Stephan Kirstein

Interactive and life-long learning for identification and categorization tasks

Interactive and life-long learning for identification and categorization tasks

Stephan Kirstein

Universitätsverlag Ilmenau
2010

Impressum

Bibliografische Information der Deutschen Nationalbibliothek
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Angaben sind im Internet über http://dnb.d-nb.de abrufbar.

Diese Arbeit hat der Fakultät für Informatik und Automatisierung der Technischen Universität Ilmenau als Dissertation vorgelegen.

Tag der Einreichung:	24. September 2009
1. Gutachter:	Prof. Dr. Horst-Michael Groß (Technische Universität Ilmenau)
2. Gutachter:	Prof. Dr. Edgar Körner (Honda Research Institute Europe GmbH)
3. Gutachter:	Prof. Dr. Martin Riedmiller (Albert-Ludwigs-Universität Freiburg)
Tag der Verteidigung:	15. April 2010

Technische Universität Ilmenau/Universitätsbibliothek
Universitätsverlag Ilmenau
Postfach 10 05 65
98684 Ilmenau
www.tu-ilmenau.de/universitaetsverlag

Herstellung und Auslieferung
Verlagshaus Monsenstein und Vannerdat OHG
Am Hawerkamp 31
48155 Münster
www.mv-verlag.de

ISBN 978-3-939473-80-0 (Druckausgabe)
urn:nbn:de:gbv:ilm1-2010000131

Titelfoto: photocase.com

Abstract

The presented thesis focuses on life-long and interactive learning for identification and categorization tasks. The fundamental and still largely unsolved problem of life-long learning with artificial neural networks is the so-called "stability-plasticity dilemma". To achieve plasticity the learning approach must be able to continuously integrate newly acquired knowledge into its internal representation, while for the stability the conservation of this knowledge is required.

To achieve interactive learning for difficult recognition problems the separation into an intermediate and fast learning short-term memory (STM) and a slower learning long-term memory (LTM) is proposed. For the incremental build up of the STM a similarity-based one-shot learning method was developed. Furthermore two different memory consolidation algorithms were proposed enabling the incremental learning of LTM representations for various identification and categorization tasks. For identification tasks different modifications to the Learning Vector Quantization (LVQ) network architecture are proposed. The major changes of the LVQ approach are an error-based node insertion rule and a node dependent learning rate. Both extensions enable incremental and life-long learning for identification problems. For life-long learning of categories this extended LVQ model was combined with a forward-feature selection method. This selection method incrementally extracts small sets of category-specific features and therefore allows for a compact representation of categories.

In several interactive and offline recognition experiments the efficiency and performance of the proposed learning methods could be shown for difficult visual recognition problems. Additionally an active vision system was developed that utilizes the proposed learning methods. This integrated system enables learning of visual representations based on natural and complex-shaped objects presented in hand.

Zusammenfassung

Die vorgelegte Arbeit beschäftigt sich mit lebenslangem und interaktivem Lernen von Identifikations- und Kategorisierungsaufgaben. Das grundlegende Problem von lebenslangem Lernen mit Hilfe künstlicher neuronaler Netze ist das sogenannte "Stabilitäts-Plastizitäts Dilemma". Um die Plastizität zu gewährleisten muss ein neuronales Netz in der Lage sein, neues Wissen zu erwerben. Zusätzlich muss für die Stabilität dieses Wissen vor dem Verlust bewahrt werden.

Um die Fähigkeit des interaktiven Lernens zu gewährleisten wurde die Aufteilung in eine temporäres und schnell lernendes Kurzzeitgedächtnis und ein langsamer lernendes Langzeitgedächtnis vorgeschlagen. Zum inkrementellen Aufbau des Kurzzeitgedächtnisses wurde ein ähnlichkeitsbasiertes und sogenanntes "one-shot learning" Verfahren vorgeschlagen. Für die Konsolidierung dieser Repräsentation in das Langzeitgedächtnis wurden zwei verschiedene Methoden betrachtet, die für das Lösen von nahezu beliebigen Identifikations- und Kategorisierungsaufgaben geeignet sind. Um diese Lernfähigkeit für Identifikationsprobleme zu erreichen, wurden verschiedenen Erweiterungen für die sogenannte "Learning Vector Quantization (LVQ)" Netzwerkarchitektur vorgeschlagen. Die wichtigstens Änderungen sind die Einführung einer fehlerbasierten Knoten-Einfügeregel und die Verwendung einer knotenspezifischen Lernrate. Beide Erweiterungen ermöglichen inkrementelles und lebenslanges Lernen für Identifikationsaufgaben. Im Gegensatz dazu wurde für das Lernen von Kategorien dieses modifizierte LVQ Netzwerkmodell mit einem Verfahren zur Merkmalsselektion kombiniert. Dieses Verfahren selektiert kategoriespezifische Merkmalsdimensionen und ist daher geeignet um besonders kompakte Kategorierepräsentationen zu lernen.

In verschiedenen Erkennungsexperimenten konnte die Effizienz und die Leistungsfähigkeit der vorgeschlagenen Verfahren belegt werden. Zusätzlich wurde ein integriertes Erkennungssystem entwickelt, welches die vorgeschlagenen Verfahren einsetzt. Dieses Gesamtsystem ermöglicht das interactive Lernen visueller Repräsentationen von beliebigen Gegenständen, welche, in der Hand gehalten, dem System gezeigt werden.

Acknowledgments

The presented work was a joint project between the Honda Research Institute Europe GmbH (HRI-EU) in Offenbach and the Neuroinformatics and Cognitive Robotics Lab at the Ilmenau University of Technology. I thank the heads of both institutes Prof. Dr. Edgar Körner and Prof. Dr. Horst-Michael Gross for giving me the possibility to work on this interesting research topic.

I also want to thank my project leader and supervisor Dr. Heiko Wersing for many fruitful discussions, valuable advise and help, but also for his constructive criticism. Additionally I would like to thank him and Stephan Hasler for providing the shape feature extraction methods used in this dissertation.

Many thanks go also to my colleagues at both institutes. Specifically I want to thank the team members of the Learning of Sensory Representations group at the Honda Research Institute Europe GmbH: Dr. Heiko Wersing, Dr. Mathias Franzius, Alexander Denecke, Stephan Hasler and Samuel John for the productive collaboration and the good working atmosphere.

Furthermore, I want to thank all readers of this manuscript, including Prof. Dr. Horst-Michael Gross, Dr. Heiko Wersing, Dr. Martina Hasenjäger and Dr. Tobias Rodemann, for their comments and valuable hints for the improvement of this work.

Finally I would like to thank all my friends for their support in the hot phases of this dissertation. I also want to thank them for motivating me to continue with the work on this thesis. Many thanks go also to my mother, my sister and my stepfather for their support and encouragement.

Contents

Chapter 1

Introduction

The progress in computer science and engineering was enormous in the recent decades. This development is especially visible if you compare the first digital computer Z3 developed by Konrad Zuse with currently available personal computers. Together with the evolution of the computer hardware also the complexity of software increased continuously. This development started with very simple mathematical programs and led to large integrated software systems with several billion lines of code. Although the progress in computer hardware and software went through dramatic changes in the past, comparably little advances has been made with respect to the modeling of cognitive capabilities like scene understanding, object identification or categorization.

These abilities are so easy for humans, that we rarely notice its importance in carrying out everyday exercises. One fundamental requirement for achieving cognitive functions is the ability to learn arbitrary representations. With respect to visual knowledge, especially the robustness and the capacity for memorizing countless objects and categories during the entire life makes the human visual system superior to all currently existing technical approaches. Furthermore also the ability to transfer memorized information and skills to completely different situations is an impressive capability of humans, where artificial cognitive systems typically can only be applied to their prespecified functions.

One aspect with respect to the distinctly slower progress for creating brain-like intelligence is that the research on artificial cognitive functions only recently began to focus more on integrated systems rather than individual functions. Nevertheless there is still the focus on individual parts like the feature extraction or the learning with neural networks. Although in isolated research fields a continuous scientific progress is visible, without the combination of many of such individual pieces from different research fields no artificial system with human-like performance will soon be achieved. This basically is because also the human brain utilizes different situation-dependent processing principles and memory representations to achieve higher cognitive functions. A further example is the development of complex humanoid robots. These robots can perform complex movements such as walking or running, but still are unable to learn relatively simple tasks like coffee brewing or understanding their surrounding environment.

Another reason for this slow progress is that the underlying higher cognitive mechanisms in the human brain are still largely unknown. Especially with respect to the formation of new memories and the extraction of knowledge only the basic principles are known so far. This rough understanding of the fundamental principles are a good starting point for the development of similar cognitive functions, but for the detailed and biologically realistic modeling further information is required. Although the precise mechanisms of knowledge acquisition are largely unknown, nevertheless it is one key component of many human abilities. Specifically it is important that this capability is open-ended, allowing learning during the complete life-time of a human. Furthermore it is known that the formation of memory is influenced by behavioral relevance. This strongly influences when new information is acquired, but also at which position in the representation this knowledge should be incorporated.

This learning capability inspired different scientific disciplines including research on artificial neural networks. The most important capability of these networks is the ability to learn arbitrary input-output mappings and therefore can be applied to a large variety of problems. Of particular interest is the learning based on pairs of input and target vectors, because this enables the approximation of problems where the critical parameters and working principles are largely unknown. The learning ability of these mathematical models is accomplished by adapting their weights or more general their internal representation based on a given optimization criterion. Especially this automatic adaptation of the network representation makes this group of learning approaches interesting for identification or categorization tasks. For these problems commonly the critical parameters (e.g. the most distinctive features to discriminate a group of objects) of the desired input-output mapping are unknown in advance, but artificial neural networks are able to approximately extract these determining factors. Therefore these learning methods can in general be utilized for the development of cognitive abilities. Nevertheless the major drawback of neural networks is that they are commonly separated into a training and test phase. Only during the training phase the network weights are adaptable, but in the test and operation phase the parameters representing the learned knowledge are fixed. This is sufficient for several engineering tasks, but for the creation of cognitive functions often a continuous and open-ended learning is required.

In the presented dissertation we therefore focus on open-ended and continuous learning, which is commonly termed as life-long learning. We believe that such a learning capability is one crucial requirement to build intelligent systems with human-level performance. Also for humans these impressive recognition abilities are not innate, but can be acquired by continuously learning and interacting with the environment. Therefore in the following life-long learning methods are proposed that can be utilized for arbitrary identification and categorization tasks. Additionally we concentrate on the integration of these methods into larger active vision systems. This integration combines learning with further visual and motor abilities to realize functions that can not be achieved with the individual system parts alone. Of particular interest in this dissertation is the development of vision systems that enable interactive learning in cluttered environments based on complex-shaped objects presented naturally held in hand. Thus we emphasize that the proposed learning algorithms are applicable in unconstrained and unpredictable environments, which normally can not be achieved with common artificial neural networks. We suppose that the combination of life-long learning approaches and their applicability in unconstrained environments provide the basis for the development of further cognitive functions.

