A System for Continuous Learning of Visual Concepts

Danijel Skočaj[1], Gregor Berginc[1], Barry Ridge[1], Aleš Štimec[1], Matjaž Jogan[1], Ondrej Vanek[1], Aleš Leonardis[1], Manuela Hutter[1], and Nick Hawes[2]

[1] University of Ljubljana, Faculty of Computer and Information Science, Slovenia
[2] University of Birmingham, School of Computer Science, United Kingdom

Abstract. We present an artificial cognitive system for learning visual concepts. It comprises of vision, communication and manipulation subsystems, which provide visual input, enable verbal and non-verbal communication with a tutor and allow interaction with a given scene. The main goal is to learn associations between automatically extracted visual features and words that describe the scene in an open-ended, continuous manner. In particular, we address the problem of cross-modal learning of visual properties and spatial relations. We introduce and analyse several learning modes requiring different levels of tutor supervision.

1   Introduction

In a real world environment, a cognitive system should possess the ability to learn and adapt in a continuous, open-ended, life-long fashion in an ever-changing environment. This learning is inherently cross-modal; the system should use all of its percepts and capabilities when trying to sense and understand the environment, and update the current knowledge accordingly. This learning should be performed efficiently via interaction with the environment and with other knowledgable cognitive systems (e.g., a tutor), which may facilitate the learning process and make it robust and reliable.

In this paper we present an artificial cognitive system for learning visual concepts that addresses the premises mentioned above. The main goal is to learn associations between automatically extracted visual features and words describing the scene in an open-ended, continuous manner. The continuous and multimodal nature of the problem demands careful system design. Our architecture is composed of vision, communication and manipulation subsystems, which provide visual input, enable verbal and non-verbal communication with a tutor and allow interaction with the scene. Such a multi-faceted active system provides means for efficient interaction with its environment facilitating user-friendly and continuous cross-modal learning.

In particular, we address the problem of learning visual properties (such as colour or shape) and spatial relations (such as 'to the left of' or 'far away'). The main goal is to find *associations* between *words* describing these concepts and simple *visual features* extracted from the images. *This symbol grounding prob-*

lem^3 is solved using a continuous learning paradigm in a cross-modal interaction between the system and the tutor. This interaction plays a crucial role in the entire learning process, since the tutor provides very reliable information about the scenes in question. This information can also be inferred by the system itself, reducing the need for tutor supervision, however also increasing the risk of false updates and degradation of the current knowledge. In this paper we introduce and analyse several different learning modes requiring different levels of tutor supervision.

Similar problems have often been addressed by researchers from various fields, from psychology, to computational linguistics, artificial intelligence, and computer as well as cognitive vision. Since the *symbol grounding problem (SGP)* was introduced by Harnad in 1990 [1], a plethora of papers have been published aiming to address it [2–7]. Our work is closely related to that of Roy [4, 5], in that our framework focuses on learning qualitative linguistic descriptions of visual object properties and scene descriptions. His system in [4] was designed to learn word forms and visual attributes from speech and video recordings, and subsequently, he extended this work for generating spoken descriptions of scenes [5]. The work of Chella *et al* [2, 3] contains further attempts at developing cognitive learning frameworks involving symbol grounding. Their system in [2] attempts to ground linguistic symbols in superquadric representations of scenes using neural networks. On the systems-oriented front, Bauckhage *et al* expound a framework for robot-human linguistic interaction in [8] that is similarly modular to ours.

Our framework however, while vying for similar goals to those of the above authors, differs significantly in two key respects: firstly, it performs continuous online learning and secondly, it employs multiple learning modes featuring varying degrees of tutor interaction. Moreover, the learning mode may be altered dynamically at any point during the learning process.

A fair amount of work exists on the subject of online learning [9–11], particularly with regard to object recognition, e.g., [9], where the authors have developed a biologically motivated feature-hierarchy based method that combines notions of short-term and long-term memory to achieve online learning of objects. The authors of [10] developed an online learning system for the AIBO robot that uses a similar dialogue setting for tutor interaction to ours here. In our system, we utilise online learning for learning qualitative object properties with a view toward using this as a basis for further learning.

