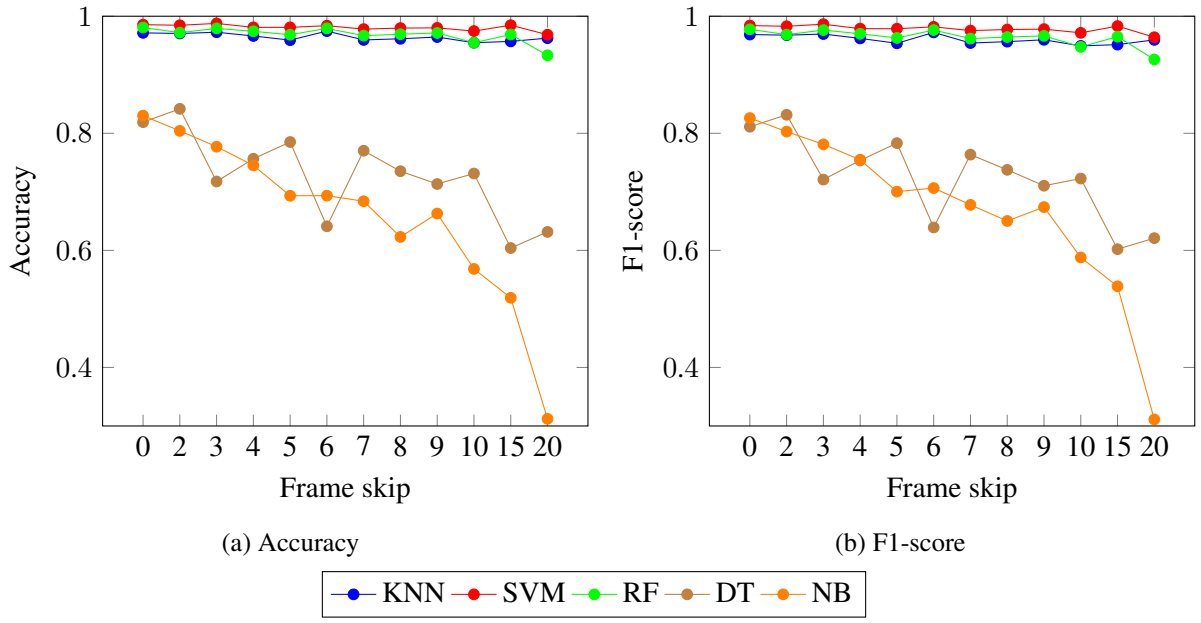
a person is new before trying to classify it into the identities it already knows.

In this experiment, the dataset described in Section 5.1 was used for training. The frameskip was set to 10 seconds with a frame skip of 10 (1 frame selected of every 10). The test set was made up of videos with some people from the training set and others that were not. The results are shown in Table 1. The ACK value represents the accuracy of the second classifier C2 after C1 has classified the person as known. The ACU value represents the accuracy of the first classifier C1 to classify unknown examples.

According to the results obtained, the best performance trade-off was achieved by the Clustering-based Local Outlier Detector because it shows high accuracy with known and unknown examples. In this case, the results for ACU show a very high accuracy for unknown people detection without losing too much accuracy on known examples. The training time for this first classifier was 1.21 seconds, so it can be retrained almost once per second and is suitable for a fast application. Another interesting choice would be the Local Outlier Factor. This method has lower accuracy classifying unknown examples (ACU) but the ACK value shows a better precision with known subjects. However, the notable difference in ACU value (13% lower compared with the previous method versus 5% higher in ACK) shows that the Clustering-based Local Outlier Detector performance is more balanced. Nevertheless, the model time of the second method was only 0.05 seconds, so it could be used if the time requirement gets tougher. The vast majority of the other classifiers show a biased classification, with a large number of the examples classified as known or unknown only, as can be seen in the ACU accuracy results. In Figure 14 a random sample is shown for qualitative evaluation of the whole system.

