in: Memetic Computing, 1 (2009) 4, pp. 291-304

A Vision Architecture for Unconstrained and Incremental Learning of Multiple Categories

Stephan Kirstein[1,2], Alexander Denecke[1,3], Stephan Hasler[1],
Heiko Wersing[1], Horst-Michael Gross[2] and Edgar Körner[1]

[1]Honda Research Institute Europe GmbH
Carl-Legien-Str. 30 63073 Offenbach am Main, Germany
{stephan.kirstein, stephan.hasler, heiko.wersing, edgar.koerner}@honda-ri.de

[2]Ilmenau University of Technology
Neuroinformatics and Cognitive Robotics Lab
P.O.B. 10 05 65, 98684 Ilmenau, Germany
horst-michael.gross@tu-ilmenau.de

[3]Bielefeld University
CoR-Lab
P.O.B. 10 01 31, 33501 Bielefeld, Germany
adenecke@cor-lab.uni-bielefeld.de

Abstract

We present an integrated vision architecture capable of incrementally learning several visual categories based on natural hand-held objects. Additionally we focus on interactive learning, which requires real-time image processing methods and a fast learning algorithm. The overall system is composed of a figure-ground segregation part, several feature extraction methods and a life-long learning approach combining incremental learning with category-specific feature selection. In contrast to most visual categorization approaches, where typically each view is assigned to a single category, we allow labeling with an arbitrary number of shape and color categories. We also impose no restrictions on the viewing angle of presented objects, relaxing the common constraint on canonical views.

1 Introduction

An amazing capability of the human visual system is the ability to learn an enormous repertoire of visual categories. This large amount of categories is acquired incrementally during our life and requires at least partially the direct interaction with a tutor. Inspired by the child-like knowledge acquisition we propose an architecture for learning several visual categories in an incremental and interactive fashion. The architecture is composed of several building blocks including figure-ground segregation, feature extraction, a category learning module and user interaction. All these modules together allow the training of categories based on natural objects presented in hand.

The learning system proposed in this paper is partly based on earlier work dealing with online object identification in cluttered scenes (Wersing et al., 2007). For our learning system a novel incremental category learning method is proposed that combines a learning vector quantization (LVQ) (Kohonen 1989) network to approach the "stability-plasticity dilemma" with a category-specific forward feature selection. Based on this combination we are able to interactively learn a category-specific long-term memory (LTM) representation, where previous LTM models proposed by Kirstein, Wersing, & Körner (2008) could only be learned offline. Other major contributions are the integration of an enhanced figure-ground segregation method and the extraction of parts-based feature. In the following further related work with respect to categorization frameworks, online learning methods and life-long learning architectures is discussed in more detail.

1.1 Visual Categorization Architectures

In the past few years many architectures dealing with object detection and categorization tasks have been proposed in the computer vision community. Interestingly many of those approaches are based on local parts-based features, which are extracted around some defined interest points e.g. (Leibe et al., 2004; Willamowski et al., 2004; Agarwal et al., 2004) or on agglomerative clustering (Mikolajczyk, Leibe, & Schiele 2006) to build up object models for categories like faces or cars. The advantages of these approaches are their robustness against partial occlusion, scale changes, and the ability to deal with cluttered environments. One drawback is that such methods are typically restricted to the canonical view of a certain category. Thomas et al. (2006) try to overcome this limitation by training several pose-specific implicit shape models (ISM) (Leibe, Leonardis, & Schiele 2004) for each category. Afterwards detected parts from neighboring pose-dependent ISMs are linked by so-called "activation links". This allows the detection of categories from many viewpoints. Such categorization architectures, however, are designed for offline usage only, where the required training time is not important. This makes them unsuitable for our desired online and interactive training. A recent work of Fritz, Kruijff, & Schiele (2007) addresses this issue and proposes a semi-supervised and incremental clustering method for interactive category learning. This approach is, however, restricted to the canonical view of the categories.

1.2 Online and Interactive Learning Systems

The development of online and interactive learning systems became more and more popular in the recent years, see e.g. (Roth et al., 2006), (Steels & Kaplan, 2001), (Arsenio, 2004) or (Wersing et al., 2007). All these systems are able to identify several objects in cluttered environments, but are not applicable to categorization tasks. This is because their learning methods can not extract a more variable category representation. Nonetheless those models are useful as a short-term memory (STM) representation. Afterwards this representation is consolidated into a more abstract LTM representation of categories allowing a higher generalization performance compared to the object-specific STM representation. Of particular interest with respect to online and interactive learning of categories is the work of Skočaj et al. (2007). It enables learning of several simple color and shape categories by selecting a single feature which describes the particular category most consistently. The category itself is then represented by the mean and variance of this selected feature (Skočaj et al., 2007) or more recently by an incremental kernel density estimation using mixtures of Gaussians (Skočaj et al., 2008). Especially this feature selection enhances the categorization performance, but the restriction to a single feature allows only the representation of simple categories with little appearance changes. Therefore we propose a feature selection process that can incrementally select an arbitrary number of features, if they are required for the representation of a particular category.

1.3 Life-Long Learning Architectures

Based on the STM representation, which is assumed to be limited in capacity, we propose an incremental and life-long learning method to acquire a category-specific long-term memory (LTM) representation. For the LTM we approach the so-called “stability-plasticity dilemma”. This dilemma occurs when neural networks are trained with a limited and changing training ensemble, causing the well known “catastrophic forgetting effect” (French 1999). A common strategy for life-long learning architectures e.g. (Hamker, 2001; Furao & Hasegawa, 2006; Kirstein et al., 2008) is the usage of a node specific learning rate combined with an incremental node insertion rule. This permits plasticity of newly inserted neurons, while the stability of matured neurons is preserved. The major drawback of those architectures commonly used for identification tasks is the inefficient separation of cooccuring categories. This means for natural objects, which typically belong to several different categories (e.g. red-white car), a decoupled representation for each category (for category red, white and car) should be learned. This decoupling leads to a more condensed representation and higher generalization performance compared to object identification architectures. Another approach to the “stability-plasticity dilemma” was proposed by Ozawa et al. (2005). Here representative input-output pairs are stored into

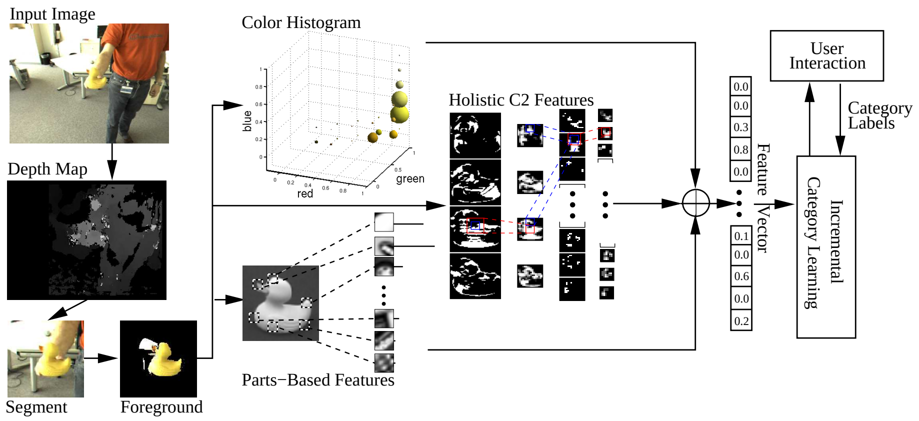


Figure 1: **Category Learning System.** Based on an object hypothesis extracted from the depth map a figure-ground segregation is performed. The detected foreground is used to extract color and shape features. Color features are represented as histogram bins in the RGB color space. In contrast to most other categorization approaches we combine general category independent features obtained from a detection hierarchy with parts-based features. All extracted features are concatenated into a single structureless vector. This vector together with the category labels provided by an human tutor, is the input to the incremental category learning module.

a long-term memory for stabilizing an incremental learning radial basis function (RBF) like network. Additionally it also accounts for a feature selection mechanism based on incremental principal component analysis, but no class-specific feature selection is applied. Therefore this method it unsuitable for categorization tasks without modification.

In the following we describe step by step the building blocks of our learning system illustrated in Fig. 1. The first processing block extracts the object hypothesis from cluttered scenes. This hypothesis is further refined by a figure-ground segregation method as described in Section 2. Additionally we describe all used feature extraction methods in Section 3. The extracted shape and color information is combined and used to train the proposed life-long learning vector quantization approach described in Section 4, which is trained in direct interaction with a human tutor. The target of our system is interactive and life-long learning of categories. Therefore in Section 5 the learning results of our proposed methods are shown for differently complex databases. Additionally we show the interactive learning capability of the proposed learning system under real-world constraints. Finally we discuss the results and related work in Section 6.