The paper is organised as follows. In the next section we present our system and the individual modules. In Section 3 we propose a general framework for continuous learning involving different learning modes and a specific implemented method embedded into this framework. We then present the experimental results in Section 4. Finally, we summarise and outline some work in progress.

---

[3] Relating/connecting (linguistic) symbols to sub-symbolic interpretations of the physical world.

2 Integrated System

Any artificial cognitive system is by definition a compound of a number of components, including sensory components, communication sub-systems, processing modules, and possibly manipulation components. All of these components have to be tightly integrated in a unified system enabling the robust performance of the individual components and efficient communication between them to ensure synchronised and holistic functioning. Fig. 1(a) shows the setup of our system, while Fig. 1(b) depicts all of the components of our system and the connections between them in a schematic way. The dashed arrows indicate how requests are passed from module to module and the solid arrows indicate the flow of results (data). All of the modules are briefly described in the next subsection, followed by a subsection on the integration framework used to connect all of the components together in a unified system.
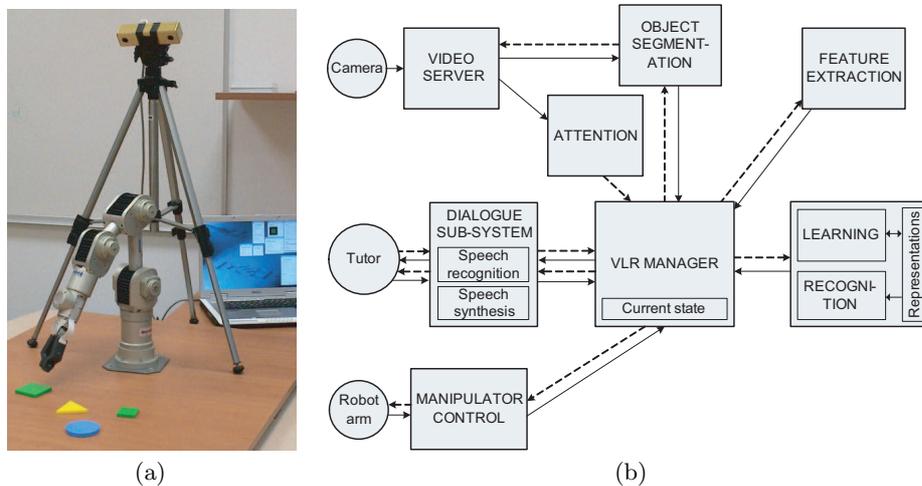

Fig. 1. (a) System setup. (b) System diagram.

2.1 System Modules

Video Server. This module provides visual input to the system. Images are retrieved from a V4L (Video4Linux) compliant video device and placed in a circular buffer with a preset number of frames. Frames are identified by unique timestamps which are pushed to other components when each frame is retrieved.

Attention. The attention module is used to detect changes in the scene. Every frame is pulled from the video server and compared with the previous one. When a substantial change in the image has been detected the attention module waits for the scene to settle down, and then notifies the system specifying the region of interest where the change has occurred.

Object Segmentation. This module serves for figure/background segmentation. Since the camera is static, it first learns the representation of the background and then uses this information for segmenting objects from the background. Each new object is stored along with its segmentation mask.

Feature Extraction. This module gets a part of the image (ROI) along with the corresponding segmentation mask and returns the features, which are then used for recognition and/or learning. In principle, the system could use any type of feature detectors; in the current implementation a few simple appearance, shape and distance features are extracted.

Learning and Recognition. The Learning and recognition module maintains the representations of all of the visual concepts that are being learned. Using the features extracted in the Feature extraction module, it is able to recognize the already learned visual concepts and to update the current representations. The quantitative results are then returned to the VLR manager, which processes them further. A more detailed description of the representations and the learning and recognition method is given in Section 3.

VLR Manager. The Visual learning and recognition manager is the central module in the system. It continuously monitors and waits for recognition/learning requests from the dialogue system and from the attention module. It then processes these requests (given in a symbolic form) and subsequently calls the corresponding modules. Afterwards it processes the obtained replies and again acts accordingly - whether it sends a request for forming a question or an answer to the dialogue system, or contacts the manipulator control to perform an action, or continues with the learning process. These decisions are made in accordance with the current state of the system and with the applied learning mode. A detailed description of different learning modes and actions that can be taken is given in Section 3.