A video of this setup can be seen at [2]. First, the target subject was recorded and its features extracted (frameskip=10) and appended to both C1 (Clustering-based Local Outlier Detector) and C2 (KNN) models. The backbone was ResNet50.

[2] https://youtu.be/c1biTDNnLsg



(a) Accuracy (b) F1-score

Fig. 8. Accuracy and F1-score results for KARD's testing videos using ResNet50 with frame skip from 0 to 20.

Table 1
Results for the known-unknown classifiers and the whole recognition system

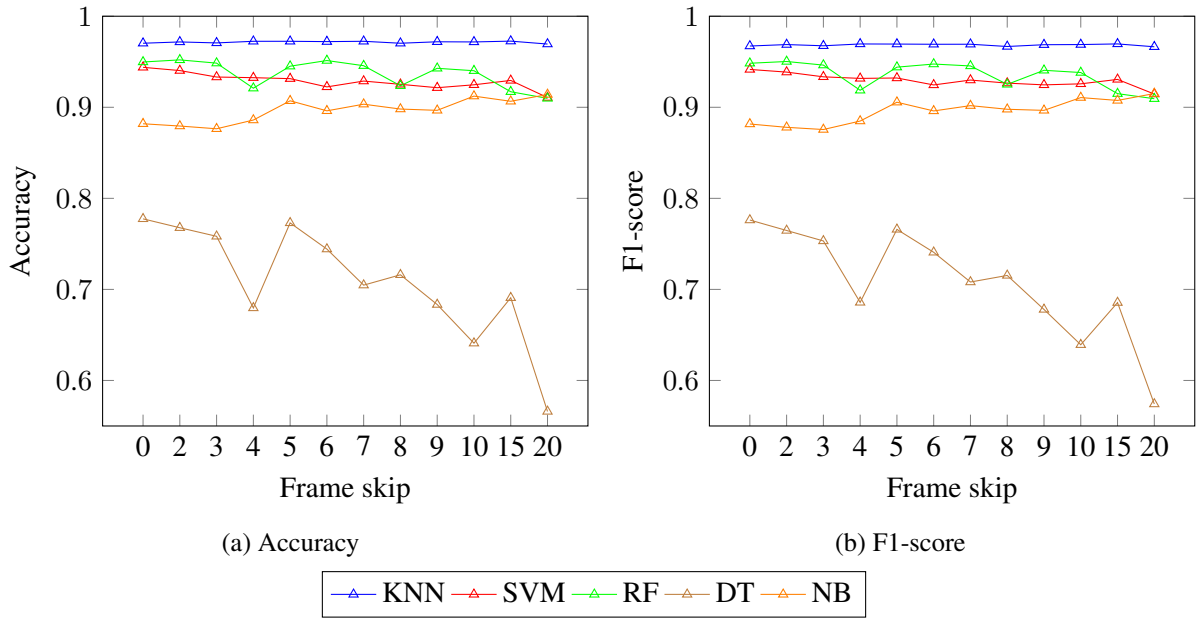
Algorithm	ACK (%)	ACU (%)	Model time (s)
Angle-based Outlier Detection [27]	84.86	37.87	0.95
Clustering-based Local Outlier Detector [20]	78.84	89.89	1.21
Histogram-based Outlier Detection [16]	91.27	0.45	1.96
Isolation Forest [33]	90.31	11.17	0.28
Local Outlier Factor [6]	83.30	76.36	0.05
Minimum Covariance Determinant [45]	40.00	99.99	97.63
One-class SVM [9]	86.37	16.11	0.03
Principal Components Analysis [55]	82.15	23.91	0.04
Stochastic Outlier Selection [23]	92.74	0.00	0.53

This setup was that previously worked best in terms of accuracy and training time. Then, the pipeline was used for tracking the target within a crowded street. The video was not edited at all, that is, between the training and the testing videos both models were trained live.

It is important to notice that the results obtained with the classifiers are aimed at showing how well they perform on the same features (the same DL networks), and not to state that other classifiers could not obtain similar results if they were applied with different configurations properly set.