2 Preprocessing and Figure-ground Segregation

One of the essential problems when dealing with learning in unconstrained environments is the definition of a shared attention concept between the learning system and the human tutor. Specifically this is necessary to decide what and when to learn. In our architecture we use the peri-personal space concept (Goerick et al., 2006), which basically is defined as the manipulation range around an active vision system. Everything in this short distance range is of particular interest to the system with respect to interaction and learning. Therefore we use a stereo camera system with a pan-tilt unit and parallelly aligned cameras, which deliver a stream of image pairs. Depth information is calculated after the correction of lens distortions. This depth information is used to generate an interaction hypothesis in cluttered scenes, which after its initial detection is actively tracked until it disappears from the peri-personal attention range. Additionally we apply a color constancy method (Pomierski & Gross 1996) and a size normalization of the hypothesis. Both operations ensure invariances, which are beneficial for any kind of recognition system, but are essential for fast online and interactive learning in unconstrained environments. Finally a region of interest (ROI) of an object view is extracted and scaled to a fixed segment size of 144x144 pixel.

The extracted segment $\mathbf{j}^i$ contains the object view, but also a substantial amount of background clutter as can be seen in Fig. 2. For the incremental build-up of category representations it is beneficial to suppress such clutter, otherwise it would slow down the learning process and considerably more training examples are necessary. Therefore we apply an additional figure-ground segregation as proposed by Denecke et al. (2009) to reduce this influence. The basic idea of this segregation method illustrated in Fig. 2 is to train for each segment $\mathbf{j}^i$ a learning vector quantization (LVQ) network based on a predefined number of distinct prototypes for foreground and background. As an initial hypothesis for the foreground the noisy depth information belonging to the extracted segment is used. The noise of this hypothesis is caused by the ill-posed problem of disparity calculation and is basically located at the corner of the corresponding object view. Furthermore also "holes" at textureless object parts are common. Due to the fact that the objects are presented by hand, skin color parts in the segment are systematic noise, which we remove from the initial foreground hypothesis based on the detection method proposed by Fritsch et al. (2002). Due to this skin color removal faces and gestures can not be learned with this preprocessing. Nevertheless with a modified preprocessing as proposed in Wersing et al. (2007) a combined learning of objects and faces can be achieved. The learning of each LVQ prototype is based on feature maps consisting of RGB-color features as well as the pixel positions. Instead of the standard Euclidean metrics for the distance computation an extended version of the generalized matrix LVQ (Schneider, Biehl, & Hammer 2007) approach is used. This metric adaptation is used to learn relevance factors for each prototype and feature dimension. These local relevance factors are adapted online and weight dynamically the different feature maps to discriminate between foreground and background. For the

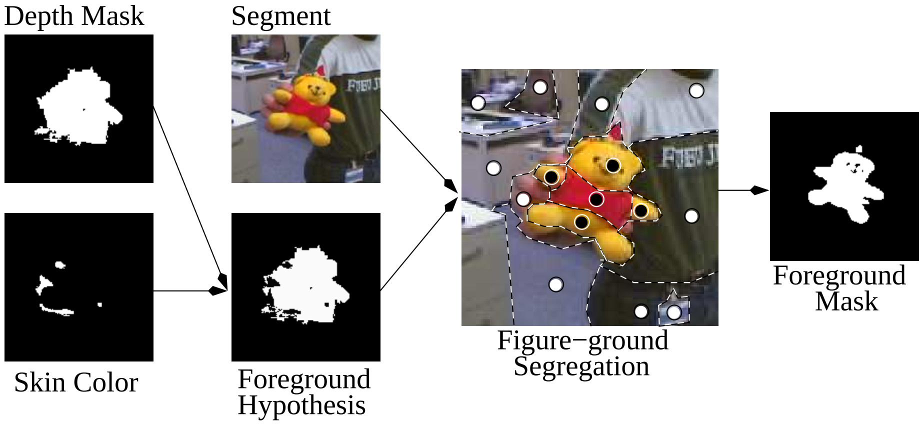


Figure 2: **Figure-ground Segregation.** Based on the extracted segment, the corresponding depth mask and a skin color removal a foreground hypothesis is generated. This hypothesis includes a considerable amount of noise and clutter, which the applied figure-ground segregation method strongly reduces. The noise in the foreground hypothesis is a consequence of the ill-posed problem of disparity calculation, which introduces noise mainly around the object or at textureless parts of the object. After generating this hypothesis a generalized matrix LVQ network is trained, based on a predefined number of prototypes and prototype specific relevance factors. Based on the learned network the refined foreground mask is calculated. Only the foreground pixels are used for feature extraction in the following steps.

purpose of figure-ground segregation such local matrices lead to a significantly better foreground classification (Denecke et al., 2009), which directly enhances the category learning. Additionally these local relevance factors generate more complex decision boundaries based on a small set of LVQ prototypes allowing for figure-ground segregation in real-time. The output of this segregation step is a binary mask $\boldsymbol{\xi}^i$ defining the foreground. In the following processing steps only foreground pixels are used to extract features.

3 Feature Extraction

For our category learning system we use several feature extraction methods providing shape and color information, but we do not give this qualitative separation of the extracted features to the learning system as a priori information. For our categorization task we are particularly interested in discovering the structure from the high-dimensional but sparse feature vectors by using a flexible metrical adaptation. Assume you want to learn the category “fire engine”, where all training examples are mainly of red color. If the learning of this category is restricted to shape features only, it would be difficult to distinguish it from other cars and trucks. This is because the most distinctive feature, the red color, is not included in the feature representation. Therefore we let the learning algorithm decide which feature combinations are most suitable to represent a category. As a consequence we concatenate all extracted features of an object view into a single high-dimensional and structureless feature vector $\mathbf{x}^i = (x_1^i, \ldots, x_F^i)$, where F denotes the total number of features. Although the overall dimensionality $F > 10000$ is high, typically only a subset of 15-30% of the features are activated with $x_f^i > 0$. Additionally each vector $\mathbf{x}^i$ is assigned to a list of category labels $\mathbf{t}^i = \{t_1^i, \ldots, t_C^i\}$. We use C to denote the current number of represented color and shape categories, where each $t_c^i \in \{-1, 0, +1\}$ labels an $\mathbf{x}^i$ as positive or negative example of category c. The third state $t_c = 0$ is interpreted as unknown category membership, which means that all vectors $\mathbf{x}^i$ with $t_c^i = 0$ have no influence on the representation of category c.

3.1 Histogram Binning for Color Feature Extraction

For the representation of color information we use the common histogram binning method which combines robustness against view and scale changes with computational efficiency (Swain & Ballard 1991). Overall F_{co} =6x6x6=216 histogram bins within the RGB color space are used, where typically a small amount of features are specific for a complete color category.

3.2 Hierarchical Feed-Forward Shape Feature Extraction

We use a feed-forward feature extraction architecture inspired by the Neocognitron (Fukushima 1980) to extract shape features. This architecture is based on weight-sharing and a succession of feature detection and pooling stages (see (Wersing & Körner 2003) for details). The feature detectors of this architecture are obtained through unsupervised learning, providing a set of general but less category-specific features. Starting point for the feature extracting process is the segment $\mathbf{j}^i$ and the foreground mask $\boldsymbol{\xi}^i$. The first feature-matching layer S1 is composed of four orientation sensitive Gabor filters $\mathbf{z}_{s1}^m(x, y)$ with $m = 1, \ldots, 4$ that perform a local orientation estimation. To compute the response $\hat{P}_{s1}^{mi}(x, y)$ of a simple cell of this layer, responsive to feature type m at position (x, y) first the segment $\mathbf{j}^i$ is convolved with a Gabor filter $\mathbf{z}_{s1}^m(x, y)$:

$$\hat{P}_{s1}^{mi}(x, y) = \begin{cases} |\mathbf{j}^i * \mathbf{z}_{s1}^m(x, y)| & : \quad \xi^i(x, y) > 0 \\ 0 & : \quad else \end{cases} . \tag{1}$$

This computation of local edge responses is restricted to the positions in the foreground mask with $\xi^i(x, y) > 0$, whereas the $*$ denotes the inner product of two vectors. Additionally a winners-take-most (WTM) mechanism between features at the same position is performed and a simple threshold function with a common threshold for all cells in layer S1 is applied. We denote the final output of the S1 layer at position (x, y) as $P_{s1}^{mi}(x, y)$. The following C1 layer subsamples the S1 output $\mathbf{P}_{s1}^{mi}$ by pooling down to a quarter in each direction (e.g. 144x144 S1 features are pooled down to 36x36 C1 features):

$$P_{c1}^{mi}(x,y) = \tanh\left(\mathbf{P}_{s1}^{mi} * \mathbf{z}_{c1}(x,y)\right), \tag{2}$$

where $\mathbf{z}_{c1}(x,y)$ is a normalized Gaussian pooling kernel with width σ_{c1}, identical for all features m, and tanh is the hyperbolic tangent function.