Dialogue Subsystem. The dialogue subsystem serves as an interface between the system and the user, processing users spoken utterances and generating symbolic descriptions and vice-versa, producing sentences in natural language from symbols obtained from the VLR Manager. The speech recognition interface uses Sphinx 4, a pure Java and open source speech recognition framework. It uses a medium sized vocabulary that is part of the Sphinx 4 standard distribution. The grammar is defined in the Java Speech Grammar Format (JSGF). Since the speech recognition system is not completely reliable we also use a standard input with a keyboard and mouse as a backup solution when necessary. Robot responses are processed using the FreeTTS (Text-to-Speech) Java library and the generated sentences are then played back to the user in the natural language.

Manipulator Control. To also enable active manipulation, we included the Neuronics Katana Arm 6M180 manipulator in the system. It has five degrees of freedom and a gripper, and is able to handle various objects up to 500g in weight and 8cm in size. It is currently used to enrich the interaction between the user and the system. The robot arm is able to point at objects in the scene so as to resolve ambiguities and ease communication.

2.2 Integration Framework

To facilitate communication between most of the components we use the process communication framework BALT [12]. For cross-language compatibility, the

toolkit supports components written in the Java and C++ languages. To hide communication details from the end-user, the toolkit uses CORBA (Common Object Request Broker Architecture), as its underlying communication architecture, thus making it possible for the components to communicate over any TCP/IP network. The system can thus be distributed over several computers. Components running on the same machine can share data using shared memory.

BALT components can be connected using two types of connections, namely push and pull. If the provider of certain data needs to notify other components, the push connection is used. An example of this is the video server component that pushes frame timestamps on every frame change to registered components. Pull connections are used by components to query providers for data. As an example, the attention module uses a pull connection to the video server to get a new frame when the timestamp is received.

3   Continuous Learning Framework

At the heart of the system is a continuous learning framework that processes requests, performs recognition, and updates the representations according to the current learning mode. In this section we define several learning modes which alter the behaviour of the system and require different levels of tutor involvement.

When implementing a continuous learning mechanism, two main issues have to be addressed. Firstly, the representation, which is used for modeling the observed world, has to allow for updates when presented with newly acquired information. This update step should be efficient and should not require access to previously observed data, while still preserving the previously acquired knowledge. Secondly, a crucial issue is the quality of the updating, which highly depends on the correctness of the interpretation of the current visual input. With this in mind, several learning strategies can be used, ranging from completely supervised to completely unsupervised. Here we discuss three such strategies:

- **Tutor-driven approach (*TD*).** The correct interpretation of the visual input is always correctly given by the tutor.
- **Tutor-supervised approach (*TS*).** The system tries to interpret the visual input. If it succeeds to do this reliably, it updates the current model, otherwise asks the tutor for the correct interpretation.
- **Exploratory approach (*EX*).** The system updates the model with the automatically obtained interpretation of the visual input. No intervention from the tutor is provided.

We further divide *tutor-supervised learning* into two sub-approaches:

- **Conservative approach (*TSc*).** The system asks the tutor for the correct interpretation of the visual input whenever it is not completely sure that its interpretation is correct.
- **Liberal approach (*TSl*).** The system relies on its recognition capabilities and asks the tutor only when its recognition is very unreliable.

Similarly, we also allow for **conservative** and **liberal exploratory** sub-approaches (**EXc**, **EXl**).

To formalise the above descriptions, let us assume that the recognition algorithm always gives one of the following five answers when asked to confirm the interpretation of the visual scene (e.g., the question may be: "Is this circular?"): *'yes' (YES), 'probably yes' (PY), 'probably no' (PN), 'no' (NO),* and *'don't know' (DK)*. Table 1 presents actions that are taken after an answer is obtained. The system can either *ask* the tutor for the correct interpretation of the scene (or the tutor provides it without being asked), *update* the model with its interpretation, or do nothing. As is seen in Table 1, the system can communicate with the tutor all of the time (*TD* learning), often (*TSc*), occasionally (*TSl*) or even never (*EX* learning). This communication is only initiated by the tutor in the tutor-driven approach, while in other approaches the dialogue and/or the learning process is triggered by the system itself.