1.1 Problem Description

Human beings are able to acquire and maintain knowledge during their complete lifetime. This outstanding ability is called life-long learning (Bagnall, 1990). In contrast to this artificial neural networks are typically only adapted during their learning phase and the weights, presenting the learned knowledge, are fixed afterwards. Such a fixed learning architecture can be powerful in constrained and statical environmental settings but may not be suitable for technical applications like assistive robots or interactive agents. This is because these systems require a continuous error correction and need to enlarge their knowledge base to operate in changing and unpredictable environments.

The fundamental problem of life-long learning with artificial neural networks is the “stability-plasticity dilemma” (Carpenter & Grossberg, 1987). Here the term plasticity refers to the ability of a learning system to incorporate new acquired knowledge into its internal representation. One common solution to achieve this plasticity is the usage of incremental neural networks like the Growing Neural Gas (Fritzke, 1995). For this neural network architecture the training process starts with a minimal network and iteratively increases the network size based on some insertion criteria. Thus the final dimensionality reflects the complexity of the current learning task. However, the already learned knowledge should be preserved to guarantee the stability of previously acquired information. In contrast to the incorporation of new knowledge based on incremental learning techniques the preservation of knowledge is still an unsolved problem. This stability problem especially occurs if the network model is trained with a limited and changing training ensemble, which is common for life-long learning tasks because it is unfeasible to store all experiences that are encountered during the complete operation time of the system. As a consequence the training with such data ensembles typically causes the well-known “catastrophic forgetting effect” (French, 1999), which means that with the incorporation of newly acquired knowledge, parts of the previous learned information is quickly fading out.

Commonly life-long learning architectures are approaching the so-called “stability-plasticity dilemma”, but the requirements for such learning methods are also dependent on the targeted recognition task. For identification tasks, where the target is the separation of a specific instance (e.g. a physical object) from all other instances, the combination of incremental learning methods with stability considerations of matured network parts are typically sufficient. In contrast to this for categorization tasks the mapping from several instances to a shared attribute (e.g. the basic shape) is learned. This means for the example of visual categorization, where the individual objects (e.g. red-white car) typically belong to several different categories, a decoupled representation for each category (for category “red”, “white” and “car”) has to be learned. This disengagement leads to a more condensed representation and a higher generalization performance compared to object identification architectures and can be achieved with additional metrical adaptation or feature selection methods. However, due to the fact that exemplars of a category are incrementally presented, considerable changes to feature weighting and selection can occur. Therefore for categorization tasks a balance between the stability of knowledge and the correction of wrong category representations must be found, which complicates the learning of such representations compared to identification tasks. Finally for feature weighting and selection methods no a priori knowledge with respect to the extracted feature modalities should be used to allow learning of arbitrary categories based on the optimal feature combinations.

1.2 Definition of Common Terms

In the following the most commonly used terms are defined. These definitions are especially important because some of these terms are widely used in literature. Therefore it is necessary to clarify their meaning in the presented dissertation.

- **Online Learning:** This term or its synonym real-time learning is used for the ability of fast learning and immediately recognizing trained stimuli. This capability mainly applies to “one shot learning” methods like the Adaptive Resonance Theory (ART) (Carpenter et al., 1991). A special property of online learning is the possibility of active learning with an interactive correction of errors during the training process.

- **Interactive Learning:** We use the term interactive learning for the ability of an recognition system to acquire knowledge in direct interaction with a human tutor. To achieve this property a fast learning algorithm is essential to allow a rapid incorporation of newly acquired information into the overall knowledge. In general this capability can be achieved with online learning methods but also fast iterative learning approaches are possible.

- **Incremental Learning:** This term refers to the ability of a network architecture to allocate increasing numbers of neurons, dependent on the complexity of the current task. Such network architectures are normally initialized with a minimal number of neurons and are able to add resources based on some node insertion criteria using the training error.

- **Life-Long Learning:** We define life-long learning as the ability to continuously acquire new knowledge. This property of a neural network can be achieved by combining incremental learning with additionally considering the stability of already learned knowledge. Specifically the maintenance of the long-term stability is a fundamental problem of artificial neural networks that is still largely unsolved. This problem especially occurs if a changing and non-stationary training set is used, where only a portion of data is visible to the learning method.

In the presented dissertation we propose life-long learning methods for different recognition tasks. Therefore in the following a short definition of the terms identification and categorization task is given.

- **Identification Tasks:** For identification tasks several one-to-one stimulus response mappings are learned, where all entities (e.g. an object view) of an instance (e.g. a physical object) are mapped to a certain class label. Every time a new entity arrives the highest activated output neuron is selected in a one-out-of-n fashion and the new entity is assigned to the corresponding label of this neuron.

- **Categorization Tasks:** In contrast to identification, for categorization tasks the learning of a group of instances to a common response (many-to-one mapping) is required. Typically these group of instances share several common properties (e.g. a visual similarity or a similar behavior response). New entities are therefore assigned to an arbitrary number of properties. These properties are called categories in the following.

1.3 Inspiration from Biology

One inspiration from biological findings that influenced the presented work comes from several studies (Palmeri & Gauthier, 2004; Tarr & Bülthoff, 1998) showing strong evidence in favor of view-dependent and appearance-based representations in the human brain, as opposed to a strongly structuralist representation using three-dimensional primitives (Biederman, 1987). In the following a short review of further related findings from psychology and neuroscience is given with respect to insights in memory formation and life-long learning.

1.3.1 Memory and Learning in the Human Brain

There is still an ongoing debate about the brain regions and mechanisms that are essential for the short-term (STM) and the long-term memory (LTM) system in human brains (Ranganath & Blumenfeld, 2005). Nevertheless the separation of memory into STM and LTM is an established concept (Izquierdo et al., 1999) in psychology and neuroscience. LTM and STM, being optimized for different tasks, can be distinguished with regard to the level of detail, the number of items that can be stored and the time span the information can be memorized. The storage capacity of the STM is limited to a few items and the information can be memorized only for a relatively short period compared to the LTM, which can represent an enormous amount of information for long periods.

Furthermore there is a common distinction with regard to the learning speed in different brain regions. The defining property of the medial temporal lobe (MTL), including the hippocampus, is the ability to learn fast and immediately recognize a once presented stimulus, even if this stimuli was completely unknown before (O'Reilly & Norman, 2002). In technical applications this capability is often called "one shot learning". Recently it also has been affirmed that the MTL is not only important for spatial memory, but also for a non-spatial task of object recognition (Broadbent et al., 2004). In contrast to this the neocortex learns much slower, but the consolidation process between MTL and neocortex results in a reduction of the representational effort and a more generalized structure of the presented stimuli is extracted (O'Reilly & Norman, 2002). The primary questions concerning this separation are (McClelland et al., 1995):

- What is the benefit of the separation into MTL and neocortex?
- Why is new information not directly encoded into the neocortex, especially because most tasks crucially rely on these representations?
- Why is the learning process in the neocortex so slow compared to the MTL?

One opinion regarding this questions comes from the research on artificial neural networks, where the combination of a fast learning STM and a slow learning LTM is considered to be one solution to avoid the "catastrophic forgetting effect" (McClelland et al., 1995; French, 1999). We also consider this memory separation important for life-long learning with artificial neural networks, therefore in the following the memory consolidation process for the information transfer between these different memory systems is discussed in more detail.

1.3.2 Memory Consolidation

Already more than a century has passed since Müller & Pilzecker (1900) proposed the memory consolidation hypothesis. They found that memory of newly learned information is initially in an unstable state that can easily be destroyed. Another fundamental work of this early research on memory consolidation was done by the psychologist Ribot who first suggested that memories might be gradually reorganized over time (Ribot, 1881). He found out that the memory loss following a brain damage typically effects recent memories more than distantly acquired memory. This time-dependent effect became later known as Ribots gradient (Frankland & Bontempi, 2005).

A large body of experimental evidence beginning from classical work of Scoville & Milner (1957) shows, that the medial temporal lobe (MTL) is involved in the transfer of information from STM to LTM, with first changes due to learning occurring in the hippocampus (Wirth et al., 2003). However, based on studies of patients with temporally-graded retrograde amnesia the role of the medial temporal lobe and especially the hippocampus is assumed to be only temporary. After successful storing of contents in the neocortex, the LTM becomes gradually independent of the MTL structures (Squire & Zola-Morgan, 1991).

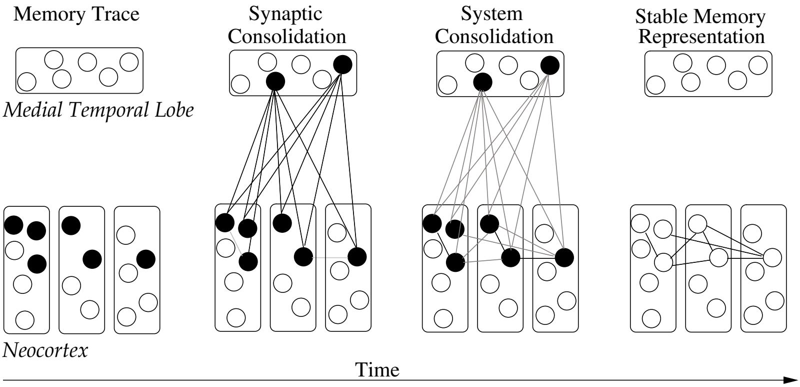


Figure 1.1: **Standard Memory Consolidation Model.** Starting point of the memory consolidation process is a memory trace in distributed neocortical regions. This trace is based on synaptic consolidation encoded and maintained in the medial temporal lobe (MTL), where especially the hippocampus plays a crucial role. This consolidation phase is commonly considered to be quite fast compared to the system consolidation that can take days to weeks to finish. During the system consolidation successive reactivation of the hippocampal-cortical network occurs that gradually leads to a strengthening or establishing of new cortical-cortical connections. Furthermore this successive reactivation causes the memory to become over time more stable and also independent from the MTL.

The current standard model of memory consolidation (Frankland & Bontempi, 2005) is illustrated in Fig. 1.1 and can at least be divided into two different phases (Medina et al., 2008). The first phase is called synaptic consolidation, which is considered to last minutes up to a few hours. Several studies have investigated this synaptic consolidation and there is increasing evidence that the creation and maintenance of hippocampal long-term potentiation (LTP) during this first phase is a precondition for memory consolidation in the neocortex (McGaugh, 2000). The second phase is called system consolidation and requires a much broader time scale to finish. It is assumed that reactivation of hippocampal memory traces are required for the reinstatement of neural activity patterns in the cortex. This reinstatement leads to a subsequent stabilization and refinement of the cortical traces (McGaugh, 2000). These processes have the effect that the internal structure of cortical connections are changed over time and after this reorganization process the memory becomes gradually independent of hippocampus and related structures (Bontempi et al., 1999). This consolidation process necessarily requires the synthesis of different structure proteins and also transcription factors, which is most probably one reason for the slow learning progress in the neocortex. Additionally it is known that several modulatory mechanisms influence the formation of memory like the emotional arousal that is connected to a part of the subcortical brain region amygdala.