The S2 layer is sensitive to local combinations of the orientation estimation features extracted from layer C1. The so-called combination features of this S2 layer (for this experiment 50 different shape features with $n = 1,\ldots,50$ are used) are trained with sparse coding (see (Wersing & Körner 2003) for details). The response $\hat{P}_{s2}^{ni}(x,y)$ of one S2 cell is calculated in the following way:

$$\hat{P}_{s2}^{ni}(x,y) = \sum_{m} \mathbf{P}_{c1}^{mi} * \mathbf{z}_{s2}^{nm}(x,y), \tag{3}$$

where $\mathbf{z}_{s2}^{nm}(x,y)$ is the receptive field vector of the S2 cell of feature n at position (x,y), describing connections to the plane m of the previous C1 cells. Similar to the S1 layer a WTM mechanism and a final threshold function are performed in this S2 layer. The final C2 layer again performs a spatial integration and reduces the resolution by half in each direction (i.e. 36x36 S2 features are down-sampled to 18x18 C2 features). For this operation the same pooling mechanism as in layer C1 is used, so that the final dimensionality is F_{c2} =50x18x18=16200.

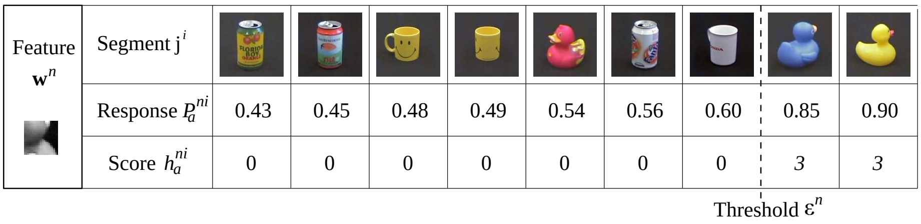
3.3 Parts-based Shape Feature Extraction

In contrast to the hierarchical feed-forward feature extraction architecture the parts-based features are trained in a supervised manner with respect to category specificity. We combine these different shape features to show the ability of the category learning method to select appropriate features out of a large amount of possible candidates. Such feature combinations are rare because most categorization methods rely on parts-features only.

3.3.1 Extraction of Parts-Based Features During Online Learning

The parts-based feature detection (see (Hasler, Wersing, & Körner 2007) for details) is based on a preselected set of SIFT-descriptors (Lowe 2004). Commonly in categorization frameworks such descriptors are only extracted at a small number of interest points. In contrast to this in our approach they are extracted at any location in the segment $\mathbf{j}^i$, with foreground mask $\xi^i(x,y) > 0$. For each segment $\mathbf{j}^i$ the similarity $P_a^{mi}(x,y)$ $(m = 1,\ldots,500)$ between the stored feature detector $\mathbf{z}_a^m$ and the SIFT-response $\hat{\mathbf{P}}_a^{mi}(x,y)$ corresponding to segment $\mathbf{j}^i$ at position (x,y) is calculated using the dot product:

$$P_a^{mi}(x,y) = \begin{cases} \hat{\mathbf{P}}_a^{mi}(x,y) * \mathbf{z}_a^m & : \quad \xi(x,y) > 0 \\ 0 & : \quad else \end{cases} \tag{4}$$



Feature $\mathbf{w}^n$	Segment j^i									
	Response P_a^{ni}	0.43	0.45	0.48	0.49	0.54	0.56	0.60	0.85	0.90
	Score h_a^{ni}	0	0	0	0	0	0	0	*3*	*3*

Threshold ε^n

Figure 3: **Illustration of Feature Candidate Scoring.** For each feature candidate $\mathbf{w}_a^n$ the corresponding response P_a^{ni} is calculated for each training image i. The threshold ϵ^n is chosen so that all $P_a^{ni} \geq \epsilon^n$ belong to the same category and are assigned to a constant scoring value $h_a^{ni} = 3$. The scoring values are used to guide the iterative selection process, by adding the feature candidate $\mathbf{w}_a^n$ to the list of selected features $\mathbf{z}_a^m$ leading to the highest additional gain.

The final response P_a^{mi} for the feature detector $\mathbf{z}_a^m$ and the current segment $\mathbf{j}^i$ is defined as:

$$P_a^{mi} = \max_{x,y}(P_a^{mi}(x,y)). \tag{5}$$

So that for each feature detector $\mathbf{z}_a^m$ only the maximum response is used ($F_a = m = 500$), neglecting all spatial and configurational information. Such information is commonly included in categorization methods like in (Leibe, Leonardis, & Schiele 2004), but requires a high amount of representational resources. Neglecting this information leads to a more compact representation with an efficient reuse and combination of parts, which enhances the learning speed for interactive category learning tasks. Another important issue is that this parts-based feature representation is invariant with respect to rotations in the image plane. As a final step the non-sparse feature activations are transformed into a sparse representation, by choosing only 10% of the features with highest detector responses for segment $\mathbf{j}^i$.

3.3.2 Scheme for Selecting Optimal Parts-Based Feature Detectors

In the following we describe how the feature detectors $\mathbf{z}_a^m$ are determined. In general this offline feature selection scheme tries to find an optimal set of detectors with respect to robust redetection of features and category specificity (Hasler, Wersing, & Körner 2007). As a first step of this scheme SIFT-descriptors are calculated for each location in the training image i with $\xi(x,y)^i > 0$. Based on these SIFT-descriptors a k-means clustering with 100 components is applied for each image i. This clustering step is done to improve the generalization performance and to reduce the number of descriptors. Based on all obtained k-means clusters, used as candidate descriptors $\mathbf{w}_a^n$ with $n = 1, .., N$, the feature responses P_a^{ni} are calculated. Afterwards the minimal thresholds ϵ^n are computed in a way that all segments i with $P_a^{ni} \geq \epsilon^n$ belong to the same category. Each image i satisfying this constraint is assigned to a constant scoring value $h_a^{ni} = 3$, which is illustrated in Fig. 3 for a single $\mathbf{w}_a^n$. The iterative feature selection determines a predefined number of features by selecting at each iteration the best feature candidate $\mathbf{w}_a^n$ that leads to the highest additional gain. This selection is therefore based on the scoring values h_a^{ni}, already selected features $\mathbf{z}_a^m$ with $m = 1, \ldots, M$ and all remaining candidates $\mathbf{w}_a^n$:

$$n = \arg\max_{n \in N} \left(\sum_i \Phi \left(h_a^{ni} + \sum_{m \in M} h_a^{mi} \right) \right), \tag{6}$$

where $\Phi(z)$ is defined as Fermi function. Finally the determined candidate feature $\mathbf{w}_a^n$ is added to the set of selected features $\mathbf{z}_a^{M+1} = \mathbf{w}_a^n$. Afterwards the collection of further candidate features $\mathbf{w}_a^n$ is repeated until a predefined number of selected features is reached. Overall this scheme selects parts-based detectors, which describe the known categories best. Additionally the selected features are general enough to represent arbitrary unknown shape categories, that are not included in the set of training images.

4 Incremental and Interactive Category Learning

Similar to the human brain different memory concepts (see Fig. 4) are used to interactively learn several visual categories. Therefore labeled training vectors are first stored into an intermediate short-term memory (STM), which is assumed to be limited in capacity. This object-specific STM performs fast one-shot learning and typically strongly reduces the number of necessary representatives $\mathbf{r}$, by performing a kind of novelty detection. Based on this limited and changing STM a knowledge transfer method is proposed that is able to iteratively consolidate the object-specific STM information into a more abstract category-specific long-term memory (LTM) illustrated in Fig. 4. For this transfer we focus on life-long learning and interactive training of arbitrary categories, which require a compact and efficient LTM representation.