5.4. Limitations

Despite the high accuracy in the test scenario, the proposed approach has some limitations. For instance, it is highly dependent on the visual features present in the training data. This means that if the person is not properly represented in the model, the system is likely to fail. This also makes the system fail under high occlusion sce-



(a) Accuracy (b) F1-score

Fig. 9. Accuracy and F1-score results for KARD's testing videos using VGG16 with frame skip from 0 to 20.

narios. Even if the AOI of the person is correctly detected, the visual features would depict the object occluding the person, leading to an eventual error. In addition, our approach is constrained to work on one camera and with specific conditions. For instance, our proposal could not be deployed for reidentification of the same person across different surveillance cameras because they would depict different points of view, usually show a large number of persons and feature low resolution and are very noisy.

6. Conclusion and Future Works

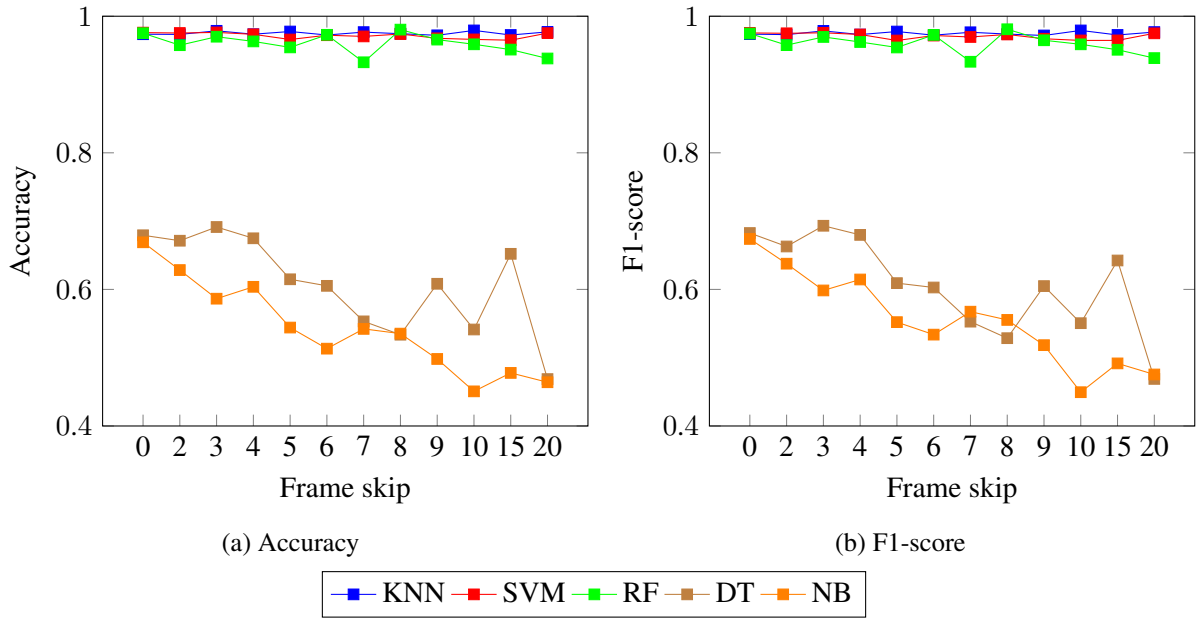
This work presents PIMS, a person identification system. Such a is critical in social robots since a robot can thus learn on the fly to recognize different people and adapt its behavior to each of them. Its impact on social applications is high and can be applied to interaction in highly populated environments or care applications.

This system performs person identification using a combination of deep learning and traditional methods which can learn fast and live. The crop and the features of every person are extracted with deep learning methods and are then classified with traditional techniques.

We have described a cognitive architecture to show how PIMS is used in a real robotic application. This integration was carried out by developing PIMS as a skill controlled by two actions that the robot can plan as part of its behavior.

Based on the experiments carried out, the proposed approach is able to correctly state the identification of a person in more than the 80% of the cases with only 10 seconds of training data, which perfectly suits the characteristics of the RoboCup challenge presented. Additionally, the results suggest that the accuracy of the model is independent of the number of samples. In fact, it is desirable to have a light model with different postures with a high variability rather than a lot of samples that are highly similar to one another.