Table 1. Update table.

	YES	PY	PN	NO	DK
TD	ask	ask	ask	ask	ask
TSc	update	ask	ask	/	ask
TSl	update	update	/	/	ask
EXc	update	/	/	/	/
EXl	update	update	/	/	/

To speed up the initial phase of the learning process and to enable development of consistent basic concepts, one could start with mainly tutor-driven learning with many user interactions. These concepts would then be used to detect new concepts with limited help from the user. Later on in the process, when the ontology is sufficiently large, many new concepts could be acquired with limited or even without user interaction.

An important part of such a framework is an update algorithm, which is able to continuously update representations of visual concepts being learned, and a recognition algorithm, which is able to query these representations and produce quantitative answers. I.e., the main task of these algorithms is to assign associations between extracted visual features and the corresponding visual concepts (visual attributes or spatial relations). It has to consider two main issues: *consistency* and *specificity*. It must determine which automatically extracted visual features are *consistent* over all images representing the same visual concept and that are, at the same time, *specific* for that visual concept only. Note that this process should be performed incrementally, considering only the current image and learned representations – previously processed images cannot be re-analysed.

In principle, any method for incremental visual learning and recognition that fulfills the above mentioned requirements could be used. In our system we use algorithms based on a generative representation of extracted features associated with visual concepts. Each visual concept is associated with a visual feature that best models the corresponding images according to the consistency and specificity criteria mentioned above. The learning algorithm thus selects the extracted feature whose values are most consistent over all images representing the same visual concept (i.e. the variance is small and the feature values are

concentrated around the mean value). At the same time it also ensures that the same does not hold true for some other visual concepts, thus satisfying the specificity criterion. A visual concept is therefore represented with the mean and variance of the best feature [13] and this representation gets continuously updated over time according to the selected learning mode.

4 Experimental Results

The cognitive vision system presented in this paper was primarily designed to work in interaction with a user; it includes both a communication sub-system and a manipulation module that ease tutor-system communication. Fig. 2 shows a sample dialogue between the tutor and the system, which indicates the process of learning of object visual properties (colour, size, and shape). The first part of the dialogue took place in the tutor-driven learning mode, when the tutor was teaching the system about the objects in the scene, while the second part of the dialogue took place using the tutor-supervised modes of learning, when the system took the initiative and asked the tutor for clarification when needed. In other experiments, the system may also have used the robot arm to point at the object of interest when many objects were present. The communication between the tutor and the system is thus performed in a very natural and user friendly way.

```
'H: This is a blue, large, and square object.'      ...
'R: OK.'                                             'R: I don't know this object. What is it like?'
                                                     'H: It is blue, small, and rectangular.'
...                                                  'R: OK.'
'H: What do you see?'
'R: I don't know this object.'                       ...
'H: It is green, small, and triangular.'             'R: Is this object large?'
'R: OK.'                                             'H: Yes, that's correct.'
                                                     'R: OK.'
...                                                  'H: It is also yellow and circular.'
'H: What do you see?'                                'R: OK.'
'R: I see a small and square object.'
'H: Yes, that's correct.'                            ...
'R: OK.'                                             'R: Is this object red?'
                                                     'H: No, it is not red.'
...                                                  'R: OK.'
'H: What do you see?'                                'H: It is yellow.'
'R:I see a green object. I think it is also large.'  'R: OK.'
'H: Yes, that's correct.'
'R: OK.'                                             ...
'H: It is also rectangular.'                         'H: What do you see?'
'R: OK.'                                             'R: I see a red, small, and triangular object.'
```

Fig. 2. An example dialogue between the tutor (H) and the system (R).

However, to comprehensively analyse the proposed learning modes, such interactive work is time consuming and impractical. Therefore, we instead performed experiments on images with known ground truth, and simulated the answers of the tutor by replacing the dialogue system. In this way the extensive tests could be automatically performed and a reliable evaluation of the proposed methods were obtained.

We tested the algorithms by running a number of experiments on both artificial and real data. Basic shapes of various different colours and sizes were

selected as test objects. Some of them are depicted in Fig. 3(a). We considered three visual attributes (colour, size and shape), and ten values of these visual attributes altogether (red, green, blue, yellow; small, large; square, circular, triangular, and rectangular).