4.1 Online Vector Quantization to Build a Short-Term Memory

The online vector quantization (oVQ) model developed by Kirstein, Wersing, & Körner (2008) provides fast appearance-based learning of complex-shaped objects. The proposed model stores exemplar-based representatives $\mathbf{r}^l$ with $l = 1, \ldots, L$ in a so-called short-term memory representation, providing a limited and changing object-specific memory. Each representative $\mathbf{r}^l$ is labeled with a class o, which corresponds to a specific pattern of category labels $\mathbf{t}_o = \{t_1, \ldots, t_C\}$. The acquisition of representatives is based on a similarity threshold ϵ_{stm}. We denote the similarity of feature vector $\mathbf{x}^i$ and representative $\mathbf{r}^l$ by

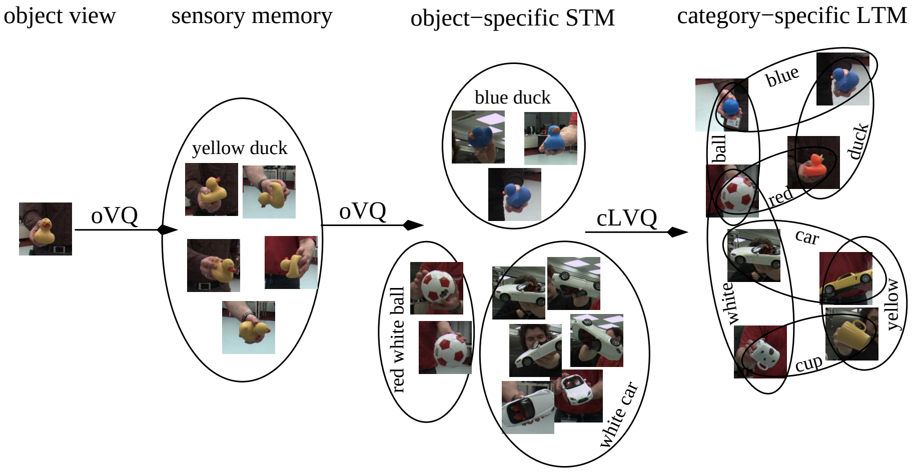


Figure 4: **Category Learning with a Coupled Short and Long-Term Memory Concept.** Object views are first buffered into the sensory memory using the online vector quantization (oVQ) method until label information is provided by the tutor. Due to our assumption that only views of a single object are collected into this memory, all collected views have the same label information, even if they are collected before the labeling. After labeling this knowledge is transferred into the STM using the same oVQ learning method. The object-specific STM is limited in capacity allowing only to store the latest presented objects. Therefore we use the life-long learning cLVQ method to deal with the "stability-plasticity dilemma" and iteratively transfer the STM information into the category-specific LTM.

A^{il}:

$$A^{il} = \exp\left(-\frac{||\mathbf{x}^i - \mathbf{r}^l||^2}{\sigma}\right), \tag{7}$$

where σ is chosen for convenience such that the average similarity is approximately equal to 0.5.

We define R_o as set of all representatives $\mathbf{r}^l$ that belong to the same class o. For one learning step the similarity A^{il} between the current feature vector $\mathbf{x}^i$ and all representatives $\mathbf{r}^l \in R_o$ is calculated and the maximum value is selected as:

$$A^i_{\max} = \max_{l \in R_o} A^{il}. \tag{8}$$

The feature vector $\mathbf{x}^i$ is added to the representation of class o, if $A^i_{max} < \epsilon_{stm}$. Otherwise we assume that the vector $\mathbf{x}^i$ is already sufficiently well represented. Based on the selected STM representatives $\mathbf{r}^l$ object identification with good detection performance is possible using a simple nearest neighbor classifier (Kirstein, Wersing, & Körner 2008).

Compared to the naive approach, where each $\mathbf{x}^i$ is stored as representative

$\mathbf{r}^l$ Kirstein, Wersing, & Körner (2008) could show that the number of representatives $\mathbf{r}^l$ can be considerably reduced by about 30% without losing generalization performance. Additionally we assume a limited memory size of the STM, requiring a deletion strategy of feature vectors if the capacity limit is reached. Therefore STM vectors are removed that belong to the same class o of a particular category label list for which almost no categorization errors occur. Such vectors are already successfully transferred to the LTM and can be deleted without information loss.

4.1.1 Short-Term Memory with Additional Sensory Memory

For interactive learning scenarios usually only few object views are presented to the system. Additionally learning systems are typically separated into a train and test phase, where commonly distinctive views are used. To relax this separation and to make the most efficient use of object views, we introduce a sensory memory concept for temporarily remembering views of the currently attended object, by using the same one-shot learning method as used for the STM. The basic assumption behind this memory concept is that only views of a single object are inserted and that the memory is cleared if the object identity changes. For this we use the disappearance of the object from peripersonal space to detect an identity change. This allows that object views can be first used to test the STM and LTM representation and after providing confirmed labels the same views can also be used to enhance the representation by transferring them into the STM, even if they where recorded before the confirmation.

4.2 Category Learning Vector Quantization to Build a Long-Term Memory

The learning method for the memory consolidation from the STM into the LTM is the most complex part of our architectures. Therefore a more detailed description is given in the following. For this learning method we propose a combination of an incremental exemplar-based network and a forward feature selection method (see (Guyon & Elissee 2003) for an introduction to feature selection methods). We call this combination category LVQ (cLVQ) in the following. The proposed cLVQ allows life-long learning and also enables a separation of cooccuring visual categories, which most exemplar-based networks can not handle. Both parts are optimized together to ensure a compact and efficient category representation, which is necessary for fast and interactive learning.

The forward feature selection method is used to find low dimensional subsets of category-specific features by predominately selecting features, which occur almost exclusively for a certain category. For guiding this selection process a feature scoring value h_{cf} for each category c and feature f is calculated. This scoring is only based on previously seen examples of a certain category and can strongly change if further information is encountered. Therefore a continuous update of the h_{cf} values is required to follow this change. The exemplar-based network part of the cLVQ method is used to approach the "stability-plasticity dilemma" of life-long learning problems. In general the exemplar-based network of our learning system represents the variations of different category members (e.g. normal car and cabriolet) and object poses (e.g. front and side view of cars). Although there are several neural network models for incremental learning available like the growing neural gas (Fritzke 1995) or growing cell structures (Fritzke 1994) those methods typically can not be used for life-long learning tasks without modification.

4.2.1 Distance Computation and Learning Rule

The LTM representative vectors $\mathbf{w}^k$ with $k = 1, \ldots, K$ are built up incrementally, where K denotes the current number of allocated vectors $\mathbf{w}$. Each $\mathbf{w}^k$ is assigned to a label vector $\mathbf{u}^k$ where $u_c^k \in \{-1, 0, +1\}$ is the model target output for category c, representing positive, negative, and missing label output, respectively. Each cLVQ node $\mathbf{w}^k$ can therefore represent several categories c. For the category-specific distance computation d_c we use a weighting Euclidean metrics with specific weight factors λ_{cf} related to the work of Hammer & Villmann (2002):

$$d_c(\mathbf{r}^l, \mathbf{w}^k) = \sum_{f=1}^{F} \lambda_{cf} (r_f^l - w_f^k)^2, \tag{9}$$

where the category-specific weights λ_{cf} are updated continuously. We denote the set of selected features for a category $c \in C$ as S_c. We choose $\lambda_{cf} = 0$ for all $f \notin S_c$, and otherwise adjust it according to a scoring procedure explained later. The winning nodes $\mathbf{w}^{k_{\min}(c)}(\mathbf{r}^l)$ are calculated independently for each category c, where $k_{\min}(c)$ is determined in the following way:

$$k_{\min}(c) = \arg\min_k \; d_c(\mathbf{r}^l, \mathbf{w}^k) \quad \forall k \text{ with } u_c^k \neq 0. \tag{10}$$

Each $\mathbf{w}^{k_{\min}(c)}(\mathbf{r}^l)$ is updated based on the standard LVQ learning rule (Kohonen 1989), but is restricted to feature dimensions $f \in S_c$:

$$w_f^{k_{\min}(c)} := w_f^{k_{\min}(c)} + \mu\, \Theta^{k_{\min}(c)} (r_f^l - w_f^{k_{\min}(c)}) \quad \forall f \in S_c, \tag{11}$$

where $\mu = 1$ if the categorization decision for $\mathbf{r}^l$ was correct, otherwise $\mu = -1$ and the winning node $\mathbf{w}^{k_{\min}(c)}$ will be shifted away from $\mathbf{r}^l$. Additionally $\Theta^{k_{\min}(c)}$ is the node-dependent learning rate as proposed in (Kirstein, Wersing, & Körner 2008).

