Computer Vision Winter Workshop 2007, Michael Grabner, Helmut Grabner (eds.)
St. Lambrecht, Austria, February 6–8
Graz Technical University

A Framework for Continuous Learning of Simple Visual Concepts

Danijel Skočaj, Barry Ridge, Gregor Berginc, and Aleš Leonardis

University of Ljubljana, Faculty of Computer and Information Science
Tržaška 25, SI-1001 Ljubljana, Slovenia
danijel.skocaj@fri.uni-lj.si

Abstract *We present a continuous learning framework for learning simple visual concepts and its implementation in an artificial cognitive system. 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 elementary visual properties and spatial relations; we show that the same learning mechanism can be used to learn both types of concepts. We introduce and analyse several learning modes requiring different levels of tutor supervision, ranging from a completely tutor driven to a completely autonomous exploratory approach.*

1 Introduction

In the real world, a cognitive system should possess the ability to learn and adapt in a continuous, open-ended, life-long fashion in an ever-changing environment. As an example of such a learning framework, we need look no further than at the successful application of *continuous learning* in human beings. As humans, we first learn a new visual concept (e.g., an object category, an object property, an action pattern, an object affordance, etc.) by encountering a few examples of one. Later, as we come across more instances different to the original examples, we not only recognise them, but also update our representation of learned visual concepts, based on the salient properties of the new examples and without having visual access to the previous examples. In this way, we update or enlarge our ontology in an efficient and structured way by encapsulating new information extracted from the perceived data, which enables adaptation to new visual inputs and the handling of novel situations we may encounter.

While the primary focus of this idea is on the incremental nature of the knowledge update, another key aspect should be noted; that being the scrutinisation of various visual features and the determination of which features are useful for representing the chief visual attributes of the object or scene in question. Since a continuous learning framework would not retain complete data from previously learned samples, it would not have the luxury of being able to reference specific details across multiple samples in order to learn. Given this restriction, continuous learning lends itself to an abstract multi-modal system involving interaction with a user.

In this paper we present a framework for learning simple 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 implemented system is composed of vision and communication subsystems providing the visual input and enabling verbal dialogue with a tutor. Such a multi-faceted active system provides means for efficient communication 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 problem*[1] 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. As the main contribution, 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 [5], a plethora of papers have been published aiming to address it [1, 2, 3, 4, 10, 9, 11, 14]. Harnad proposed a hybrid model [5] as a means of solving the SGP that would mix the useful elements of both symbolic and connectionist systems by connecting the symbols manipulated by an autonomous agent to the perceptual data they denote. This formed the basis of further analyses by Mayo [7] and Sun [12] in a similar vein. Subsequently, a number of authors re-analysed the problem [14, 8] and attempted to extend the hybrid model in various directions. In particular, Davidsson's 1993 study of symbol grounding [3] emphasises the importance of incremental learning for concept formation and grounding of concepts, a methodology which we explicitly conform to

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

here.

Our work is closely related to that of Roy [10, 9], in that our framework focuses on learning qualitative linguistic descriptions of visual object properties and scene descriptions. Roy and Pentland's system in [10] was designed to learn word forms and visual attributes from speech and video recordings, and subsequently, Roy extended this work for generating spoken descriptions of scenes [9]. The work of Chella *et al* [1, 2] contains further attempts at developing cognitive learning frameworks involving symbol grounding. Their work is based on Gärdenfors' paradigm of three levels of inductive inference [4], and their implementation of this paradigm in [1] involves grounding linguistic symbols in superquadric representations of scenes using neural networks.

Our framework however, while vying for similar goals to those of the above authors, differs significantly in two key respects: firstly, it performs continuous 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 continuous learning process.

The paper is organised as follows. In the next section we propose a general framework for continuous learning involving different learning modes. In Section 3 we present a specific method for incremental learning and embed it in the proposed framework. We then present a practical implementation of the proposed framework in Section 4, followed by the experimental results, and an evaluation and discussion of the proposed learning mechanisms in Section 5. Finally, we summarise and outline some work in progress.

2 Continuous learning framework

The interaction between a tutor and an artificial cognitive system plays an important role in a continuous learning framework. The goal of the learning mechanism is to find associations between words spoken by the tutor and visual features automatically extracted by the cognitive visual system, i.e., to ground the semantic meaning of the visual objects and their properties into the visual features. Such a continuous learning framework should process requests, perform recognition, and update 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 from the recognition process. 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 it is stated 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 initiated 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 without user interaction.