Finally it can be stated that the early memory consolidation mechanisms in the hippocampus are partially understood, but the knowledge about how this information is transformed into life-long memories in cortical networks is much weaker (Frankland & Bontempi, 2005). This is due to the fact that system consolidation can take several weeks or longer. Therefore it is difficult to distinguish changes in the metabolism of the brain due to new experiences from the final stages of the system consolidation process. Furthermore with respect to memory consolidation there is evidence that for some kind of memories sleep is beneficial for memory formation in the neocortex (Maquet, 2001; Buzsáki, 1996).

1.3.3 The Cholinergic System

The cholinergic system is one of the phylogenetically old modulatory systems that influences the brain by a diffuse projection of acetylcholine (ACh) into the extracellular space of many brain regions. In general it is known that the concentration of ACh increases with attention and novelty of sensory stimuli (Hasselmo & McGaughy, 2004). Although there are many effects assigned to the cholinergic system (Sarter et al., 2003) in the following we concentrate on the effect of this system to the memory formation and its modulatory influence on synapses.

It is known that cholinergic modulation enhances long-term potentiation (Patil et al., 1998) and therefore high ACh concentrations are beneficial with respect to the temporary storage of new information in the MTL. On the contrary a low concentration of ACh enables the consolidation process by a disinhibition of cortico-cortical connections. As a consequence this substance influences the memory formation by switching between modes of acquisition and consolidation (Hasselmo & McGaughy, 2004). Furthermore this modulatory influence is known to reduce the interference of new representations with already stored knowledge by stronger suppressing previously potentiated synapses compared to naive synapses (Linster et al., 2003). In technical terms this effect can be interpreted as a modulation of the learning rate for each synapse with the effect that the stronger a synapse was potentiated in the past the smaller the learning rate gets. Therefore this modulatory effect of the cholinergic system supports life-long learning by reducing the so-called "catastrophic forgetting effect" in the neocortex.

1.4 The Scope of this Dissertation

We are targeting for interactive learning that requires fast algorithms and an efficient storage of information. However, the interactive learning and the targeted ability to solve complex recognition tasks is contradictory. Therefore we propose an intermediate and fast learning short-term memory (STM) representation to relax this conflict. The learning of the STM representation utilizes a similarity-based one-shot learning method and is combined with a previously developed feature extraction hierarchy. Based on this combination we could show interactive learning of many different natural objects under full 3D rotation. Although good recognition performance can be achieved with this learning method the high representational costs for storing a large number of representatives is the major drawback.

Therefore two different life-long learning approaches are proposed enabling the memory consolidation from the limited and continuously changing STM into a condensed long-term memory (LTM) representation. The proposed memory consolidation methods are the major contribution of the presented dissertation and can in general be applied to arbitrary identification and categorization tasks. For both recognition tasks different modifications to the Learning Vector Quantization (LVQ) (Kohonen, 1989) network architecture are suggested. For the identification tasks we propose a novel rule for incrementally allocating LVQ prototype nodes to efficiently adapt the network size to the difficulty of the recognition task. Furthermore a node-dependent learning rate is suggested to maintain the stability of matured network parts. Based on these modifications we could show incremental and life-long learning of many complex-shaped objects with a good identification performance.

For the more challenging task of incremental learning of multiple categories we propose to combine an LVQ-based learning method with a category-specific feature selection. The vector quantization part utilizes similar modifications as suggested for the identification problems. Additionally the feature selection enables several metrical "views" on the representation space of each individual prototype node. This capability is important to efficiently separate cooccuring categories, because natural objects commonly belong to different categories. Similar to the allocation of new prototype nodes also the category-specific features are incrementally learned based on a dynamic feature weighting procedure. The incremental acquisition of categories therefore requires a balance between the correction of wrong representations and the stability of the learned knowledge. This makes the incremental learning of categories distinctly complicated compared to identification problems with respect to approaching the "stability-plasticity dilemma". Nevertheless we could demonstrate the efficiency of our approach for a challenging categorization tasks, where the learning is applied for several complex-shaped and rotated objects.

With respect to the learning capabilities of our proposed methods we also focused on the scalability with regard to the overall feature dimensionality and the difficulty of the learning task. This scalability is especially important for the integration of the proposed learning methods into active vision systems. Therefore we suggest to integrate further visual and motor functionalities to enable learning based on complex-shaped objects held in hand as illustrated in Fig. 1.2. Based on this integration we could show high-performance object identification and learning of visual categories in unconstrained and changing environmental conditions. Interactive learning under such experimental conditions is typically more challenging, whereas research on life-long learning normally concentrates on offline learning only. We suggest that the possibility of life-long learning in such unconstrained environments is one major step with respect to the development of artificial cognitive systems with human-level performance.

1.5 Structure of this Thesis

The presented dissertation is structured in the following way (see Fig. 1.3). In the next Chapter 2 a brief outline over standard neural network archi-

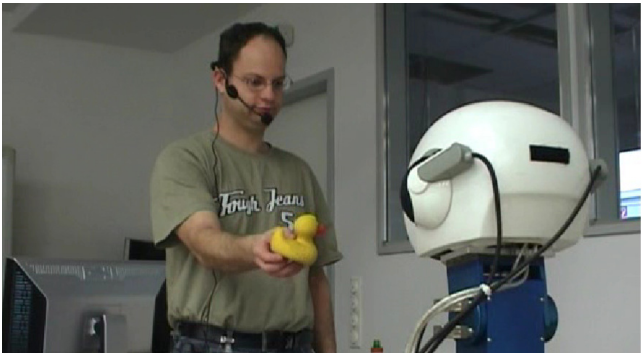


Figure 1.2: **Interactive Learning of Visual Representations.** The interactive learning of different visual representations is based on complex-shaped and hand-held objects presented in the near range of an active vision system. Furthermore the corresponding class hypothesis of the current presented object is communicated based on speech phrases, while a human tutor can confirm or correct this hypothesis based on the integration of a speech recognition system into a state-based user interface.

tectures is given, where we concentrate on the basic functionality and their applicability to life-long and interactive learning problems. The major contribution of this dissertation is the proposal of novel life-long learning approaches for different learning tasks. In Chapter 3 we first concentrate on life-long learning for identification tasks. The proposed learning model is based on the interaction between an online learning STM and a memory consolidation process for the learning of the corresponding LTM representation. In the following Chapter 4 a life-long learning categorization algorithm is proposed that combines incremental learning of an exemplar-based neural network with category-specific feature extraction. This proposed learning method enables the extraction of a category-specific LTM representation based on the object-specific STM model proposed in Chapter 3.

After the introduction of the different life-long learning methods for identification and categorization tasks we concentrate in Chapter 5 on the integration of these methods into a larger vision system. Besides the different building blocks that are required to allow interactive learning in unconstrained environments, we also investigate the scalability of the proposed learning methods with respect to the overall feature dimensionality and the difficulty of the learning task. Finally the achieved

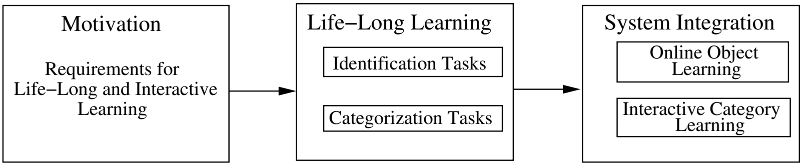


Figure 1.3: **Structure of this Thesis.** The presented dissertation is roughly subdivided into three parts. The first part relates standard neural network models to the requirements of life-long and interactive learning. In the second part we concentrate on life-long learning methods to solve identification and categorization tasks that is the major contribution of this dissertation. After the introduction of the learning approaches in the last part a vision system was realized that allows identification or categorization of hand-held objects in direct interaction with a human tutor.

results are summarized in Chapter 6, where additionally an outlook is given.

Chapter 2

Life-Long Learning with Standard Neural Network Architectures

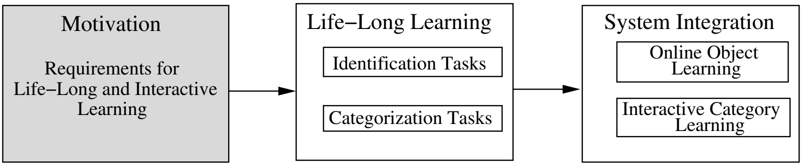


Figure 2.1: **Requirements for Life-Long and Interactive Learning.** In the following chapter the capabilities of standard neural network architectures are discussed with respect to their usability for life-long and interactive learning tasks.

The topic of this dissertation is life-long learning for identification and categorization tasks. Before the proposed approaches are described in the next chapters we give a brief overview over standard neural network models. The selected models are commonly used neural networks and therefore can be applied to various learning tasks. For this overview we compare the network topology and the basic learning principles. Additionally we discuss the usability of these network models with respect to life-long learning, which requires the consideration of the so-called “stability-plasticity dilemma” (Carpenter & Grossberg, 1987). Besides life-long learning we are also targeting for fast and interactive learning. Therefore an additional important aspect of this overview is the required training time. Finally we constitute the selected artificial neural network model that is the foundation of the proposed life-long learning methods.

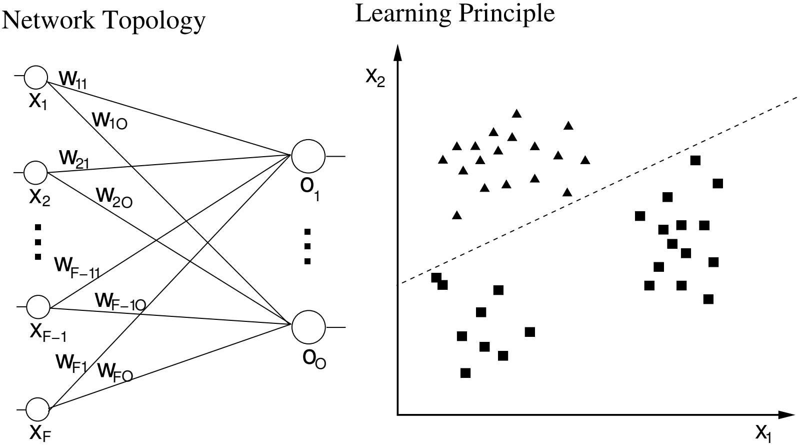


Figure 2.2: **Single Layer Perceptron.** The Single Layer Perceptron (SLP) is composed of an input and an output layer. The network output of neuron o is calculated based on an activation and transfer function. As activation function typically the scalar product between the weights $\mathbf{w}$ and the input vector $\mathbf{x}^i$ is used, where commonly a Fermi function is used as transfer function. Based on this network topology an SLP is able to learn any linearly separable problem, which is illustrated on the right for a two-class problem.