4.2.2 Feature Scoring and Category Initialization

The scoring value h_{cf} is updated for every new STM representative $\mathbf{r}^l$ that was inserted for the last presented object:

$$h_{cf} = H_{cf} / (H_{cf} + \bar{H}_{cf}). \tag{12}$$

The H_{cf} and $\bar{H}_{cf}$ are the number of previously seen positive and negative training examples of category c, where the corresponding feature f was active ($r_f > 0$). For computational efficiency of the learning dynamics, explained in the next section, it is beneficial to make the feature scoring update step only if the training object disappeared from the peri-personal space and not for every newly inserted $\mathbf{r}^l$. For each newly inserted $\mathbf{r}^l$ of the last presented object, the counter values H_{cf} and $\bar{H}_{cf}$ are updated with $H_{cf} := H_{cf} + 1$ if $\mathbf{r}^l$ is labeled with $t_c^l = +1$ and $r_f^l > 0$ and $\bar{H}_{cf} := \bar{H}_{cf} + 1$ if $t_c^l = -1$ and $r_f^l > 0$. The score h_{cf} defines the metrical weighting in the cLVQ representation space. We thus choose $\lambda_{cf} = h_{cf}$ for all $f \in S_c$ and $\lambda_{cf} = 0$ otherwise.

For each category c with STM vectors $\mathbf{r}^l$ and corresponding category label $t_c^l = +1$ that occurred the first time for the last presented object, we initialize this category c with a single feature and one cLVQ node. We select the feature $v_c = \arg\max_f(h_{cf})$ with the largest score value and initialize $S_c = \{v_c\}$. As the initial cLVQ node the training vector $\mathbf{r}^l$ is selected, where the selected feature v_c has the highest activation, i.e. $\mathbf{w}^{K+1} = \mathbf{r}^q$ with $r_{v_c}^q \geq r_{v_c}^l$ for all l. The attached label vector is chosen as $u_c^{K+1} = +1$ and zero for all other categories.

4.2.3 Learning Dynamics

The learning dynamics of the cLVQ memory architecture is based on an optimization loop (see Fig. 5), which applies iteratively small changes to the representation of erroneous categories in the following steps:

Step 1: Feature Testing. The target of this step is the addition or removal of features for the category-specific metrics, based on the available STM representatives $\mathbf{r}^l$ and the corresponding training errors. For each category c we determine the set of positive errors $E_c^+ = \{l | t_c^l = +1 \wedge t_c^l \neq u_c^{k_{min}(c)}(\mathbf{r}^l)\}$ and negative errors $E_c^- = \{l | t_c^l = -1 \wedge t_c^l \neq u_c^{k_{min}(c)}(\mathbf{r}^l)\}$. Afterwards we compare the total number of positive errors $\#E_c^+$ with the corresponding number of negative ones $\#E_c^-$. If $\#E_c^+ \geq \#E_c^-$ then we compute $e_{cf}^+ = \sum_{l \in E_c^+} \Phi_{ltm}(r_f^l) / \sum_{l \in E_c^+} 1$, where Φ_{ltm} is a Heaviside function.

The score e_{cf}^+ is the ratio of active feature entries for feature f in the positive training errors of class c. We want to add now a feature to the category feature set S_c, which both contributes to c by having a high scoring value h_{cf} and also is very active for the encountered error set E_c^+. Therefore we choose $v_c = \arg\max_{f \notin S_c}(e_{cf}^+ + h_{cf})$ and add $S_c := S_c \cup \{v_c\}$. The added feature dimension modifies the cLVQ metrics by changing the decision boundaries of all Voronoi clusters assigned to category c, which potentially reduces the remaining categorization errors. Therefore the change of the categorization error is calculated based on the newly added feature v_c. If the performance increase for category c is larger than a threshold ϵ_{ltm}^1, then v_c is permanently added. Otherwise it is removed and excluded for further training iterations until the h_{cf} values of category c are updated. An analog step is performed, if the number of negative errors is larger than the number of positive errors ($\#E_c^+ < \#E_c^-$). In

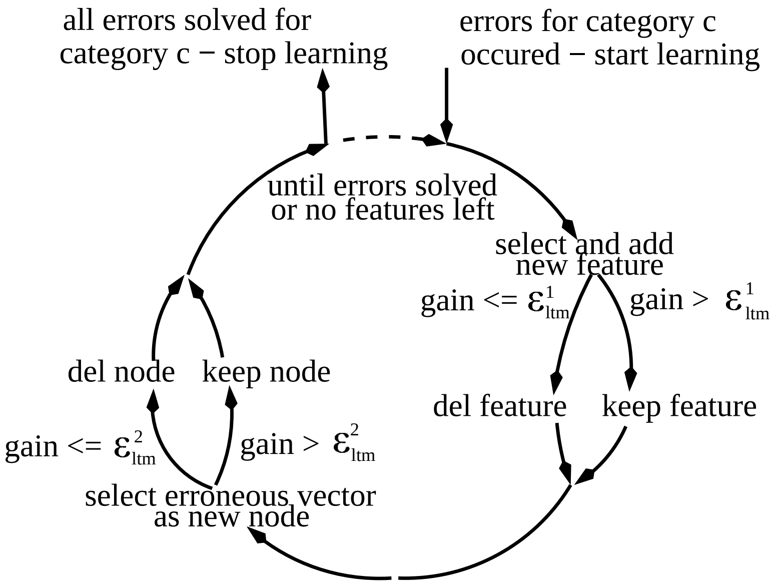


Figure 5: **Illustration of the cLVQ Optimization Loop.** The basic idea of this loop is to make small modifications to the representation of erroneous categories. If the gain in categorization performance based on all available training examples of category c is above the insertion threshold the modification is kept and otherwise retracted.

such cases a feature is removed and then again the performance gain is computed for the final decision on the removal.

Step 2: cLVQ Node Testing. We test new cLVQ nodes similar to Step 1 only for erroneous categories. In previous work nodes are inserted for training vectors with smallest distance to wrong winning nodes (Kirstein, Wersing, & Körner 2008). In this paper we propose to insert new cLVQ nodes based on training vectors $\mathbf{r}^l$ with the most categorization errors $t^l_c \neq u_c^{k_{min}(c)}(\mathbf{r}^l)$ for all categories C, until for each erroneous category c at least one new node is inserted. This leads to a more compact representation, because a single node typically improves the representation of several categories.

Again we calculate the performance increase based on all currently available training vectors. If this increase for category c is above the threshold ϵ^2_{ltm}, we make no modifications to the cLVQ node labels of the corresponding newly inserted nodes. Otherwise we set the corresponding labels u^k_c of the newly inserted nodes $\mathbf{w}^k$ to zero, so that node k does not contribute to the representation of category c. Finally we remove nodes where all u^k_c are zero.

Step 3: Stop condition. Iterate Step 1 and Step 2 until all remaining categorization errors are resolved or all possible features f of erroneous categories c are tested.

4.3 User Interaction

For interactively providing label information to the STM and LTM we use a simple state-based user interface. This user interface is based on a list of predefined audio labels. This list additionally includes some wild card labels (e.g. "property one"), to allow the labeling of categories for which no category label is defined. All labels can be provided to the system in an arbitrary order and combination. In general the user interaction is composed of two operation modes. For the default user interaction mode the learning system first integrates the category decisions over 5 seconds ($\approx$ 20-30 segments), where $u_c^{k_{\min}(c)} = +1$ attached to the winning node $\mathbf{w}^{k_{\min}(c)}$ means that the category c was detected and $u_c^{k_{\min}(c)} = -1$ that the category c was not detected. To generate the hypothesis list for the currently presented object we calculate for all categories $c \in C$ the ratio between segments assigned to $u_c^{k_{\min}(c)} = +1$ and the total number of already integrated category decisions. If this ratio for category c is above the empirically determined threshold 0.65 the category c is added to the hypothesis list. Finally all categories added to this hypothesis list are communicated to the user. In cases where the detection certainty of all categories c is below 0.65 the system respond with "unknown category". Additionally the hypothesis list is repeatedly communicated to the user (in 5 second intervals), while newly acquired segments are also used to refine this list. As a reaction to this communicated hypothesis the human tutor can confirm or correct this list. After the human response new training views are collected to enhance the category representation in the LTM. Furthermore it is also possible for the user to directly provide category labels, in order to label previously unknown categories.