The objects were first perspective-rectified and segmented from the background. Then the visual features were extracted. We used six simple one-dimensional features; three colour features (median of hue, saturation and intensity over all pixels in the segmented region) and three simple shape descriptors (area, perimeter and compactness of the region). The main goal was to find associations between ten given attribute values and six extracted features.

We put half of the images in the training set and other half in the test set and kept incrementally updating the representations with the training images using different learning strategies. At each step, we evaluated the current knowledge by recognising the visual properties of all test images. The evaluation measure we used is *recognition score*, which rewards successful recognition (true positives and true negatives) and penalises incorrectly recognised visual properties (false positives and false negatives).

The results (the curves of the evolution of the recognition score through time) of the experiment on the synthetic images (averaged over 40 trials on different sets of generated images with added noise) are presented in Fig. 3(c). All different learning strategies presented in Section 3 were tested. First, we applied the various learning modes starting with one training image from the beginning of each run (denoted as *TSc1*, *TSl1*, etc.). After that we repeated the experiment by first applying the tutor driven mode (*TD*) to the first 10 images, and then continuing by incrementally adding the rest of the images using other approaches (*TSc10*, *TSl10*, etc.). Fig. 3(c) shows the plots of recognition scores.

The tutor-driven learning successfully associates the colours of the input objects with the *hue* feature, their sizes with the *area* feature and their shapes with the *compactness* feature. Recognition of visual attributes is very successful; it almost gets the maximal score (640 in this case). However, the tutor has to provide all information (about 10 visual attributes) to the system at every step. Tutor-supervised learning proved to be quite successful as well. In this case conservative strategy yields better results, since it asks the tutor for reliable information more often. In the beginning the system does not have a lot of knowledge, so the tutor is asked for help more frequently. After the knowledge is acquired, the number of questions decreases (from 10 at the beginning to 2 after 20 updates). The explorative approach, which does not involve interaction with the tutor, does not significantly improve the model. So, as expected, there is a trade-off between the quality of the results and the autonomy of the system. Similar conclusions can also be drawn from the results of the experiment on real data shown in Fig 3(d).

Exactly the same system was also used for learning simple spatial relations. We only changed the features that were to be extracted from the image. In this case we used five distance features – horizontal and vertical position of the object in the scene, absolute differences in the horizontal and vertical positions

of two objects, and euclidian distance between them, when two objects were present in the scene. Using these five features, the learning framework was able to learn eleven spatial relationships (six binary relations between two objects (with respect to the observer/camera): 'to the left of', 'to the right of', 'closer than', 'further away then', 'near to', 'far from', and five unary relations describing the position of the object in the scene: 'on the left', 'in the middle', 'on the right', 'near', and 'far away'). The correctly assigned associations, along with the previously learned visual attributes, enabled the automatic detection of objects and the production of scene descriptions such as those presented in Fig. 3(b).
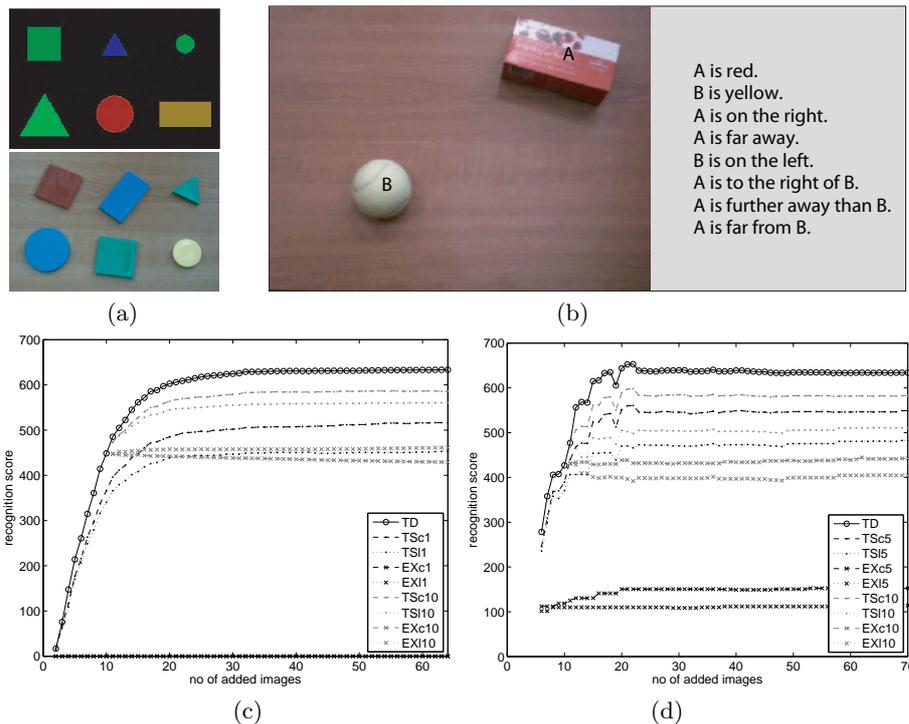