2.1 Single Layer Perceptron

The Perceptron or Single Layer Perceptron (SLP) is the simplest form of a neural network (Haykin, 1994) that can learn arbitrary linear functions (Minsky & Papert, 1969). Such an SLP consists of an input layer and an output layer as illustrated in Fig. 2.2. The biologically inspired Perceptron (Rosenblatt, 1958) was proven to converge to a hyperplane if the classes are linear separable (Rosenblatt, 1962). Based on a theoretical investigation it is known that the percentage of linear separable functions with respect to the total number of possible functions rapidly decays if the number of input neurons is increased (Wassermann, 1989). Nevertheless for high-dimensional and sparse feature representations a good generalization performance can be experimentally achieved compared to more complex learning models (Kirstein et al., 2008). The output p_o^i for each neuron o in the output layer is calculated based on scalar product activation and a transfer function Φ (e.g. a Fermi function) in the following way:

$$p_o^i = \Phi(\mathbf{x}^i \mathbf{w}_o). \tag{2.1}$$

Here $\mathbf{x}^i = (x_1^i, \ldots, x_F^i)$ is the training vector corresponding to input pattern i with an overall feature dimensionality F. Furthermore $\mathbf{w}_o = (w_{1o}, \ldots, w_{Fo})$ are the network weights from the input layer to the output neuron o. Commonly SLP networks are trained with a gradient descent method based on the generalized delta learning rule (Rumelhart et al., 1986):

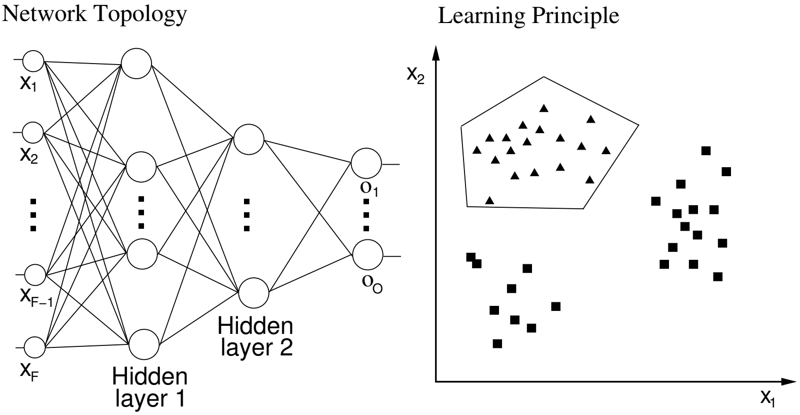
$$\Delta w_{fo} = \Theta \; (t_o^i - p_o^i) \; \Phi'(p_o^i) \; x_f^i, \tag{2.2}$$

where Θ is the learning rate and t_o^i is the teach signal for input pattern i and output neuron o. Additionally the $\Phi'(p_o^i)$ is the first derivative of the transfer function and x_f^i is the feature activation of feature f.

The major advantage of SLP networks is the fast learning ability. Thus the SLP architecture is applicable for interactive learning tasks, especially if sparsely activated features are used (Wersing et al., 2008). Nevertheless with respect to life-long learning problems, were typically the training set is limited and continuously changing, this network model is unsuited. Basically this is caused by the fact that at each learning step all network weights $\mathbf{w}_o$ are modified. This has the effect that the SLP is adapted to the current training ensemble, but already learned knowledge is quickly fading out. Therefore SLP networks suffer from the well-known “catastrophic forgetting effect” (French, 1999).

2.2 Multi Layer Perceptron

The Multi Layer Perceptron (MLP) is a cascade of SLP networks (Reed & Marks II, 1998) and therefore can be seen as the generalization of the simpler Single Layer Perceptron. In contrast to the SLP these networks are composed of an input, an output layer and an arbitrary number of hidden layers. Although the number of hidden layers can in principle be large, Lippmann (1987) showed that already networks with two hidden layers are sufficient to create arbitrary decision functions. Therefore MLP are considered to be universal function approximators (Reed & Marks II, 1998). The basic requirement for this multi-layered networks was the development of the error back-propagation algorithm. This



Network Topology

Learning Principle

Figure 2.3: **Multi Layer Perceptron.** The Multi Layer Perceptron (MLP) is composed of an input, an output layer and an arbitrary number of hidden layers. Similar to SLPs, the output of each neuron is calculated based on an activation and a transfer function. Furthermore it was shown that already two hidden layers are sufficient to learn any decision function. Therefore MLPs are considered as universal function approximator.

method was originally proposed by Werbos (1974) and was popularized based on the book of Rumelhart et al. (1986).

In general MLPs are multi-layered and fully connected networks as shown in Fig. 2.3. The output of each neuron is calculated, analogous to the SLP, based on scalar product activation and a transfer function Φ. The fundamental difference is the propagation of the network error back through the different hidden layers. The weight update of neuron m connected to neuron n in the previous layer is calculated in the following way:

$$\Delta w^i_{nm} = \Theta \; \delta^i_m \; p^i_n. \tag{2.3}$$

The p^i_n is the output activity of neuron n for the input pattern $\mathbf{x}^i$. Furthermore the local gradient δ^i_m is defined as:

$$\delta^i_m = \begin{cases} \Phi'(p^i_m)\;(t^i_o - p^i_o) & : \text{ for output neurons} \\ \Phi'(p^i_m)\;\sum_k \delta^i_k \; w_{mk} & : \text{ for hidden neurons} \end{cases}, \tag{2.4}$$

where k are all neurons that get input from node m. Additionally $(t^i_o - p^i_o)$ is the difference between the desired network output t^i_o and the network output p^i_o for class o.

Similar to the SLP in each training step all network weights are adapted. As a consequence also MLP networks can not maintain the long-term stability of learned knowledge if the training ensemble is limited and continuously changing. In contrast to the previous network model, MLP networks can be utilized for arbitrary learning tasks. Nevertheless the selected size of the hidden-layers considerably effects the generalization performance (Haykin, 1994). Therefore typically several networks are trained to find the optimal compromise, but this is not applicable for life-long learning tasks. Furthermore the back-propagation algorithm is known to converge very slowly (Fahlman & Lebiere, 1990). Therefore several modifications to overcome this limitation were proposed like the Momentum Term (Rumelhart et al., 1986), Quickprob (Fahlman, 1988) or the Resilient Propagation (Riedmiller & Braun, 1993). Nevertheless even with these modifications MLP networks can only be applied to offline learning tasks.

2.3 Cascade Correlation

Cascade Correlation (Fahlman & Lebiere, 1990) was originally developed to overcome limitations and drawbacks of the popular back-propagation learning algorithm (Rumelhart et al., 1986). The basic idea of the Cascade Correlation approach (see Fig. 2.4) is that single hidden units are incrementally added to the network until the given stopping criterion is reached. This allows a self-adaptation of the network to the difficulty of the learning problem, whereas commonly the network size have to be prespecified. Furthermore the training of the input connections to the hidden unit is separated from the output weights. This separation was proposed to avoid the moving target problem of MLP networks (Fahlman & Lebiere, 1990). This problem occurs because each neuron is trying to achieve a useful function in the overall network. Typically this is very difficult if all other neurons are also changed, so that often many neurons learn similar detectors and therefore worsen the convergence speed.

The learning method begins with a similar network architecture like the SLP networks, so that first the direct input-output connections are trained. For the training of these connections the same learning procedures as proposed for the MLP networks can be utilized (see Section 2.2).

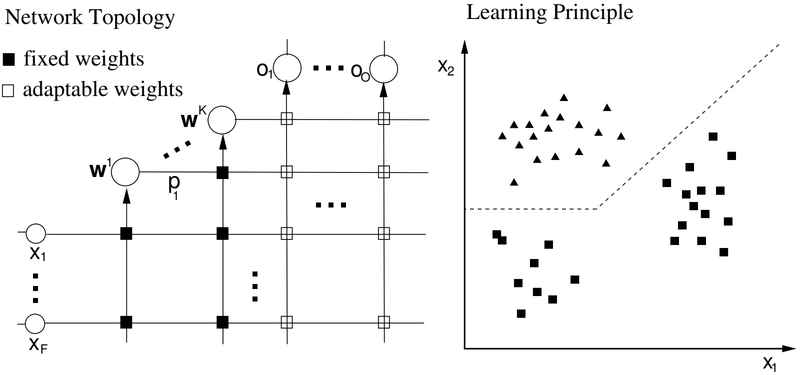


Figure 2.4: **Cascade Correlation.** The Cascade Correlation is an incremental learning method, where hidden neurons $\mathbf{w}^k$ are added until the predefined stopping criterion is reached. For each new hidden unit first the input connections are updated based on a gradient ascent method to maximize the magnitude of correlation between the candidate units output and the residual network output error. After the convergence of this gradient ascent the units input connections are frozen and all connections to the output layer are updated. The outcome of this iterative learning procedure is a narrow but deep neural network. Due to the incremental learning of hidden neurons Cascade Correlation can be applied to nonlinear learning problems. Furthermore the complexity of the learnable decision function is dependent on the total number of $\mathbf{w}^k$. For our simple two-class problem we consider that after the insertion of the first hidden unit the learning procedure converges. Therefore the decision function consists only of two linear functions.

If after several iterations no significant error reduction has occurred the remaining error is calculated over the entire training set. If the performance is already sufficiently well the learning is stopped. Otherwise a new hidden unit $\mathbf{w}^{K+1}$ is created with connections to all input and all pre-existing hidden units. The input connections of this new hidden unit are updated to maximize Π, the magnitude of the correlation between the units output p_{K+1} and the residual error E_o of each output node o and input pattern $\mathbf{x}^i = \{x_1, \ldots, x_F, p_1, \ldots, p_K\}$:

$$\Pi = \sum_o \left| \sum_i (p^i_{K+1} - p_{\bar{K}+1})\, (E^i_o - \bar{E}_o) \right| . \qquad (2.5)$$

Here $p_{\bar{K}+1}$ and $\bar{E}_o$ are the averages of p_{K+1} and E_o. In order to maximize Π a gradient ascent is used. The derivative of Π with respect to the m-th incoming weight w_m^{K+1} is defined as:

$$\frac{\partial \Pi}{\partial w_m^{K+1}} = \sum_o \sum_i \sigma_o \left(E_o^i - \bar{E}_o\right) \Phi'(\mathbf{x}^i)\, x_m^i, \tag{2.6}$$

where σ_o is the sign of the correlation between the candidate units output p_{K+1}^i and and the network output o. Additionally $\Phi'(\mathbf{x}^i)$ is the derivative for pattern $\mathbf{x}^i$ of the candidate units activations functions with respect to the sum of its inputs. Finally x_m^i is the input that the candidate unit receives from unit m. After the convergence of this gradient ascent the unit's input weights are frozen and all connections to the output layer are trained again. Afterwards it is considered if a further node is inserted or whether the learning procedure stops.

The self-adaptation capability of the network dimensionality makes this model interesting for life-long learning tasks, because an efficient adaptation to the difficulty of the learning problem can be achieved. Additionally also the fixation of the input connections of each hidden node guarantees the stability of the feature detectors. Compared to SLP and MLP this at least alleviate the "catastrophic forgetting effect" (Tetewsky et al., 1995; French, 1999). Nevertheless due to the continuous adaptation of the connection to the output layer long-term stability can not be guaranteed. Although compared to MLPs Cascade Correlation learns quickly (Fahlman & Lebiere, 1990; Reed & Marks II, 1998) it is still to slow too enable learning in direct interaction with a human tutor.