3 Learning algorithm

Important parts of such a framework are 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 (e.g., 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 (or a very recent set of images) 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 [13] of extracted features associated with visual concepts. Each visual concept is associated with an extracted visual feature that best models the corresponding images according to the consistency and specificity criteria mentioned above. The learning algorithm thus selects from N_F one-dimensional features (e.g., median hue value, area of segmented region, etc.), the feature whose values are most consistent over all images representing the same visual concept (e.g., all images of large objects, or circular objects, etc.), thus the variance is small and the extracted 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 concept, thus satisfying the specificity criterion. With this in mind, we represent a visual concept using the mean and variance of the best feature.

The main idea is described in an algorithmic form in Algorithm 1. In the basic batch form, the algorithm would require all training images to be given in advance, together with a list of visual concepts (e.g., red, large, square) for each image. Since the mean and variance of a set of feature values can be calculated in an incremental way without losing any information, this algorithm can be incrementalised. Algorithm 2 shows the pseudo-code of one update step. Using this algorithm, the model can be sequentially updated by considering only one image at a time and is well suited to be embedded in the proposed continuous learning framework.

Once the models of visual concepts have been acquired, the system is able to recognise visual properties of a novel object using Algorithm 3 (e.g., answering the question "Is this object circular?"). If the probability that the value of a feature associated with a particular visual concept comes from the same probability distribution as the training values for that visual concept (i.e., the feature value is sufficiently close to the mean of the previously observed values), then the system answers 'yes'. Based on the weighted distance from the typical value of the feature, the system may also answer "probably yes", "probably not", or "no". By changing the thresholds $Tyes$, Tpy, and Tno, we can achieve more conservative or more liberal behaviour of the recognition algorithm. Combining this recognition algorithm with the incremental learning algorithm and by considering the update table presented in Table 1, we arrive at the incremental learning framework described in the previous section.

Algorithm 1 : Batch learning

Input: Set of training images $\mathcal{X}$, list of visual concepts $\mathcal{VC}_i$ for every training image X_i

Output: Models of visual concepts $mVC_i, i = 1 \dots N_{VC}$

1: Extract all visual features $F_j, j = 1 \dots N_F$ from every training image in $\mathcal{X}$.
2: **for** each $VC_i, i = 1 \dots N_{VC}$ **do**
3: Find a set of images $\mathcal{X}_i$ containing all images labeled with VC_i.
4: Calculate *means* and *variances* of the values of every feature F_j over all images in $\mathcal{X}_i$.
5: **end for**
6: Calculate *min* and *max* of all the values of every feature F_j.
7: Normalise all the variances with the obtained intervals of feature values, i.e., $nvar_{ij} = var_{ij}/(maxVar_j - minVar_j)^2$, $i = 1 \dots N_{VC}, j = 1 \dots N_F$.
8: **for** each $VC_i, i = 1 \dots N_{VC}$ **do**
9: Select the feature F_j with the smallest normalised variance $nvar_{ij}$.
10: Store *mean* and *variance* of the selected F_j to form mVC_i, a model of VC_i.
11: **end for**

Algorithm 2 : Update step

Input: Models of visual concepts $mVC_i, i = 1 \dots N_{VC}$, feature statistics FS, new image X and corresponding visual concept value VC

Output: Updated $mVC_i, i = 1 \dots N_{VC}$ and FS

1: If the model for VC has not been learned yet, initialise it.
2: Update feature *means* and *variances* related to VC (stored in FS).
3: Update total feature *mins* and *maxs*.
4: Proceed with the steps 7-11 of Algorithm 1.

Algorithm 3 : Recognition

Input: Image X, question "`Is this VC?`"

Output: Answer.

1: If the model for VC has not been learned yet, answer '`Don't know.`'
2: Determine which feature F_j the visual concept VC is associated with in the model mVC.
3: Extract the value of this feature F_j from the image X.
4: Calculate $d = \sqrt{(F_j - mVC.mean)^2/mVC.var}$.
5: If $d \in [0, Tyes]$, answer '`Yes.`'
6: If $d \in (Tyes, Tpy]$, answer '`Probably yes.`'
7: If $d \in (Tpy, Tno]$, answer '`Probably not.`'
8: If $d \in (Tno, \infty)$, answer '`No.`'

4 Implementation of the framework

The proposed framework for continuous learning of simple visual concepts is inherently multi-modal. The learning process involves acquisition, processing and analysis of visual data, as well as communication with the tutor. Therefore an artificial cognitive system, which would implement such a framework, has to consist of a number of components, including sensors, processing modules, communication subsystems, as well as the learning and recognition modules. 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.