5 Experimental Results

The proposed category learning approach is the most critical part and defines the overall system performance. Therefore in the following section several offline and interactive learning experiments are performed. For the interactive learning experiment complex-shaped objects are freely rotated by hand in front of our active camera system. Based on the extracted segment and the corresponding foreground mask, color, parts-based, and hierarchical shape features (C2 features) are extracted and concatenated into a single high-dimensional but sparse feature vector $\mathbf{x}^i$. These $\mathbf{x}^i$ together with the corresponding category labels $\mathbf{t}^i$ are used to incrementally learn the category-specific LTM representation under real-world conditions. For the offline experiments two databases with the same objects but different experimental setups are used. Additionally we investigated the effect of using different shape feature sets. The first set is composed of color and parts-based features with $F = F_{co} + F_a = 716$, while for

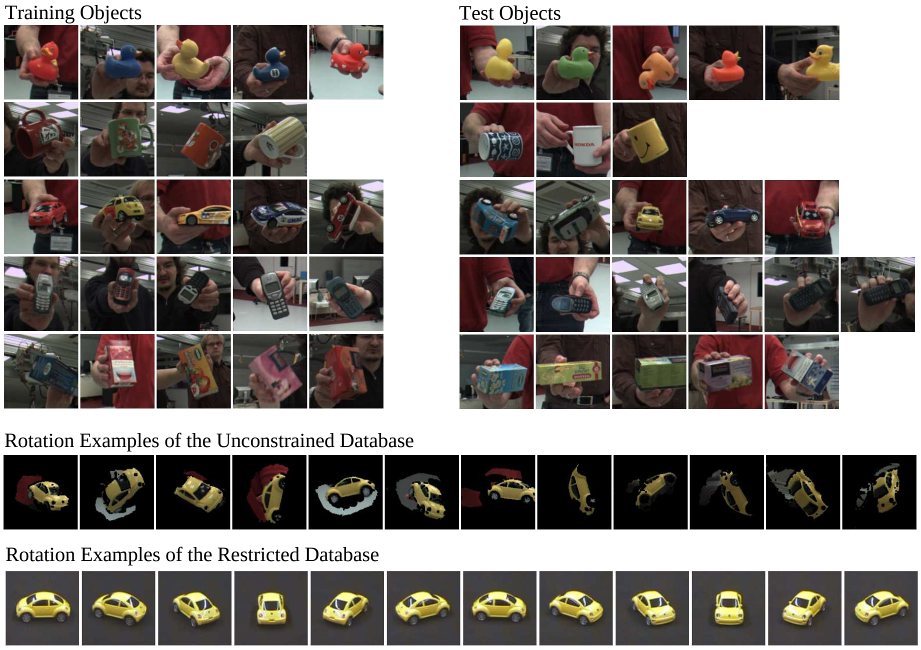

Training Objects
Test Objects
Rotation Examples of the Unconstrained Database
Rotation Examples of the Restricted Database

Figure 6: **Categorization Database.** Training and test objects used for the offline categorization experiments based on two different databases collected under different experimental settings. The objects are aligned so that each row corresponds to one of the five shape categories. For the *restricted database* all objects where shown in front of a black background and are rotated around the vertical axis. For the *unconstrained database* the same training and test objects are used, but the objects are shown in a cluttered office environment and are freely rotated by hand covering almost the complete viewing sphere. Additionally some rotation examples are shown for each database, where for the examples of the *unconstrained database* also the corresponding foreground mask is applied to show the segment part used for feature extraction.

the second set C2 features are added. These C2 features are obtained with the feature extraction hierarchy, so that the overall feature dimensionality increases to $F = F_{co} + F_a + F_{c2} = 16916$. The major difference between the offline and interactive learning experiments is that no sensory memory is required for the offline experiments. Additionally a simplified STM concept for the offline experiments is used, where all collected object views are stored into the limited STM, similar to the experiments described in (Kirstein et al., 2008).

5.1 Offline Categorization Experiments

We compare the categorization performance of our proposed cLVQ method with a single layer perceptron (SLP) for different databases and feature sets summarized in Fig. 7. We use the SLP for comparison because it is the simplest architecture for this category learning task. Therefore it characterizes the difficulty and the baseline performance of this learning task. Although the SLP is only a linear method for high-dimensional and sparse feature vectors it reaches similar results compared to more complex learning methods, at least if the STM is not limited (Kirstein, Wersing, & Körner 2008). The SLP output for each category is given as $out_c^{slp}(\mathbf{x}^i) = 1/(1 + \exp(-\mathbf{w}_c^{slp} * \mathbf{x}^i - \theta_c))$, where $\mathbf{w}_c^{slp}$ is a single linearly separating weight vector with bias θ_c for each category c. Training of the SLP consists of standard stochastic gradient descent in the sum of quadratic difference errors between training target and model output. In contrast to the more commonly used receiver operating characteristic (ROC) curves we estimate the rejection thresholds during the learning process to allow categorization of new object views at any time. This is an essential requirement for interactive learning tasks. The estimation of rejection thresholds is based on the average SLP output of category c calculated for training vectors $\mathbf{x}^i$ labeled with $t_c^i = +1$ and also for $\mathbf{x}^i$ labeled with $t_c^i = -1$. The rejection threshold for category c is then set to the mean value of both calculated values.

5.1.1 Experimental Setup

For the offline experiments two databases of the same training and test objects shown in Fig. 6 are collected using different experimental setups. Overall 24 objects for training and a complementary set of 24 test objects are collected for both databases. The objects of the first database are collected in front of a black background making foreground masks unnecessary. For each object 300 views are collected by rotating it around the vertical axis. We refer to this database in the following as *restricted database*. Although we call this a *restricted database* it already contains more appearance variations than databases of most other categorization approaches where typically only the canonical views are considered. Additionally some objects are multi-colored (e.g. some cars or boxes) where not only the base color should be detected, but also all other prominent colors, covering more than about one third of the visible object surface. This multi-detection constraint complicates the categorization task compared to the case where only the best matching category or the best matching category of a specified group of visual attributes (e.g. one for color and one for shape) must be detected. For the second database, called *unconstrained database* in the following, each object was freely rotated around three axes in front of our active camera system covering almost the complete viewing sphere. For the collection of this database we used the same preprocessing as proposed in Section 2. In contrast to the interactive learning the objects are shown by two different persons. This additionally increases the variability of object presentation. Overall 1200 segments and their corresponding foreground masks are collected for each object. Compared to the *restricted database* it is more complex because of much higher appearance variations of objects. The categorization task is also more challenging due to brightness variation, segmentation errors and imprecise foreground masks. All these effects cause additional fluctuations to the feature responses and therefore complicate the learning process. We refer to errors as segmentation errors if some foreground parts are missing in the corresponding segment, while imprecise foreground masks are related to background parts that are assigned to the foreground. Based on both categorization databases we incrementally learn and test five different color (red, green, blue, yellow, and white) categories and five different shape (rubber duck, cup, car, cell phone, box) categories. Finally it should be mentioned that all these effects causing strong fluctuations in the feature responses. These instable responses combined with very little training examples pose a considerable problem for any kind of category learning approach.

For the offline experiments we subdivided the learning of the category-specific LTM into learning epochs. At each epoch only the feature vectors of three different objects are visible to the learning architecture, emulating a capacity-limited STM. At the beginning of each epoch a randomly selected object is added to the STM, while the oldest object in the memory is removed. Based on the currently available feature vectors, the learning methods are used to incorporate this STM knowledge into the LTM by applying the learning dynamics of the cLVQ method described in Section 4.2.3. Additionally gradient descent with a predefined number of learning steps was performed for the SLP networks. Note that the SLP is trained based on the full feature vector $\mathbf{x}^i$, without any additional feature selection. After this training phase the current categorization performance is calculated based on all test objects to show the effect of the newly presented object to the categorization performance. Finally new learning epochs are started until all training objects were presented once to the learning system. Each object is shown only once during the training epochs, and does not reappear during training. In this way we investigate the life-long learning capability of our cLVQ architecture and its ability to approach the "stability-plasticity dilemma". For all experiments, the training set is changing over time due to the incremental learning task. For evaluation, however, the categorization performance is computed on the stationary set of all test objects with their target category labels. Additionally the categorization performance is averaged over all individual categories belonging to the group of color or shape categories respectly.

5.1.2 Categorization Results

The comparison of cLVQ and SLP for the *restricted database* is shown in the upper row of Fig. 7. For the evaluations, we show the categorization performance averaged over 10 runs. It can be seen that at the beginning of the training the SLP is superior to our proposed cLVQ method, but after presenting all training objects the cLVQ performs distinctly better for the color categories, while for the shape categories cLVQ is slightly better than the SLP architecture. Although the SLP performs worse than cLVQ it still performs surprisingly well, which is somehow contrary to classification tasks with a one-out-of-n class selection where the SLP approach is known for the "catastrophic forgetting effect" (French 1999). It seems to be that for our categorization task the independent representation of categories somehow weakens the forgetting effect of SLP networks. For a larger number of shape categories and training objects the performance improvement of cLVQ over SLP is clearly visible, as was shown in earlier experiments (Kirstein et al., 2008).