Furthermore, as stated in the limitations section, the results suggest that this approach tends to fail with occluded bodies or with poses that differ from those used for training. The other possible problem is the detector not segmenting the person properly.



(a) Accuracy (b) F1-score

Fig. 10. Accuracy and F1-score results for KARD's testing videos using MobileNetV2 with frame skip from 0 to 20.

As a future work, it is planned to improve the PIMS accuracy by integrating a tracking method. For instance, if an identification in a certain moment differs from the last n predictions, it is likely a failure and could be corrected. In addition, face recognition must be included to complement and enhance the performance of the system. We also want to try newer classification algorithms, like [2, 41].

The source code to run some examples can be downloaded from [3].

Acknowledgements

This work has been funded by the Spanish Government TIN2016-76515-R grant for the COMBAHO project, supported with Feder funds. This work has also been supported by a Spanish grant for PhD studies ACIF/2017/243 and FPU16/00887. We are grateful to Nvidia for the generous donation of two Titan Xp and a Quadro P6000.

[3] https://bitbucket.org/rovitlib/person_learner

References

[1] C. E. Agüero, J. M. Canas, F. Martın, and E. Perdices. Behavior-based iterative component architecture for soccer applications with the nao humanoid. In *5th Workshop on Humanoids Soccer Robots. Nashville, TN, USA*, volume 127, 2010.

[2] M. Ahmadlou and H. Adeli. Enhanced probabilistic neural network with local decision circles: A robust classifier. *Integr. Comput.-Aided Eng.*, 17(3):197210, Aug. 2010. ISSN 1069-2509.

[3] E. Ahmed, M. Jones, and T. K. Marks. An improved deep learning architecture for person re-identification. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3908–3916, Los Alamitos, CA, USA, jun 2015. IEEE Computer Society. . URL https://doi.ieeecomputersociety.org/10.1109/CVPR.2015.7299016.

[4] E. Ahmed, M. J. Jones, and T. K. Marks. An improved deep learning architecture for person re-identification. In *CVPR*, pages 3908–3916. IEEE Computer Society, 2015. ISBN 978-1-4673-6964-0. URL http://dblp.uni-trier.de/db/conf/cvpr/cvpr2015.html#AhmedJM15.

[5] S.-H. Bae and K.-J. Yoon. Robust online multi-object tracking based on tracklet confidence and online discriminative appearance learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1218–1225, 2014.