In this section we present the implementation of our system. Fig. 1 depicts all of the components of our system and the connections between them schematically. The dashed arrows indicate how requests are passed from module to module and the solid arrows indicate the flow of results (data).

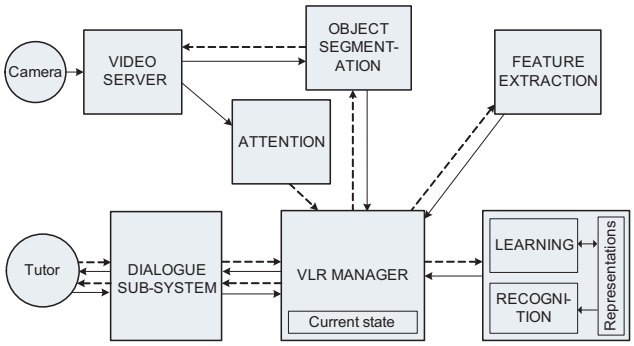


Figure 1: System diagram.

The **Visual learning and recognition manager** is the central module of the system. It continuously monitors and waits for recognition/learning requests from the dialogue sub-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 continues with the learning process. These decisions are made in accordance with the current state of the system and with the applied learning mode. This module is a practical implementation of the continuous learning framework presented in Section 2.

The visual input to the system is provided by the **Video server**. Images are retrieved from a video device (color camera) 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.

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. This module enables the system to take initiative and start a learning/recognition cycle without explicit user request.

The **Object segmentation** 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.

The detected part of the image (ROI) is then passed to the **Feature extraction** module along with the corresponding segmentation mask. This module extracts 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.

The **Learning and recognition** module is an implementation of the method described in Section 3. It 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 Visual learning and recognition manager, which processes them further, converts them to qualitative descriptions and acts according to the current learning mode.

All of the communication between the user and the system is done through the **Dialogue sub-system**. It 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.

To facilitate communication between the components we use the **process communication framework** BALT [6]. 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.

From the brief description of the modules above it can be seen that some of the modules are rather simple (e.g., Attention module, Object segmentation module) and that the implemented system works well only under certain assumptions (static background, simple objects, etc.). However, the overall system functions well and forms a firm basis for further development of individual modules and the cognitive system as a whole.

5 Experimental results

The cognitive vision system presented in the previous section was primarily designed to work in interaction with a user; it includes a communication sub-system 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

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

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

system about the objects in the scene, while the second part of the dialogue took place using the tutor-supervised mode of learning, when the system took the initiative and asked the tutor for clarification when needed. The communication between the tutor and the system is thus performed in a very natural and user friendly way. 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.

5.1 Learning of visual attributes

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. 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.

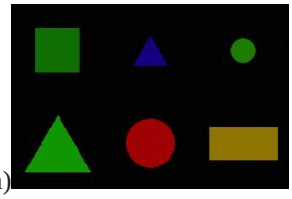
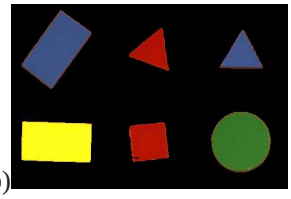
(a)  (b) 

Figure 3: (a) Synthetic images. (b) Segmented real images.

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 scoring rules are presented in Table 2; it shows how many points (-1 to 1) the system is rewarded with for each of the answers given in the first row, depending on the correct answer as given in the first column.

Table 2: Scoring table.

	YES	PY	PN	NO	DK
YES	1	0.5	-0.5	-1	0
NO	-1	-0.5	0.5	1	0

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. 4(a). All different learning strategies presented in Section 2 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.).

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. This is also evident from Fig. 4(b), which plots the amount of information, which is provided by the tutor. 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 exploratory 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 4(c).

5.2 Learning of spatial relations

The exact same system was also used for learning simple spatial relations. Only the features that were to be extracted from the image were changed. 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: '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. 5.

6 Conclusion

In this paper we presented a framework for continuous learning, which enables three modes of learning requiring different levels of tutor supervision. We proposed a method for incremental learning of visual properties by building associations between words describing an object's visual properties and visual features extracted from images. By embedding this method into the proposed learning framework, we were able to experimentally evaluate three learning strategies. The main conclusion is that the learning process should start with tutor-driven learning to enable development of consistent basic concepts. Once these concepts are acquired, the system can take the initiative and keep upgrading the knowledge in a tutor-supervised way, and when the knowledge is stable enough, even in an exploratory way.