2.4 Vector Quantization Networks

Vector quantization methods like the Learning Vector Quantization (Kohonen, 1989), Self Organizing Maps (Kohonen, 1990) or Growing Neural Gas (Fritzke, 1995) are a group of neural network architectures that generate the decision function based on representative prototypes $\mathbf{w}^k$ with $k = 1, \ldots, K$, where K denotes the total number of nodes. The training of all these network types is based on a distance computation (e.g. Euclidean distance) between the representatives $\mathbf{w}^k$ and the current feature vector $\mathbf{x}^i$. Additionally also the learning rules are similar for this group of neural networks. The major difference between these learning methods is whether a topology between the different nodes is predefined, is

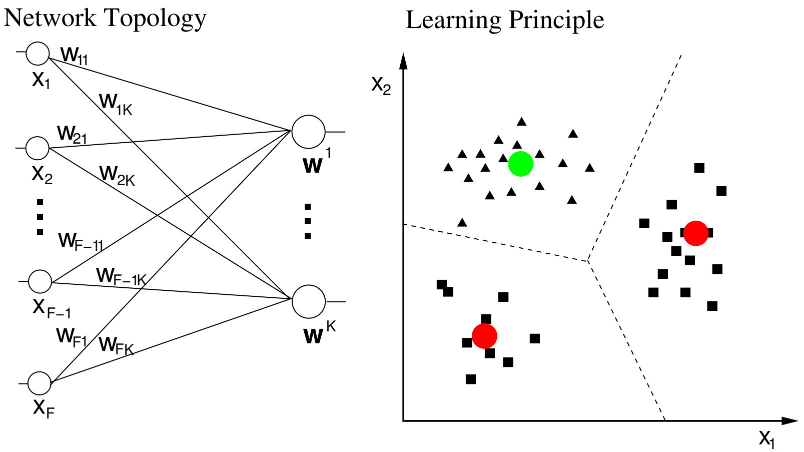


Figure 2.5: **Learning Vector Quantization.** The Learning Vector Quantization (LVQ) belongs to the group of vector quantization networks. The network consists of an input layer and a predefined number of prototype nodes $\mathbf{w}^k$, where each $\mathbf{w}^K$ is assigned to a class label o. The winning node $\mathbf{w}^{k_{\min}}$ for a given input $\mathbf{x}^i$ is calculated based on a distance measurement (e.g. Euclidean distance). Afterwards the $\mathbf{w}^{k_{\min}}$ is modified based on the correctness of the network output. This means if the class label of the winning node matches with the label of the training vector then $\mathbf{w}^{k_{\min}}$ is shifted into the direction of $\mathbf{x}^i$ and otherwise in the opposite direction. The prototype nodes subdivide the input space into smaller subregions, where for each region exactly one $\mathbf{w}^k$ becomes the winning node. Therefore LVQ networks can be applied to arbitrary nonlinear learning problems.

acquired during the learning or is not considered at all. Furthermore these network architectures are different with respect to the capability of dealing with unsupervised or supervised learning problems, while Growing Neural Gas (GNG) also enables incremental learning. In the following we use the Learning Vector Quantization (LVQ) method as an example to illustrate the working principles of this group of neural networks (see Fig. 2.5).

LVQ networks are typically trained in a supervised manner based on a predefined number of representatives $\mathbf{w}^k$ that are adapted according to a stream of randomly selected feature vectors $\mathbf{x}^i$. The actual training method is based on a distance computation between these $\mathbf{x}^i$ and all representatives $\mathbf{w}^k$, where commonly the Euclidean distance is used for this network type:

$$d(\mathbf{x}^i, \mathbf{w}^k) = ||\mathbf{x}^i - \mathbf{w}^k|| = \sum_f (x_f^i - w_f^k)^2. \tag{2.7}$$

As the next step the so-called winning node $\mathbf{w}^{k_{\min}}$ with the smallest distance to the current $\mathbf{x}^i$ is determined:

$$k_{\min} = \arg\min_k \; d(\mathbf{x}^i, \mathbf{w}^k) \quad \forall k \; . \tag{2.8}$$

Only this winning node $\mathbf{w}^{k_{\min}}$ is adapted based on the following learning rule, while all other nodes remain unchanged:

$$\mathbf{w}^{k_{\min}} := \mathbf{w}^{k_{\min}} + \mu \, \Theta(\mathbf{x}^i - \mathbf{w}^{k_{\min}}), \tag{2.9}$$

where $\mu = 1$ if the class label of the feature vector $\mathbf{x}^i$ and the class label of the winning node $\mathbf{w}^{k_{\max}}$ matches, otherwise $\mu = -1$ and the winning node will be shifted into the opposite direction of $\mathbf{x}^i$. Finally Θ is the learning rate that controls the shift to and away from the current input vector. Furthermore several extensions of this basis algorithm were proposed with respect to the convergence speed like LVQ3 (Kohonen, 1990) or Optimized LVQ (Kohonen, 1992). Also with respect to the relevance weighting of the feature dimensions different modifications are proposed like the Relevance LVQ (Bojer et al., 2001), the Generalized Relevance LVQ (Hammer & Villmann, 2002) or the Generalized Matrix LVQ (Schneider et al., 2007).

These learning methods are better suited for life-long learning, due to the fact that commonly only a single or a small group of prototype nodes is adapted during one learning step. Nevertheless especially the incremental learning Growing Neural Gas can not maintain the stability of the learned knowledge if the training set is continuously changing (Hamker, 2001; Furao et al., 2007). The learning speed of LVQ networks is considered to be faster compared to the back-propagation approach (Hawickhorst et al., 1995), but strongly depends on the overall network size and the initialization of the prototype nodes. Therefore incremental node insertion as proposed for the GNG is beneficial to find a good compromise between the overall network size and the accuracy of representation.

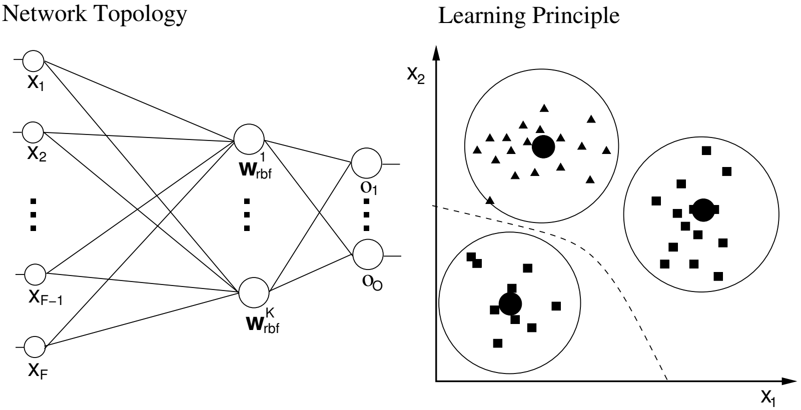


Figure 2.6: **Radial Basis Function Networks.** The Radial Basis Function network model (RBF) consists of an input, a hidden and an output layer. It combines prototype based representations similar to vector quantization methods with gradient descent related to the Single Layer Perceptron. The combination of the RBF hidden layer with the connections to the output layer make RBF in general applicable for nonlinear problems.

2.5 Radial Basis Function Networks

Radial Basis Functions (RBF) were first introduced to solve multivariate interpolation problems, where the early work on this topic was reviewed by Powell (1985). The first attempt to utilized these functions for the design of neural networks was done by Broomhead & Lowe (1988). Further major contributions with respect to the early development of RBF networks include papers by Moody & Darken (1989) and Poggio & Girosi (1989). Compared to the previously described neural network architectures RBF networks can be seen as a combination of prototype-based representations and gradient descent learning. In general RBF networks are composed of an input, a hidden and an output layer as illustrated in Fig. 2.6. The hidden layer of this network model is composed of so-called RBF-centers with Gaussian receptive fields. Similar to the MLP, RBF networks are known as universal function approximators and therefore can be applied to any learning problem (Park & Sandberg, 1991).

The activation of a single RBF-center is similar to the LVQ networks based on the Euclidean distance between the input vector $\mathbf{x}^i$ and all

RBF-nodes $\mathbf{w}^k_{rbf}$:

$$d(\mathbf{x}^i, \mathbf{w}^k_{rbf}) = ||\mathbf{x}^i - \mathbf{w}^k_{rbf}||. \tag{2.10}$$

In contrast to the LVQ, the output activity p^k_{rbf} of the RBF-center k is then computed based on a Gauss function:

$$p^k_{rbf} = \exp\left(\frac{\left(d(\mathbf{x}^i, \mathbf{w}^k_{rbf})\right)^2}{2(\sigma^k)^2}\right). \tag{2.11}$$

The variance $(\sigma^k)^2$ controls the response range of the hidden unit $\mathbf{w}^k_{rbf}$ that together with the determination of the prototype weights $\mathbf{w}^k_{rbf}$ (e.g. based on k-means clustering) are the most important parameters for the generalization capability of these kind of neural networks. Finally the network output p^i_o for each class o is calculated similar to the SLP in the following way:

$$p^i_o = \sum_k w^k_o \, p^k_{RBF}, \tag{2.12}$$

where the corresponding weights $\mathbf{w}_o$ are trained based on a gradient-descent learning method. The corresponding error function E is similar to the back propagation algorithm defined as:

$$E = \frac{1}{2} \sum_o \sum_i (t^i_o - p^i_o)^2, \tag{2.13}$$

where t^i_o is the target value for neuron o and input pattern $\mathbf{x}^i$. Furthermore p^i_o is the corresponding network output.

Due to the gradient based learning between the RBF and output layer this model has the same drawbacks with respect to life-long learning as mentioned for the SLP and MLP networks. In contrast to back-propagation RBF networks require shorter training time, but require a higher storage capacity (Hawickhorst et al., 1995). Furthermore RBF networks can be extended to allow fast and incremental learning as proposed by Fritzke (1994a). This allows an automatic adaptation to the complexity of the recognition task, so that no a priori knowledge is required to select an appropriate network dimensionality.

2.6 Fuzzy ARTMAP

The Fuzzy ARTMAP (Carpenter et al., 1992) architecture belongs to the ART network family (Grossberg, 1976; Carpenter & Grossberg, 1988;

Carpenter et al., 1991) that allows supervised learning. This architectures is composed of two different ART-networks, ART^a and ART^b interconnected by a so-called Map field as illustrated in Fig. 2.7 after (Zell, 1994). The ART^a network is the part of the architecture that is responsible for the clustering of the training vectors $\mathbf{x}^i$. The ART^b network is utilized for the clustering of the different classes, using the teach vectors $t^i = (t^i_i, \ldots, t^i_O)$ as input. The Map field refers each ART^a output node to a ART^b node, so that each ART^a node is assigned to a class in ART^b.