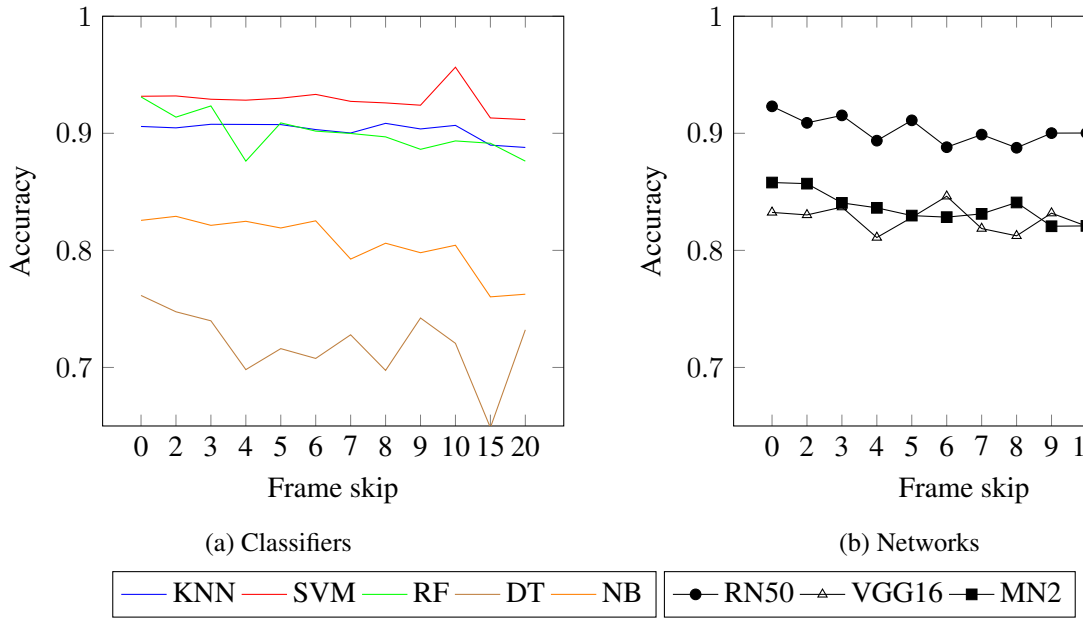


Fig. 11. Evolution of the mean accuracy results for the classifiers and DL networks. Tested using testing videos of OUR dataset with frame skip from 0 to 20.

Fig. 12. Random results of the proposed approach with the bounding box and the predicted identification of the person superimposed. Boxes and texts in green mean a hit (in this case, every prediction is correct). The model for these results was generated using the full duration of the training videos with no frame skip. Note that the identification is accurate even in unconsidered poses.

[6] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander. Lof: identifying density-based local outliers. In *ACM sigmod record*, volume 29, pages 93–104. ACM, 2000.

[7] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):15, 2009.

[8] K. Charalampous and A. Gasteratos. On-line deep learning method for action recognition. *Pattern Analysis and Applications*, 19(2):337–354, 2016.

[9] Y. Chen, X. S. Zhou, and T. S. Huang. One-class svm for learning in image retrieval. In *ICIP (1)*, pages 34–37. Citeseer, 2001.

[10] Y. Chen, X. Zhu, and S. Gong. Person re-identification by deep learning multi-scale representations. In *2017 IEEE International Conference on Computer Vision Workshops, ICCV Workshops 2017, Venice, Italy, October 22-29, 2017*, pages 2590–2600, 2017. . URL https://doi.org/10.1109/ICCVW.2017.304.

[11] S. Coradeschi and A. Saffiotti. An introduction to the anchoring problem. *Robotics and Autonomous Systems*, 43(2):85 – 96, 2003. ISSN 0921-8890. . URL http://www.sciencedirect.com/science/article/pii/S0921889003000216. Perceptual Anchoring: Anchoring Symbols to Sensor Data in Single and Multiple Robot Systems.