The addition of the C2 features to the vectors $\mathbf{x}^i$ increases the categorization performance of shape categories for the cLVQ and SLP method. Although the C2 feature representation is less category-specific, at least some of the local and topographically organized C2 features can be used to stabilize the representation of shape categories. However, for the color categories C2 features have the opposite effect causing a slight performance decrease for the cLVQ architecture. This basically results from C2 features that are dominantly active for many views of a certain object and therefore are selected to represent the color categories belonging to this object. Such general and object-specific C2 features are most probably also the reason for the strong performance loss of about 20% for the SLP color categories.

Also for the *unconstrained database* (see lower row of Fig. 7) the SLP is superior at earlier learning epochs where only a few objects were trained, while the cLVQ performs better at later learning stages. The cLVQ learning method is again distinctly better than SLP for color categories and slightly better for shape categories. The most distinctive difference to the *restricted database* experiments is the slow learning progress of shape categories resulting in poor categorization results. This is basically caused by the strong appearance variations of the objects under almost full in-depth rotation. Also segmentation errors make the learning of shape categories harder, because some parts of the objects are missing in those object views. Additionally also imprecise foreground masks cause problems for the category learning, because potentially also features extracted next to the object are used to incrementally learn the representations of categories. The appearance variations caused by full 3D object rotation induce strong fluctuations to the detection of shape features, complicating the forward feature selection process. This is caused by the fact that if there are almost no features with high scoring values the selection methods has to test many different features. Additionally the feature selection tends in such cases to select color features for the representation of shape categories, because they are the most frequent and stable ones. This is maybe also one reason for the poor generalization performance of shape categories. As a consequence the training takes typically much longer compared to the experiments with the *restricted database*, but also many more cLVQ nodes are allocated.

In contrast to this the categorization performance of color categories is equal to the experiments with the *restricted database*, because color histograms as feature representation for such categories are robust with respect to object rotation. The representation of color categories is additionally unaffected by segmentation errors, because even if object parts are missing in a segment the basic colors are typically still visible. For color categories the effect of imprecise foreground masks on the categorization performance seems also to be only minor, otherwise the performance would be considerably lower. This basically means that the occurrence of category related color features is more stable than de-

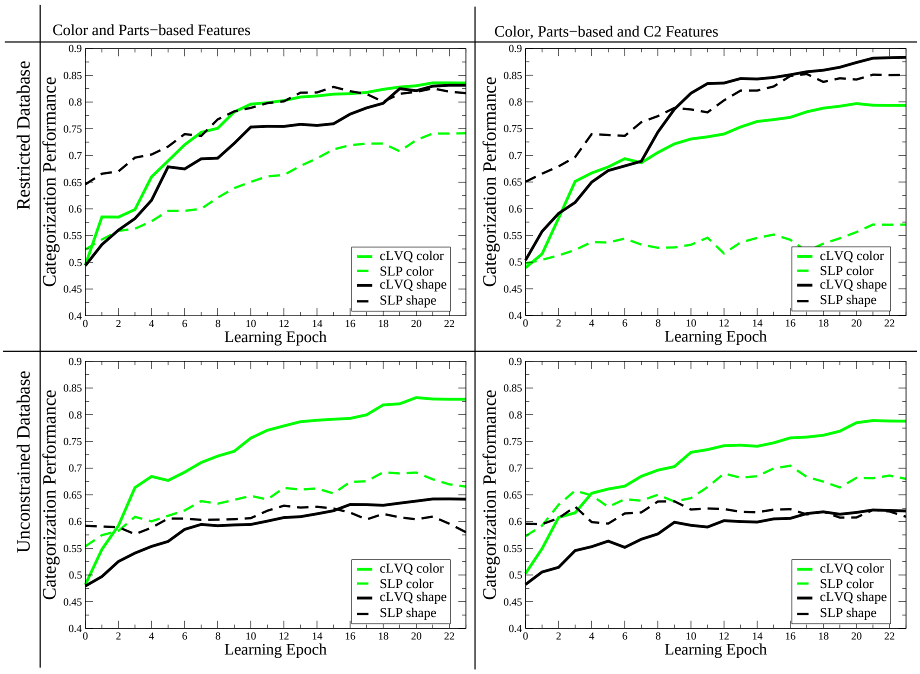


Figure 7: **Average Results of Offline Categorization Experiments.** The performance of our proposed cLVQ method and the SLP networks are compared for the *restricted database* and the *unconstrained database* using the same set of training and test objects (averaged over 10 runs). All results show the categorization performance on the test set, which was never seen during the training. The difference between both databases is that the objects for the *restricted database* are rotated around the vertical axis in front of a black background, while the *unconstrained database* was collected under relaxed constraints with cluttered background and full in-depth rotation of objects. Additionally we tested the effect of C2 features with respect to the categorization performance of shape categories. For all offline experiments the SLP method is superior at earlier learning stages, while the cLVQ is better at later learning steps. After the presentation of all training objects the cLVQ method performs distinctly better for the color categories compared to the SLP networks, while for the shape categories it is slightly better. The addition of C2 features to the feature representation increases the performance of shape categories only for the *restricted database*, while for the *unconstrained database* with much higher appearance variations no such performance changes could be measured.

tected features at background parts from the surrounding scene. For the shape categories this effect is very unlikely, because of much higher variations in the extracted shape features. Therefore the effect of imprecise foreground masks is for those categories most probable much stronger. If selected background features are reoccurring in both positive and negative category examples, then such features are weakened by the feature scoring mechanism or can be completely removed by the cLVQ learning dynamics. Although both mechanisms in general reduce the effect of wrongly selected features this typically require the presentation of a considerable amount of additional training examples. Finally for the *unconstrained database* no performance gain with respect to the shape categories could be found by additionally using C2 features. The reason is that a C2 feature is sensitive to a flexible shape primitive around one particular location in the segment (Wersing & Körner 2003), while the parts-based features are not tuned to a particular location. Therefore a single C2 feature can not provide object or category-specific information if the objects are rotated in depth.

5.2 Interactive Category Learning under Real-world Constraints

In comparison to the previously performed offline experiments an interactive learning scenario has the possibility of directly correcting errors based on tutor feedback, even if the object was already presented before. Although we do not impose any restrictions on the viewing angle of objects the appearance variations are less compared to the *unconstrained database*. This is basically because such variations can not be produced in a typical training session where the object is presented for about 30 seconds. The learning system with its different building blocks is distributed on four 3 GHz CPUs. The overall system including preprocessing, figure-ground segregation, feature extraction, category learning and user interaction runs roughly at the frame rate of our current digital camera system of approximately 6-8 Hz. This is fast enough to show the desired incremental and life-long learning ability of our categorization system.

In Fig. 8 a normal learning session is shown, where a representation of three different color and three different shape categories is learned in less than 8 minutes. We start with a completely empty STM and LTM representation, therefore the system responds for the first presented object with "unknown category". After the training of the first object it only knows the categories *yellow* and *duck* but at this state it can not separate both categories. Thus the system responds with "yellow duck" to the next presented green duck. Afterwards successively new objects are presented and trained. Usually after the presentation of 2-3 examples of a specific category the system can generalize to previously unseen objects, while still being able to correctly categorize already known objects. To check this, the yellow duck is also shown at a later learning stage, followed by two previously unseen toy cars. The presented white toy car is labeled as "toy car" because the category *white* is so far not known. It also shows that at this learning stage the different color and shape categories are automatically separated by the learning algorithm, which is a necessary precondition to