Each of the two ART networks is composed of a comparison and a recognition layer. The activation in the recognition layer of each ART network is defined for each node k in the following way:

$$p^k(\mathbf{x}^i) = \frac{|\mathbf{x}^i \wedge \mathbf{w}^k|}{\alpha + |\mathbf{w}^k|}, \tag{2.14}$$

where $\alpha > 0$ is the so-called choice parameter. Furthermore $\wedge$ is the fuzzy AND operator that is defined as:

$$(x^i \wedge w^k)_f = \min(x^i_f, w^k_f). \tag{2.15}$$

Based on these calculated node responses $p^k(\mathbf{x}^i)$ the winning neuron $\mathbf{w}^{k_{\max}}$ is calculated as follows:

$$k_{\max} = \max_k(p^k(\mathbf{x}^i)). \tag{2.16}$$

In the next step, based on the match function, it is checked whether the winning $\mathbf{w}^{k_{\max}}$ neuron matches the similarity criterion that is defined by the vigilance parameter ρ:

$$\frac{|\mathbf{x}^i \wedge \mathbf{w}^{k_{\max}}|}{|\mathbf{w}^{k_{\max}}|} \geq \rho. \tag{2.17}$$

If this criterion is not fulfilled node $\mathbf{w}^{k_{\max}}$ is reseted and the next higher activated neuron is tested. Otherwise the ART network reaches the resonance state.

Based on resonance state of the selected ART^a and ART^b winning nodes the correctness of the recognition result is checked. This is the case when both network parts activate the same neuron in the so-called map field. If this is the case the Fuzzy ARTMAP network reaches the final network

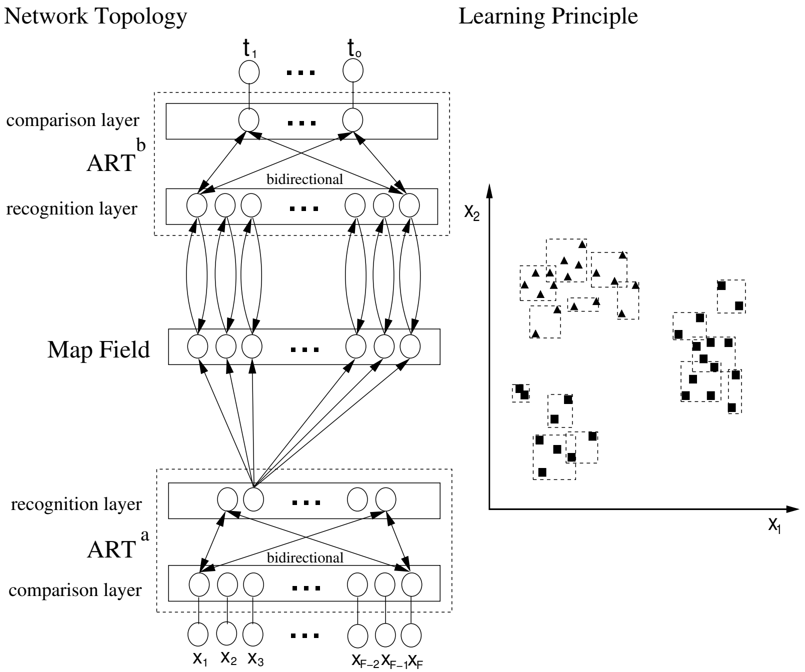


Figure 2.7: **Fuzzy ARTMAP.** The Fuzzy ARTMAP belongs the ART network family. It enables supervised learning based on the combination of two ART networks ART^a and ART^b that are inter-connected based on a so-called Map field. For this network architecture the ART^a is typically utilized to cluster the input vectors $\mathbf{x}^i$, where the ART^b clusters the label vectors $\mathbf{t}^i$. For Fuzzy ARTMAP complement coding is typically used to prevent the undesired property that too many weights converge to zero. If complement coding is used the learned representation can be geometrically interpreted as hypercubes, as illustrated on the right side of this figure.

state for the current $\mathbf{x}^i$ and in both network parts the weights of the winning neurons $\mathbf{w}^{k_{\max}}$ are adapted in the following way:

$$\mathbf{w}^{k_{\max}} := \mathbf{x}^i \wedge \mathbf{w}^{k_{\max}}. \tag{2.18}$$

Otherwise an inter-art-reset is triggered, where the winning nodes in both network parts are reseted and additionally the vigilance parameter ρ is increased. The increase is done in a way that the actual winning nodes can not become again the winning node for the input vector $\mathbf{x}^i$.

Fuzzy ARTMAP was designed to approach the "stability-plasticity dilemma". Therefore this method is a candidate for solving life-long learning tasks, but Fuzzy ARTMAP is known to be very sensitive to the selection of the vigilance parameter, to the noise level and the presentation order of the training data (Polikar et al., 2001). Additionally Fuzzy ARTMAP tends to have problems with complex decision boundaries, which results in worse performance compared to vector quantization methods like the Growing Neural Gas (Heinke & Hamker, 1998). The major advantage of this learning approach is the fast one-shot learning mode that is beneficial for interactive learning tasks.

2.7 Discussion

Based on this review of standard artificial neural networks, we believe that local learning methods like the Fuzzy ARTMAP or the vector quantization networks are better suited for life-long learning tasks compared to global learning methods, because only small portions of the network are adapted. In contrast to this for global learning methods like the SLP or MLP all network weights are updated, so that the stability of learned knowledge is more difficult to achieve. Furthermore Cascade Correlation and RBF networks are a compromise between local and global learning, because the learning of the input connections of hidden neurons are separated from the learning of the output connections.

For our targeted life-long learning tasks the different classes are incrementally presented and after their storage in the network model do not reappear again. Therefore the learning method should be able to find nearly optimal solutions for many tasks based on arbitrary starting conditions and a broad range of network parameters. For all local learning approaches this requirement can be better achieved with prototype-based vector quantization networks rather than Fuzzy ARTMAP. Out of the group of prototype-based methods the Learning Vector Quantization (LVQ) network model was selected as basis for our proposed life-long learning methods, because of its simplicity compared to the Growing Neural Gas (GNG) and its applicability for supervised learning problems.

With respect to interactive learning a one shot-learning method like the Fuzzy ARTMAP is beneficial. Therefore we propose to store the knowledge into an intermediate and fast learning short-term memory (STM) representation. Based on this STM representation a memory consolidation into a long-term memory is proposed using extended LVQ networks. This combination relaxes the constraints on the learning speed for the memory consolidation, especially if the STM storage capacity is large enough. Nevertheless if the overall network size and the effectively used feature dimensionality (e.g. sparsity of the feature representation) is small enough LVQ networks can be utilized for fast interactive learning.

Chapter 3

Life-Long Learning for Identification Tasks

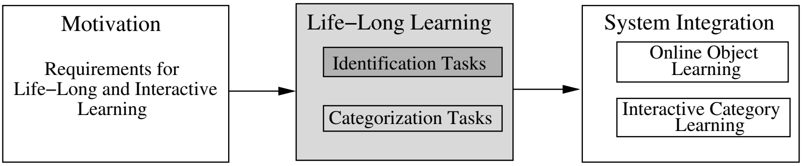


Figure 3.1: **Life-Long Learning for Identification Problems.** In the following chapter we concentrate on interactive and life-long learning for identification tasks. The proposed memory model is composed of an intermediate short-term memory that is build up based on a one-shot learning method. Additionally a memory consolidation into a more condensed long-term memory is proposed that is one of the major contributions of the presented dissertation.

In the following chapter an object identification architecture is proposed that allows life-long learning of complex-shaped objects. The development of this memory model started with the proposal of a short-term memory (STM) representation (Kirstein, 2004; Kirstein et al., 2005b). This learning method is based on a feature extraction hierarchy (Wersing & Körner, 2003) that was motivated by the ventral pathway of the human visual system. This combination enables fast interactive learning of many complex-shaped objects as was shown in (Kirstein et al., 2005b; Kirstein et al., 2008). Although this method achieves good generalization performance the representational costs for storing many high-dimensional feature vectors is the major drawback. Therefore we assume a limited STM and proposed a memory consolidation into a long-term memory (LTM) representation (Kirstein et al., 2005a; Kirstein et al., 2008). The target of this combined memory model is to obtain a flexible representation that is capable of high-performance appearance-based object identification. In several experiments we could show that the desired targets could be reached. The presented results in this chapter are basically based on (Kirstein et al., 2008).

3.1 Related Work

Most research on trainable and model-free object identification algorithms has so far focused on learning based on large data sets of images recorded beforehand and then performing offline training of the corresponding classifiers. In these approaches learning speed is not a primary optimization goal, so that the offline training times typically last many hours. This is usually caused by the natural high dimensionality of visual sensorial input, which poses a challenge to most current learning methods. Another problem is that most powerful classifier architectures such as the Multi Layer Perceptrons (MLP) or Support Vector Machines (SVM) do not allow online training with the same performance as for offline batch training.

3.1.1 Online Learning and Man-machine Interaction

To cope with the dimensionality problem one approach is to reduce the complexity of the sensorial input to simple blob-like stimuli (Jebara & Pentland, 1999), for which only positions are tracked. Based on the positions, interactive and online learning of behavior patterns in response to these blob stimuli can be performed. A slightly more complex representation was used by Garcia et al. (2000), who have applied the coupling of an attention system using features like color, motion, and disparity with a fast learning of visual structure for simple colored geometrical shapes like balls, pyramids, and cubes. They represent shape as low-resolution feature maps computed based on convolutions with Gaussian partial derivatives. Using shape and color map representations the system can learn to direct attention to particular objects.

Histogram-based methods are another common approach to tackle the problem of high dimensionality of visual object representations. Steels & Kaplan (2001) have studied the dynamics of learning shared object concepts based on color histograms in an interaction scenario with a dog robot. The object representation allows online learning using the limited computational resources of the pet robot, but lacks a stronger concept of shape discrimination. A model of word acquisition, that is based on multidimensional receptive field histograms (Schiele & Crowley, 2000) for shape and color representation was proposed by Roy & Pentland (2002). The learning proceeds online by using a short-term memory for identifying reoccurring pairs of acoustic and visual sensory data, that are then passed to a long-term representation of extracted audio-visual objects.

Arsenio (2004) has investigated a developmental learning approach for humanoid robots based on an interactive object segmentation model that can use both external movements of objects by a human and internally generated movements of objects by a robot manipulator. Using a combination of tracking and segmentation algorithms the system is capable of learning objects online by storing them using a geometric hashing (Rigoutsos & Wolfson, 1997) representation. Based on a similarity threshold, objects are separated into different classes using color and pairwise edge histograms. The discriminatory power, however, seems to be limited to a small number of objects and still strongly depends on color. What is more important is the integration of the online object learning into a model for tracking objects and learning task sequences and to recognize objects employed on such tasks from human-robot interactions.