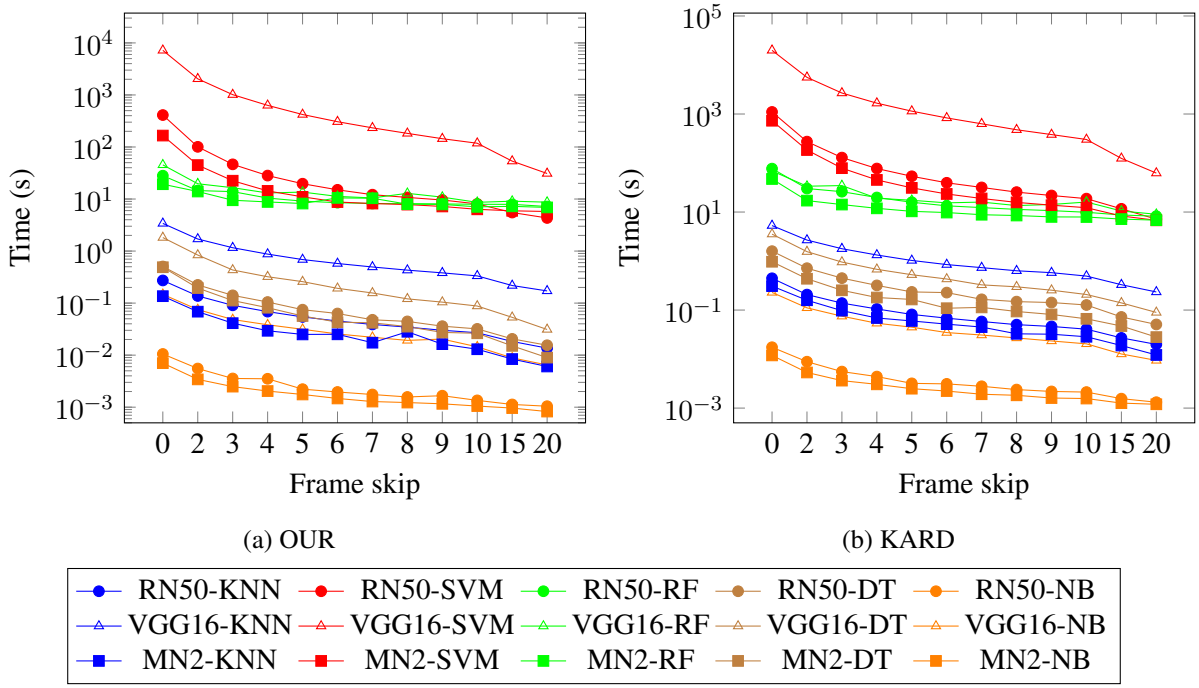


Fig. 13. Time consumed to generate the models from OUR's and KARD's training videos using frame skip from 0 to 20.

Fig. 14. Random results of the proposed approach with the bounding box and the predicted identification of the person superimposed. Boxes and texts in green mean a hit with known people. Blue boxes and texts mean they have been classified as unknown. In the first example, it can be seen that the body detector has identified a reflection on the glass as a person, but the proposed system has managed to identify it as unknown. The following examples show the potential problems of the system, the partial or total occlusion of the body. These examples will be discussed later in the Limitations section.

[12] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[13] W. Dodd and R. Gutierrez. The role of episodic memory and emotion in a cognitive robot. In *ROMAN 2005. IEEE International Workshop on Robot and Human Interactive Communication, 2005.*, pages 692–697, Aug 2005. .

[14] R. J. Duro, J. A. Becerra, J. Monroy, and F. Bellas. Perceptual generalization and context in a network memory inspired long-term memory for artificial cognition. *International Journal of Neural Systems*, 29 (06):1850053, 2019. . URL https://doi.org/10.1142/S0129065718500533. PMID: 30614325.

[15] S. Gaglio, G. L. Re, and M. Morana. Human activity recognition process using 3-d posture data. *IEEE Transactions on Human-Machine Systems*, 45(5):586–597, Oct 2015. .

[16] M. Goldstein and A. Dengel. Histogram-based outlier score (hbos): A fast unsupervised anomaly detection

algorithm. *KI-2012: Poster and Demo Track*, pages 59–63, 2012.

[17] S. Gong, M. Cristani, S. Yan, and C. C. Loy. *Person Re-Identification*. Springer Publishing Company, Incorporated, 2014. ISBN 1447162951, 9781447162957.

[18] D. Gray and H. Tao. Viewpoint invariant pedestrian recognition with an ensemble of localized features. In D. Forsyth, P. Torr, and A. Zisserman, editors, *Computer Vision – ECCV 2008*, pages 262–275, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[19] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. URL http://arxiv.org/abs/1512.03385.

[20] Z. He, X. Xu, and S. Deng. Discovering cluster-based local outliers. *Pattern Recognition Letters*, 24(9-10): 1641–1650, 2003.

[21] M. Hirzer, C. Beleznai, P. M. Roth, and H. Bischof. Person re-identification by descriptive and discriminative classification. In A. Heyden and F. Kahl, editors, *Image Analysis*, pages 91–102, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg. ISBN 978-3-642-21227-7.

[22] W. C. Ho, K. Dautenhahn, M. Y. Lim, P. A. Vargas, R. Aylett, and S. Enz. An initial memory model for virtual and robot companions supporting migration and long-term interaction. In *RO-MAN 2009 - The 18th IEEE International Symposium on Robot and Human Interactive Communication*, pages 277–284, Sep. 2009. .