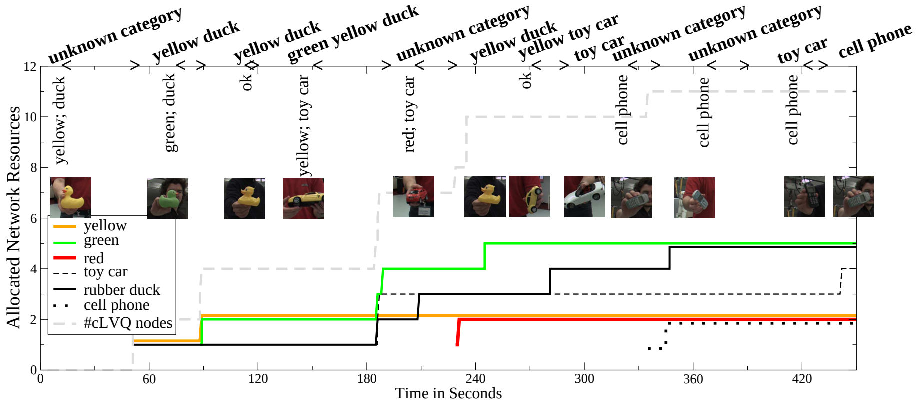


Figure 8: **Incremental Learning of Visual Categories.** The incremental selection of features for each category is shown over time, while presenting different objects. The model starts from a completely blank memory. Additionally the total number of cLVQ representatives is plotted, which are allocated during the interactive learning session. We also added the categorization decisions of the learning system, communicated to the user on top of the figure with sloped text. Furthermore the confirmed category labels provided by the user are denoted underneath. Note that "ok" means the confirmation of the categorization decisions on top of the figure and that not every time confirmed labels are provided. Additionally the intervals where new training vectors are collected into the STM are marked with <>. The transfer of the STM to the LTM occurs gradually according to the parallely running cLVQ and is not fully synchronized to the speech labels.

achieve a higher generalization performance compared to object identification. After the presentation of the white toy car the category *cell phone* is trained. It should be mentioned that the learning system responded in most cases with "unknown category", while the rejection of unknown objects typically cause major problems for object identification systems.

6 Discussion

We have presented a learning system able to interactively learn general visual categories in a life-long learning fashion. To our knowledge this is the first online learning system that allows category learning based on complex-shaped objects held in hand. In offline experiments we could show the difficulty of the learning of categories under real-world conditions by comparing the categorization performance of the same object set taken under different experimental setups. Nevertheless we are able to learn categories under such conditions in an interactive and life-long learning fashion. Comparable architectures as proposed by (Skočaj et al., 2007) or (Fritz, Kruijff, & Schiele 2007) learn categories based on objects placed on a table, which simplifies the ROI detection and figure-ground segregation. Additionally this constraint strongly reduces the appearance variations of the presented objects and therefore makes the category learning task much easier. We also allow different categories for a single object, while in related work typically the categories are trained independently.

We could show that our learning system can efficiently perform all necessary processing steps including figure-ground segregation, feature extraction and incremental learning. Especially the ability to handle high-dimensional but sparse feature vectors is necessary to allow interactive and incremental learning, where often additional dimension reduction techniques like the principal component analysis are required to allow online learning. This high feature dimensionality is also challenging for the used feature selection method, because of the large amount of possible feature candidates. Nevertheless the learning system is able to extract small sets of category-specific features out of many possible feature candidates.

References

Agarwal, S., Awan, A., & Roth, D. (2004). Learning to detect objects in images via a sparse, part-based representation. IEEE Transaction Pattern Analysis and Machine Intelligence 26(11), 1475–1490.

Arsenio, A. M. (2004). Developmental learning on a humanoid robot. In Proc. International Joint Conference on Neuronal Networks (IJCNN), pp. 3167–3172.

Denecke, A., Wersing, H., Steil, J. J., & Körner, E. (2009). Online figure-ground segmentation with adaptive metrics in generalized LVQ. Neurocomputing 72(7-9), 1470–1482.

French, R. M. (1999). Catastrophic forgetting in connectionist networks. Trends in Cognitive Sciences 3(4), 128–135.

Fritsch, J., Lang, S., Kleinehagenbrock, M., Fink, G. A., & Sagerer, G. (2002). Improving adaptive skin color segmentation by incorporating results from face detection. In Proc. IEEE Int. Workshop on Robot and Human Interactive Communication (ROMAN), Berlin, Germany, pp. 337–343.

Fritz, M., Kruijff, G.-J. M., & Schiele, B. (2007). Cross-modal learning of visual categories using different levels of supervision. In Proc. International Conference on Vision Systems (ICVS).

Fritzke, B. (1994). Growing cell structures - a self-organizing network for unsupervised and supervised learning. Neural Networks 7(9), 1441–1460.

Fritzke, B. (1995). A growing neural gas network learns topologies. In G. Tesauro, D. S. Touretzky, & T. K. Leen (Eds.), Advances in Neural Information Processing Systems 7, Cambridge MA, pp. 625–632. MIT Press.

Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biological Cybernetics 36(4), 193–202.

Furao, S. & Hasegawa, O. (2006). An incremental network for on-line unsupervised classification and topology learning. Neural Networks 1(19), 90–106.

Goerick, C., Mikhailova, I., Wersing, H., & Kirstein, S. (2006). Biologically motivated visual behaviours for humanoids: Learning to interact and learning in interaction. In Proc. International Conference on Humanoid Robots (Humanoids), pp. 48–55.

Guyon, I. & Elissee, A. (2003). An introduction to variable and feature selection. Journal of Machine Learning Research, 3, 1157–1182.

Hamker, F. H. (2001). Life-long learning cell structures–continously learning without catastrophic interference. Neural Networks, 14, 551–573.

Hammer, B. & Villmann, T. (2002). Generalized relevance learning vector quantization. Neural Networks 15(8-9), 1059–1068.

Hasler, S., Wersing, H., & Körner, E. (2007). A comparison of features in parts-based object recognition hierarchies. In Proc. International Conference on Artificial Neural Networks (ICANN), pp. 210–219.

Kirstein, S., Wersing, H., Gross, H.-M., & Körner, E. (2008). A vector quantization approach for life-long learning of categories. In Proc. International Conference on Neural Information Processing (ICONIP), pp. 803–810. Springer.

Kirstein, S., Wersing, H., & Körner, E. (2008). A biologically motivated visual memory architecture for online learning of objects. Neural Networks, 21, 65–77.

Kohonen, T. (1989). Self-Organization and Associative Memory. Springer Series in Information Sciences, Springer-Verlag, third edition.

Leibe, B., Leonardis, A., & Schiele, B. (2004). Combined object categorization and segmentation with an implicit shape model. In ECCV workshop on statistical learning in computer vision, pp. 17–32.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60(2), 91–110.

Mikolajczyk, K., Leibe, B., & Schiele, B. (2006). Multiple object class detection with a generative model. In Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Ozawa, S., Toh, S. L., Abe, S., Pang, S., & Kasabov, N. (2005). Incremental learning of feature space and classifier for face recognition. Neural Networks 18(5-6), 575–584.

Pomierski, T. & Gross, H.-M. (1996). Biological neural architecture for chromatic adaptation resulting in constant color sensations. In Proc. IEEE International Conference on Neural Networks (ICNN), pp. 734–739.

Roth, P. M., Donoser, M., & Bischof, H. (2006). On-line learning of unknown hand held objects via tracking. In Proc. Second International Cognitive Vision Workshop (ICVW).

Schneider, P., Biehl, M., & Hammer, B. (2007). Relevance matrices in LVQ. In Similarity-based Clustering and its Application to Medicine and Biology, Number 07131 in Dagstuhl Seminar Proceedings. Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany.

Skočaj, D., Berginc, G., Ridge, B., Štimec, A., Jogan, M., Vanek, O., Leonardis, A., Hutter, M., & Hewes, N. (2007). A system for continuous learning of visual concepts. In Proc. International Conferance on Vision Systems (ICVS).

Skočaj, D., Kristan, M., & Leonardis, A. (2008). Continuous learning of simple visual concepts using incremental kernel density estimation. In Proc. International Conference on Computer Vision Theory and Applications (VISAPP), Funchal, Madeira, Portugal, pp. 598–604.

Steels, L. & Kaplan, F. (2001). AIBO's first words. The social learning of language and meaning. Evolution of Communication 4(1), 3–32.

Swain, M. J. & Ballard, D. H. (1991). Color indexing. International Journal of Computer Vision 7(1), 11–32.

Thomas, A., Ferrari, V., Leibe, B., Tuytelaars, T., Schiele, B., & Gool, L. V. (2006, June). Towards multi-view object class detection. In Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), New York, USA.

Wersing, H., Kirstein, S., Götting, M., Brandl, H., Dunn, M., Mikhailova, I., Goerick, C., Steil, J., Ritter, H., & Körner, E. (2007). Online learning of objects in a biologically motivated architecture. International Journal of Neural Systems, 17, 219–230.

Wersing, H. & Körner, E. (2003). Learning optimized features for hierarchical models of invariant object recognition. Neural Computation 15(7), 1559–1588.

Willamowski, J., Arregui, D., Csurka, G., Dance, C. R., & Fan, L. (2004). Categorizing nine visual classes using local appearance descriptors. In Proc. ICPR Workshop on Learning for Adaptable Visual Systems.