An interesting approach to supervised online learning for object recognition was proposed by Bekel et al. (2004). Their classification architecture consists of three major stages. The two feature extraction stages are based on vector quantization and a local Principal Component Analysis (PCA) measurement. The final stage is a supervised classifier using a Local Linear Map architecture. The image acquisition of new object views is triggered by pointing gestures on a table, and is followed by a short training phase, which takes some minutes. The main drawback is the lack of an incremental learning mechanism to avoid the complete retraining of the architecture.

Online learning has also been investigated for robotics in domains of behavior and movement control. In this field the dimensionality of the representation space can be still quite large for robotic systems with many degrees of freedom although it does not reach the full complexity of visual input. As an important example that particularly focuses on incremental online learning we would like to mention Vijayakuma et al. (2005), who propose a locally weighted projection regression (LWPR) algorithm, which is especially used for learning robot movements. The advantage of this method is the possibility to train complex robot movements online with only a few trials. The basic idea of the LWPR algorithm is to reduce the high number of possible input dimensions (up to 90 joints) to the essential ones necessary for the particular movement. The proposed method works well, if such a low dimensional distribution in the input space exists.

3.1.2 Network Architectures for Incremental and Life-Long Learning

One established neuronal network architecture that is able to learn online with the same performance as for offline training is the Adaptive Resonance Theory (ART) and especially Fuzzy ARTMAP (Carpenter et al., 1992). The relation of this network architecture to our short-term memory model will be discussed later (see Section 3.2.1) in more detail. In recent years the ART network family was applied to several problems including recognition of handwritten digits (Carpenter et al., 1992) and a sensorimotor anticipation architecture for robot navigation (Heinze et al., 2001). An overview of several other ART-based applications can be found in (Carpenter & Grossberg, 1998).

Incremental Radial Basis Function (RBF) networks (Fritzke, 1994a) and the Growing Neuronal Gas (GNG) model (Fritzke, 1995) were suggested with a focus on incremental learning. Although it is possible to train these networks with a slowly changing training set, these architectures are mainly designed for offline training. Typically these networks cannot be trained on a limited training set without significantly losing generalization performance, because of a permanent increase in the number of neurons and the drift of nodes to capture the current training data (Hamker, 2001).

Furao & Hasegawa (2006) propose several improvements to the unsupervised version of the GNG and especially target the life-long learning of non-stationary data for problems like clustering of faces or topology learning of images. They use a two-layered network, where the first layer is used to generate a topology structure of the input data and the second layer is used to determine the number of clusters. Furthermore they propose several utility estimation measurements for evaluating the insertion of nodes or to decide which nodes can be removed. Additionally they use an individual learning rate for each node, which strongly improves the life-long learning capability. A related approach was proposed by Hamker (2001), who introduced a neuronal network architecture for supervised learning, called life-long learning cell structures (LLCS). The LLCS networks are based on the Growing Cell Structures (Fritzke, 1994b) and provide several extensions, like the calculation of an individual node learning rate, the definition of an insertion rule and the use of several measurements to detect useless nodes. The LLCS networks are also able to detect regions in low dimensional data where points of different classes overlap. This avoids an unlimited insertion of neurons in those areas.

Other approaches to the “stability-plasticity dilemma” where proposed by Polikar et al. (2001) and Ozawa et al. (2005). Polikar et al. (2001) proposed the “Learn++” approach that is based on the boosting (Schapire, 1990) technique. This method combines several weak classifiers to a so-called strong classifier based on a majority-voting schema, where the weak classifiers are incrementally added to the network and afterwards are kept fixed. The proposed “Learn++” can therefore be used for life-long learning tasks, but for more complex tasks a large amount of such weak classifiers is required to represent the different classes. In contrast to this Ozawa et al. (2005) proposed to store representative input-output pairs into a long-term memory for stabilizing an incremental learning Radial Basis Function (RBF) like network.

3.2 Life-Long Learning Model for Object Identification

In the following we consider life-long learning at different time scales to represent arbitrary objects. The corresponding memory model (see Fig. 3.2) is motivated by the functional differentiation in the two STM and LTM systems of human brains. Our target is to perform super-

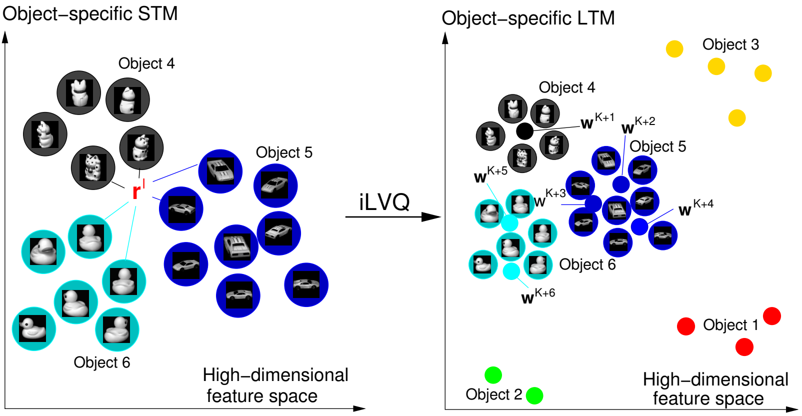


Figure 3.2: **Illustration of the Life-Long Learning Model for Object Identification Tasks.** Based on a stream of input views and a similarity calculation representatives are stored into a limited short-term memory (STM). The overall number of representatives is typically only a subset of all presented input views and depends on the complexity of the object itself. In the next step these representatives are transferred into a long-term memory (LTM) based on a life-long learning vector quantization approach. This LTM model learns much slower compared to the STM but considerably reduces the amount of required network resources resulting in a compact prototype-based representation $\mathbf{w}^k$.

vised and online learning of object views using the STM, which has the ability to incrementally build up an object representation without destroying already learned knowledge. This STM provides fast learning, but also has a limited capacity. For the buildup of the LTM we propose an incremental Learning Vector Quantization (iLVQ) method. This approach realizes the transfer from the fast learning STM into the slower learning LTM, which results in a more integrated and condensed object representation. Furthermore we define for our LTM model different extensions to Learning Vector Quantization (LVQ) networks (Kohonen, 1989) that are necessary for our target of an incrementally and life-long learning system. We demonstrate the technical realization of the proposed approach in an interactively trainable online learning system that can robustly recognize several objects.

3.2.1 Online Vector Quantization to Build a Short-Term Memory

The online vector quantization (oVQ) model provides fast appearance-based learning of three-dimensional objects, which can immediately be recognized. Therefore oVQ enables interactive learning, so that continuously new objects can be learned, while already represented objects can be refined. The proposed model stores template-based representatives $\mathbf{r}^l$ in a so-called short-term memory. The number of representatives $\mathbf{r}^l$ for a specific object is related to the complexity of the object and is not specified beforehand. The learning process is based on the similarity to already stored representatives $\mathbf{r}^l$ of the same object. Therefore this online vector quantization model reduces the number of representatives $\mathbf{r}^l$ in contrast to a naive approach where every feature vector $\mathbf{x}^i = (x_1^i, \ldots, x_F^1)$ is stored in memory. Especially already seen views or very similar views are not collected into the short-term memory.

The labeled feature vectors $\mathbf{x}^i$ are stored in a set of L representatives $\mathbf{r}^l$, $l = 1, \ldots, L$, that are incrementally collected, and labeled with class o with $o = 1, \ldots, O$ assigned to $\mathbf{x}^i$. The acquisition of templates is based on a similarity threshold ϵ_{stm}. New views of an object are only collected into the short-term memory (STM) representation if their similarity to the previously stored views is less than ϵ_{stm}. The parameter ϵ_{stm} is critical, characterizing the compromise between representation resolution and computation time needed for one training or validation step. We denote the similarity between feature vector $\mathbf{x}^i$ and representative $\mathbf{r}^l$ by A^{il}:

$$A^{il} = \exp\left(-\frac{||\mathbf{x}^i - \mathbf{r}^l||}{\sigma}\right). \tag{3.1}$$

Here, σ is chosen for convenience such that the average similarity in a generic recognition setup is approximately equal to 0.5. We use the exponential function just to obtain an intuitive notion of similarity, any other monotonous transformation of the Euclidean distance would also be possible.

We define R_o as the set of representatives $\mathbf{r}^l$ that belong to object o. For one learning step the similarity A^{il} between the current training vector $\mathbf{x}^i$, labeled as object o and all representatives $\mathbf{r}^l \in R_o$ of the same object

o is calculated and the maximum value is computed as:

$$A^i_{\max} = \max_{l \in R_o} A^{il}. \tag{3.2}$$

The training vector $\mathbf{x}^i$ with its class label o is added to the object representation, if $A^i_{max} < \epsilon_{stm}$. If L representatives were collected before, then $\mathbf{x}^i$ is added as representative $L + 1$ with $\mathbf{r}^{L+1} = \mathbf{x}^i$ and is labeled with class o attached to $\mathbf{x}^i$. Otherwise we assume that the vector $\mathbf{x}^i$ is already sufficiently well represented by one $\mathbf{r}^l$, and do not add it to the representation. We call this basic template-based representation online vector quantization (oVQ). Due to the non-destructive incremental learning process, online learning and recognition can be done at the same time, without a separation into training and testing phases. To model a limited STM capacity, in the simulations an upper limit can be set on the number of objects that can be represented. This means that, when too many objects are presented, representatives belonging to the oldest learned object are removed from the STM.

For the online recognition of a new and unclassified feature vector $\mathbf{x}^i$ a nearest neighbor search on the set of all representatives stored in the short-term memory is performed. The nearest neighbor search selects the best matching node $\mathbf{r}^{l_{\max}}$, where $l_{\max}$ satisfies:

$$l_{\max} = \arg\max_l (A^{il}). \tag{3.3}$$

The class label $o^{l_{\max}}$ of the winning representative $\mathbf{r}^{l_{\max}}$ is then assigned to the current unclassified test view $\mathbf{x}^i$.

The oVQ algorithm can handle the used high-dimensional feature representation proposed by Wersing & Körner (2003) in an efficient way. It is especially suited for the sparsity of this feature representation, which allows us to store ten thousands of representatives, while keeping the ability to train and validate new occurring feature vectors online. The similarity threshold ϵ_{stm}, the only critical parameter in our STM model, controls the tradeoff between a more detailed and exhaustive object view sampling and the amount of representatives in the STM.

Based on the description of our oVQ algorithm the relation to Fuzzy ARTMAP (Carpenter et al., 1992) and Fuzzy ART (Carpenter et al., 1991) will be discussed in the following. Both architectures have the common feature that they can immediately recognize a specific object view after a single occurrence (“one shot learning”), which makes them suitable for online learning. It is also possible to incrementally add new objects without destroying already learned capabilities and the learning process in both algorithms is based on a similarity condition called vigilance ρ for ART networks.