[23] J. Janssens, F. Huszár, E. Postma, and H. van den Herik. Stochastic outlier selection. *tech. rep.*, 2012.

[24] G. Koch, R. Zemel, and R. Salakhutdinov. Siamese neural networks for one-shot image recognition. 2015.

[25] B. Komer, J. Bergstra, and C. Eliasmith. Hyperopt-sklearn: automatic hyperparameter configuration for scikit-learn. In *ICML workshop on AutoML*, volume 9. Citeseer, 2014.

[26] J. Krause, M. Stark, J. Deng, and L. Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the 2013 IEEE International Conference on Computer Vision Workshops*, ICCVW '13, pages 554–561, Washington, DC, USA, 2013. IEEE Computer Society. ISBN 978-1-4799-3022-7. . URL https://doi.org/10.1109/ICCVW.2013.77.

[27] H.-P. Kriegel, M. Schubert, and A. Zimek. Angle-based outlier detection in high-dimensional data. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 444–452. ACM, 2008.

[28] B. Lavi, M. F. Serj, and I. Ullah. Survey on deep learning techniques for person re-identification task. *CoRR*, abs/1807.05284, 2018. URL http://arxiv.org/abs/1807.05284.

[29] M. Li, F. Shen, J. Wang, C. Guan, and J. Tang. Person re-identification with activity prediction based on hierarchical spatial-temporal model. *Neurocomputing*, 275:1200 – 1207, 2018. ISSN 0925-2312. . URL http://www.sciencedirect.com/science/article/pii/S0925231217315837.

[30] M. Li, X. Zhu, and S. Gong. Unsupervised person re-identification by deep learning tracklet association. In V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, editors, *Computer Vision – ECCV 2018*, pages 772–788, Cham, 2018. Springer International Publishing.

[31] W. Li, R. Zhao, T. Xiao, and X. Wang. Deepreid: Deep filter pairing neural network for person re-identification. In *CVPR*, pages 152–159. IEEE Computer Society, 2014.

[32] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014. URL http://arxiv.org/abs/1405.0312.

[33] F. T. Liu, K. M. Ting, and Z.-H. Zhou. Isolation forest. In *2008 Eighth IEEE International Conference on Data Mining*, pages 413–422. IEEE, 2008.

[34] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. *Lecture Notes in Computer Science*, page 2137, 2016. ISSN 1611-3349. . URL http://dx.doi.org/10.1007/978-3-319-46448-0_2.

[35] F. Martín-Rico, J. Ginés, D. Vargas, F. J. Rodríguez-Lera, and V. Matellán-Olivera. Planning-centered architecture for robocup sspl @home. In R. Fuentetaja Pizán, Á. García Olaya, M. P. Sesmero Lorente, J. A. Iglesias Martínez, and A. Ledezma Espino, editors, *Advances in Physical Agents*, pages 287–302, Cham, 2019. Springer International Publishing. ISBN 978-3-319-99885-5.

[36] F. Martin-Rico, F. Gomez-Donoso, F. Escalona, M. Cazorla, and J. Garcia-Rodriguez. Artificial semantic memory with autonomous learning applied to social robots. In J. M. Ferrández Vicente, J. R. Álvarez-Sánchez, F. de la Paz López, J. Toledo Moreo, and H. Adeli, editors, *From Bioinspired Systems and Biomedical Applications to Machine Learning*, pages 401–411, Cham, 2019. Springer International Publishing. ISBN 978-3-030-19651-6.

[37] D. McDermott, M. Ghallab, A. Howe, C. Knoblock, A. Ram, M. Veloso, D. Weld, and D. Wilkins. Pddl-the planning domain definition language, 1998.

[38] M. Oliveira, G. H. Lim, L. S. Lopes, S. H. Kasaei, A. M. Tomé, and A. Chauhan. A perceptual memory system for grounding semantic representations in intelligent service robots. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2216–2223, Sep. 2014. .