(a)

A is red.
B is yellow.
A is on the right.
A is far away.
B is on the left.
A is to the right of B.
A is further away than B.
A is far from B.

(b)

(c)

(d)

Fig. 3. (a) Synthetic shapes and real objects of similar shape. (b) Automatically obtained scene description. (c) Recognition score on synthetic images. (d) Recognition score on real images.

5 Conclusion

In this paper we presented an artificial cognitive system for continuous learning of visual concepts. It comprises of vision, communication and manipulation sub-systems and it is based on a framework for continuous learning that enables three modes of learning requiring different levels of tutor supervision. We experimentally evaluated these three learning strategies and concluded that the learning

process should start with tutor-driven learning to enable development of consistent basic concepts, which could be updated later on in a tutor-supervised way requiring fewer tutor interventions.

Beyond this work, we aim to improve the learning method as well as to further analyse the proposed framework and evaluate different learning strategies under various conditions and in various applications. We will employ the robot arm for more advanced tasks, so that the system will actually be able to actively plan and perform actions and explore effects of the actions on objects, thus learning the object affordances as well. We thus aim to develop an even more general system for continuous learning that is capable of extending its ontology with other types of visual concepts as well.

References

1. Harnad, S.: The symbol grounding problem. Physica D: Nonlinear Phenomena **42** (1990) 335–346
2. Ardizzone, E., Chella, A., Frixione, M., Gaglio, S.: Integrating subsymbolic and symbolic processing in artificial vision. J. of Intell. Systems **1(4)** (1992) 273–308
3. Chella, A., Frixione, M., Gaglio, S.: A cognitive architecture for artificial vision. Artificial Intelligence **89**(1–2) (1997) 73–111
4. Roy, D.K., Pentland, A.P.: Learning words from sights and sounds: a computational model. Cognitive Science **26**(1) (2002) 113–146
5. Roy, D.K.: Learning visually-grounded words and syntax for a scene description task. Computer Speech and Language **16(3)** (2002) 353–385
6. Steels, L., Vogt, P.: Grounding adaptive language games in robotic agents. In: Proceedings of the ECAL'97. Complex Adaptive Systems (1997) 474–482
7. Vogt, P.: The physical symbol grounding problem. Cognitive Systems Research **3**(3) (2002) 429–457
8. Bauckhage, C., Fink, G.A., Fritsch, J., Kummert, F., Lömker, F., Sagerer, G., Wachsmuth, S.: An Integrated System for Cooperative Man-Machine Interaction. In: IEEE International Symposium on Computational Intelligence in Robotics and Automation, Banff, Canada (2001) 328–333
9. Kirstein, S., Wersing, H., Körner, E. Rapid online learning of objects in a biologically motivated recognition architecture In: 27th DAGM (2005) 301–308
10. Steels, L., Kaplan,F. AIBO's first words. The social learning of language and meaning. Evolution of Communication **4(1)** (2001) 3–32
11. Arsenio, A.M. Developmental Learning on a Humanoid Robot In: IEEE International Joint Conference On Neural Networks **4** (2004) 3167–3172
12. Hawes, N.: BALT & CAAT: Middleware for cognitive robotics. Technical Report CSR-07-1, University of Birmingham, School of Computer Science (2007)
13. Skočaj, D., Ridge, B., Leonardis, A.: On different modes of continuous learning of visual properties. In: ERK 2005, Portorož, Slovenia (2006) 105–108