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 have also included a robot arm in our learning system to enable reacher interaction with the environment; the system will thus 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 a general system for continuous learning that is capable of extending its ontology with other types of visual concepts that go beyond the elementary visual concepts addressed here.

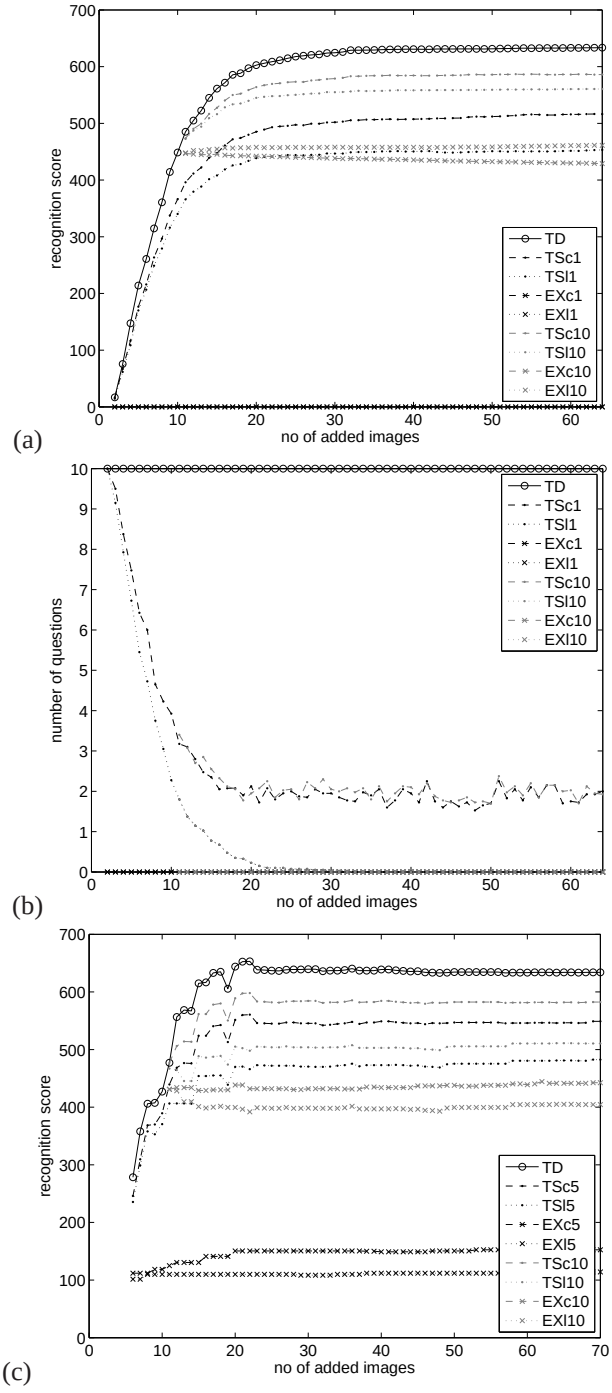


Figure 4: (a) Recognition score and (b) number of questions on synthetic images, (c) recognition score on real images.