[39] J. Phillips and D. Noelle. Biologically inspired working memory framework for robots. In *Proceedings of the 27th Annual Meeting of the Cognitive Science Society.*, pages 1237–1383, 2005.

[40] X. Qian, Y. Fu, T. Xiang, Y. Jiang, and X. Xue. Leader-based multi-scale attention deep architecture for person re-identification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2019. .

[41] M. H. Rafiei and H. Adeli. A new neural dynamic classification algorithm. *IEEE Transactions on Neural Networks and Learning Systems*, 28(12):3074–3083, Dec 2017. ISSN 2162-2388. .

[42] J. Redmon and A. Farhadi. Yolov3: An incremental improvement. *arXiv*, 2018.

[43] Robocup. *RoboCup@Home 2019*, 2019 (accessed October 24, 2019). URL https://athome.robocup.org/wp-content/uploads/Rulebook2019GermanOpen.pdf.

[44] F. J. Rodríguez-Lera, V. Matellán-Olivera, M. Á. Conde-González, and F. Martín-Rico. Himop: A three-component architecture to create more human-acceptable social-assistive robots. *Cognitive Processing*, 19(2):233–244, May 2018. ISSN 1612-4790. . URL https://doi.org/10.1007/s10339-017-0850-5.

[45] P. J. Rousseeuw and K. V. Driessen. A fast algorithm for the minimum covariance determinant estimator. *Technometrics*, 41(3):212–223, 1999.

[46] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen. Mobilenetv2: Inverted residuals and linear bottlenecks, 2018.

[47] Y. Shen, H. Li, S. Yi, D. Chen, and X. Wang. Person re-identification with deep similarity-guided graph neural network. In *ECCV (15)*, volume 11219 of *Lecture Notes in Computer Science*, pages 508–526. Springer, 2018.

[48] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[49] M. Skubic, D. Noelle, M. Wilkes, K. Kawamura, and J. M Keller. A biologically inspired adaptive working memory for robots. pages 1–8, 01 2004.

[50] D. Stachowicz and G. M. Kruijff. Episodic-like memory for cognitive robots. *IEEE Transactions on Autonomous Mental Development*, 4(1):1–16, March 2012. ISSN 1943-0604. .

[51] M. Tenorth and M. Beetz. Representations for robot knowledge in the knowrob framework. *Artificial Intelligence*, 247:151 – 169, 2017. ISSN 0004-3702. . URL http://www.sciencedirect.com/science/article/pii/S0004370215000843. Special Issue on AI and Robotics.

[52] C. Wang, Q. Zhang, C. Huang, W. Liu, and X. Wang. Mancs: A multi-task attentional network with curriculum sampling for person re-identification. In *ECCV*, 2018.

[53] K. Wang, H. Wang, M. Liu, and X. Xing. Survey on person re-identification based on deep learning. *CAAI Transactions on Intelligence Technology*, 06 2018. .

[54] T. Wang, S. Gong, X. Zhu, and S. Wang. Person re-identification by video ranking. In D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 688–703, Cham, 2014. Springer International Publishing. ISBN 978-3-319-10593-2.

[55] S. Wold, K. Esbensen, and P. Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.

[56] R. Yu, Z. Dou, S. Bai, Z. Zhang, Y. Xu, and X. Bai. Hard-aware point-to-set deep metric for person re-identification. In V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, editors, *Computer Vision – ECCV 2018*, pages 196–212, Cham, 2018. Springer International Publishing. ISBN 978-3-030-01270-0.

[57] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian. Scalable person re-identification: A benchmark. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1116–1124, Dec 2015. .

[58] L. Zheng, Z. Bie, Y. Sun, J. Wang, C. Su, S. Wang, and Q. Tian. Mars: A video benchmark for large-scale person re-identification. September 2016. URL https://www.microsoft.com/en-us/research/publication/mars-video-benchmark-large-scale-person-re-identification/.

[59] Z. Zheng, L. Zheng, and Y. Yang. Unlabeled samples generated by GAN improve the person re-identification baseline in vitro. *CoRR*, abs/1701.07717, 2017.