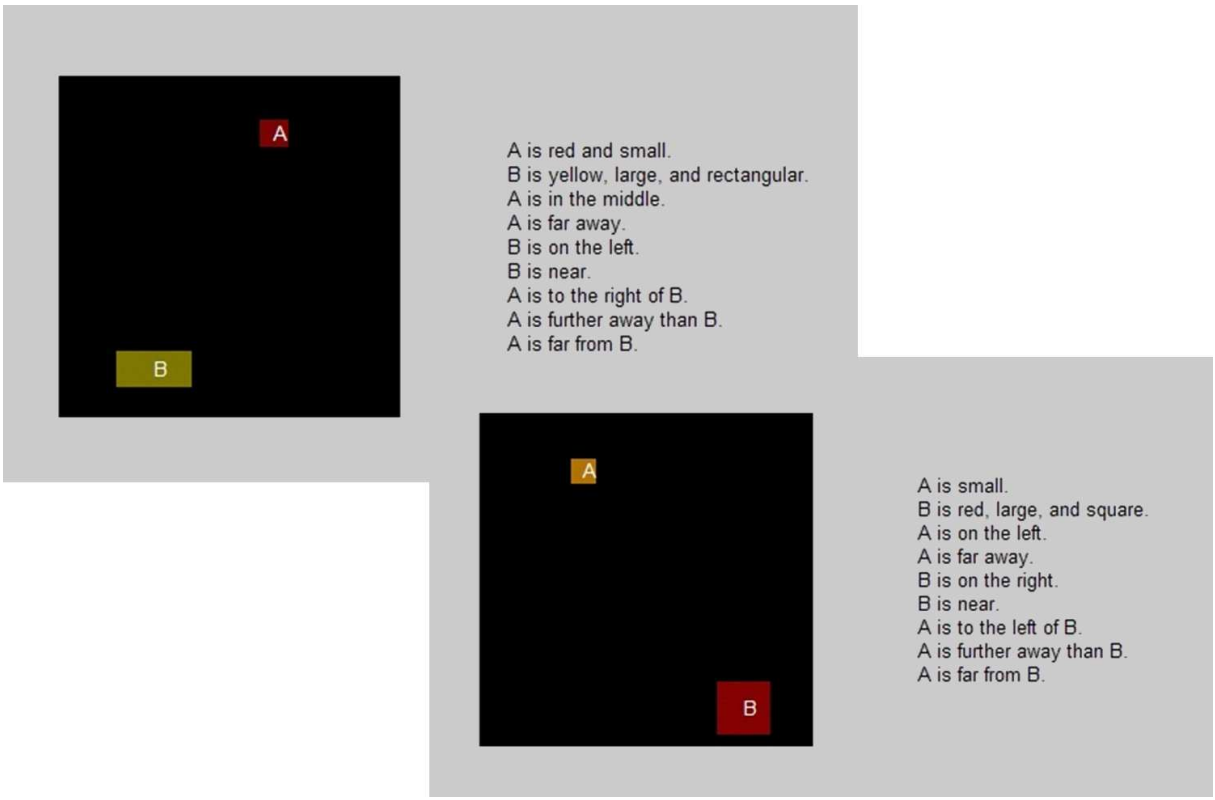


Figure 5: Automatically obtained scene descriptions.

Acknowledgement

This research has been supported in part by the following funds: Research program Computer Vision P2-0214 (RS), EU project CoSy (FP6-004250-IP), and EU project VISIONTRAIN (MRTN-CT-2004-005439).

References

[1] E. Ardizzone, A. Chella, M. Frixione, and S. Gaglio. Integrating subsymbolic and symbolic processing in artificial vision. *Journal of Intelligent Systems*, 1(4):273–308, 1992.

[2] A. Chella, M. Frixione, and S. Gaglio. A cognitive architecture for artificial vision. *Artificial Intelligence*, 89(1–2):73–111, 1997.

[3] P. Davidsson. Toward a general solution to the symbol grounding problem: Combining machine learning and computer vision. In *AAAI Fall Symposium Series, Machine Learning in Computer Vision: What, Why and How?*, pages 157–161. AAAI Press, 1993.

[4] P. Gärdenfors. Three levels of inductive inference. *Logic, Methodology, and Philosophy of Science IX*, pages 427–449, 1994.

[5] S. Harnad. The symbol grounding problem. *Physica D: Nonlinear Phenomena*, 42:335–346, 1990.

[6] N. Hawes. BALT & CAAT: Middleware for cognitive robotics. Technical Report CSR-07-1, University of Birmingham, School of Computer Science, 2007.

[7] M. J. Mayo. In *ACSC '03: Proceedings of the twenty-sixth Australasian conference on Computer science*, pages 55–60.

[8] M. T. Rosenstein and P. R. Cohen. Symbol grounding with delay coordinates. In *AAAI Technical Report WS-98-06, The Grounding of Word Meaning: Data and Models*, pages 20–21, 1998.

[9] D. K. Roy. Learning visually-grounded words and syntax for a scene description task. *Computer Speech and Language*, 16(3):353–385, 2002.

[10] D. K. Roy and A. P. Pentland. Learning words from sights and sounds: a computational model. *Cognitive Science*, 26(1):113–146, 2002.

[11] L. Steels and P. Vogt. Grounding adaptive language games in robotic agents. In *Proceedings of the Fourth European Conference on Artificial Life, ECAL'97*, Complex Adaptive Systems, pages 474–482, 1997.

[12] R. Sun. Symbol grounding: a new look at an old idea. *Philosophical Psychology*, 13(2):149–172, 2000.

[13] I. Ulusoy and C. M. Bishop. Generative versus discriminative methods for object recognition. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2*, pages 258–265, Washington, DC, USA, 2005. IEEE Computer Society.

[14] P. Vogt. The physical symbol grounding problem. *Cognitive Systems Research*, 3(3):429–457, 2002.