Besides the drawbacks summarized in Section 2.6 the more complex Fuzzy ARTMAP architecture is not suitable for our object identification task. This is related to the sparsity of the feature vectors, which essentially requires complement coding to avoid that too many adaptive weights become zero. A large amount of zero weights is an unattractive condition for Fuzzy ART networks that should be prevented (Carpenter et al., 1992), because in such a case the "choice function" used for calculating the winner node always results in nearly perfect matches, which results in choosing a winner node independent of the input. Additionally the already very high-dimensional feature vectors are doubled in size by this coding schema. Based on the complement coding and the vigilance parameter ρ, input vectors are assigned to hypercubes around the representative vectors with the size inversely proportional to ρ. This vigilance parameter ρ is, similar to the ϵ_{stm} in our model, a critical parameter. The ρ parameter should therefore similar to ϵ_{stm} chosen as small as possible, to avoid the allocation of an enormous amount of resources. On the contrary, small vigilance parameters ($\rho < 0.9$) cause other problems, because it allows the creation of large hypercubes during the learning process. This leads to the undesired convergence of many adaptive weights to zero as a consequence of strong intra-class variations of the sparse feature vectors $\mathbf{x}^i$. These intra-class variations are caused by appearance changes of objects during rotation. This together with relatively closely located vectors of related objects in similar poses will most probably result in many partially overlapping hypercubes. If such hypercubes are belonging to different classes and validation vectors are located in these areas, then the generalization ability of the network will be reduced. This is because the "choice function" results in the same optimal value for all nodes involved in this overlap and the selection of the winner is dependent on the search order.

3.2.2 Incremental LVQ to Build a Long-Term Memory

The STM model provides fast learning and achieves good recognition performance, as we will demonstrate in the results section. Nevertheless the large amount of memory for storing the high-dimensional feature vectors of all objects is the main disadvantage and is also biologically not plausible. Therefore we propose a transfer from the STM into the LTM, inspired by the transfer from medial temporal lobe into the Neocortex in biological vision. To build up such an LTM model we use an incremental LVQ algorithm (iLVQ). This network architecture described in the following section should strongly reduce the representational effort of objects without reducing the generalization performance of the identification system. Additionally the LTM model is approaching the life-long learning problem, which allows learning of objects during the complete history of the iLVQ network.

The labeled STM representatives $\mathbf{r}^l$ in the high-dimensional feature space provide the input ensemble for our proposed long-term memory (LTM) representation, which is optimized and built up incrementally. The main reason for training the long-term memory based on the collected STM representatives $\mathbf{r}^l$ is that the STM already rejects very similar object views and reduces the number of training views for the long-term memory. This reduction causes a reduced training time in contrast to the case where every input view is used. Additionally we assume a limited STM capacity with only the most recently shown objects being represented. Therefore an algorithm is needed that is able to incrementally add new objects or even refine object representations without destroying already learned object knowledge, thereby taking into account the “stability-plasticity dilemma”.

The Learning Vector Quantization (LVQ) networks proposed by Kohonen (1989) are a well known neuronal network architecture for supervised learning. The single-layered LVQ networks are typically trained with a fixed number of nodes; therefore the number of nodes for each class must be selected before the training phase starts. It is quite difficult to accurately determine the necessary number of nodes for a particular class. If the number of nodes is too large convergence is slow, whereas a too low number only provides a poor generalization performance of the network. Additionally the number of necessary nodes is related to the complexity of a particular class itself. To take care of this fact a lot of a priori knowledge must be available to select an appropriate number of LVQ nodes. To avoid this problem we use an incremental approach for the LTM model, which is able to automatically determine the necessary number of nodes, based on the complexity of the object and the difficulty of the learning task. We also extend the basic LVQ networks with respect to the "stability-plasticity dilemma" of life-long learning tasks. All extensions of the basic LVQ network architecture will be described in the following.

For the training of our incremental LVQ (iLVQ) network, a stream of randomly selected input STM training vectors $\mathbf{r}^l$ is presented and classified using the labeled iLVQ representatives in a Euclidean metric. The training classification errors are collected, and each time a given sufficient number of classification errors has occurred, a set of new iLVQ nodes will be inserted. The addition rule is designed to promote insertion of nodes at the class boundaries. During training, iLVQ nodes are adapted with standard LVQ weight learning that moves nodes into the direction of the correct class and away from wrong classes. An important change to the standard LVQ method is an adaptive modification of the individual node learning rates to deal with the "stability-plasticity dilemma" of incremental learning. The learning rate of winning nodes is more and more reduced to avoid too strong interference of newly learned representatives $\mathbf{r}^l$ with older parts of the object long-term memory.

We denote the set of iLVQ representative vectors $\mathbf{w}^k$ with $k = 1, \ldots, K$, where K is the current number of nodes. The training of the iLVQ nodes is based on the current set of labeled STM nodes $\mathbf{r}^l$ that serve as input vectors for the LTM. Each iLVQ node $\mathbf{w}^k$ obtains an individual learning rate:

$$\Theta^k = \Theta_0 \exp\left(-\frac{a^k}{d}\right), \tag{3.4}$$

where Θ_0 is an initial value, d is a fixed scaling factor, and a^k is an iteration-dependent age factor. Furthermore the age factor a^k is incremented every time the corresponding $\mathbf{w}^k$ becomes the winning node.

New iLVQ nodes are inserted, if a given number $G_{\max}$ of training vectors are misclassified during the iterative presentation of the $\mathbf{r}^l$. We choose a value of $G_{\max} = 30$, since a high $G_{\max}$ value guarantees an optimal representation of objects with a minimal number of LVQ nodes, but also slows down the convergence speed of this learning algorithm. Within this error history, misclassifications are memorized with input $\mathbf{r}^l$ and the corresponding winning iLVQ node $\mathbf{w}^{k_{\min}}(\mathbf{r}^l)$. We denote S_o as the set of previously misclassified $\mathbf{r}^l$ within this error history that belong to class o. For each nonempty S_o a new node $\mathbf{w}^m$ is added to the representation, independent of the number of entries in S_o. This insertion technique limits the insertion of nodes, if many views of a particular class are wrongly classified. The iLVQ insertion rule is illustrated in Fig. 3.3. New nodes are initialized to the element of $\mathbf{r}^l \in S_o$ with minimal distance to its corresponding but wrong winning iLVQ node $\mathbf{w}^{k_{\min}}(\mathbf{r}^l)$ and is labeled with class o. This rule adds new nodes primarily near to class borders, where typically most classification errors occur. This node insertion rule can be related to boundary classifiers like Support Vector Machines (see (Burges, 1998) for an introduction to SVM), where so-called support vectors at the classification border are selected to form the decision boundary. In contrast to this the iLVQ algorithm forms Voronoi clusters, where the cluster centers can be quite far apart from the classification border.

A test view $\mathbf{x}^i$ is classified by determining the winning iLVQ node $\mathbf{w}^{k_{\min}}$ with smallest distance to the current feature vector $\mathbf{x}^i$ and is assigned to the corresponding label o attached to $\mathbf{w}^{k_{\min}}$ as the output class.

The formal definition of the iLVQ learning algorithm will be described in the following:

1 Choose randomly a representative $\mathbf{r}^l$ from the set of current STM nodes. Calculate the Euclidean distance between the $\mathbf{r}^l$ and all iLVQ nodes $\mathbf{w}^k$ and select the winning node with minimal distance to the $\mathbf{r}^l$:

$$k_{\min} = \arg\min_k(||\mathbf{r}^l - \mathbf{w}^k||). \tag{3.5}$$

After this selection process the winning node $\mathbf{w}^{k_{\min}}$ is adapted using the common LVQ learning rule:

$$\mathbf{w}^{k_{\min}} := \mathbf{w}^{k_{\min}} + \mu\Theta^{k_{\min}}(\mathbf{r}^l - \mathbf{w}^{k_{\min}}), \tag{3.6}$$

where $\mu = 1$ if the class label of the representative $\mathbf{r}^l$ and the class label of the winning node $\mathbf{w}^{k_{\min}}$ are identical, otherwise $\mu = -1$ and the winning node will be shifted into the opposite direction as the input representative $\mathbf{r}^l$. The learning rate $\Theta^{k_{\min}}$ for the winning node $\mathbf{w}^{k_{\min}}$ at time step t is calculated according to Eq. 3.4.

2 After the adaptation of the winning node $\mathbf{w}^{k_{\min}}$ the age factor $a^{k_{\min}}$ of this node will be incremented:

$$a^{k_{\min}} := a^{k_{\min}} + 1. \tag{3.7}$$

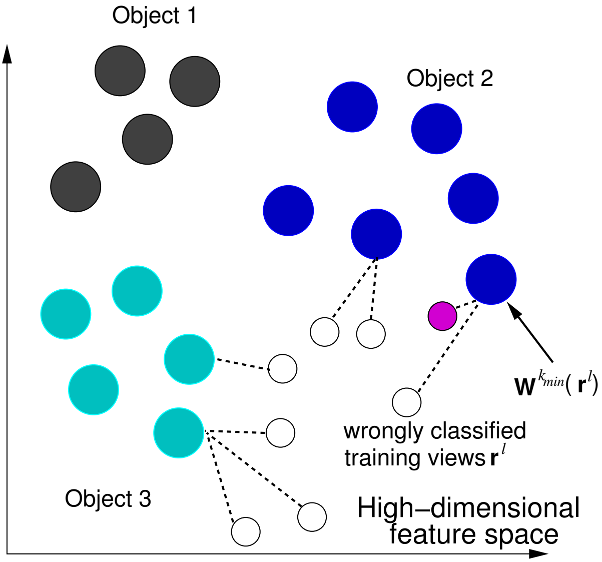

Object 1
Object 2
$\mathbf{w}^{k_{min}}(\mathbf{r}^l)$
wrongly classified training views $\mathbf{r}^l$
Object 3
High–dimensional feature space

Figure 3.3: **Illustration of the iLVQ Node Insertion Rule**. Wrongly classified training views $\mathbf{r}^l$ of class o are collected into S_o, which contains all wrongly classified views of the given class o. These views $\mathbf{r}^l \in S_o$ are shown with small circles, whereas the iLVQ nodes are shown as large filled circles. Additionally the distance of the $\mathbf{r}^l$ to their corresponding but wrong winning iLVQ node is shown (dashed lines). The insertion rule determines the wrongly classified $\mathbf{r}^l$ with minimal distance to the iLVQ node $\mathbf{w}^{k_{\min}}(\mathbf{r}^l)$. This training view (the small filled circle) is then used for initializing a new iLVQ node with class label o assigned to $\mathbf{r}^l$.

This increment of $a^{k_{\min}}$ results in a slightly smaller learning rate if the $\mathbf{w}^{k_{\min}}$ iLVQ node becomes in a further training step again the winning node.

3 If the current representative $\mathbf{r}^l$ was misclassified, then the number G of misclassified training vectors will be increased ($G := G + 1$) and $\mathbf{r}^l$ will be added to the current set of misclassified views S_o of the corresponding object class o attached to $\mathbf{r}^l$.

4 Every training step it will be checked if $G = G_{\max}$, if so we insert for each $S_o \neq \emptyset$ a new iLVQ node. If more than one representative $\mathbf{r}^l$ of class o was wrongly classified, it must be decided which $\mathbf{r}^l$

is used to initialize the new iLVQ node. For the initialization of the new iLVQ node of class o we determine the index of the iLVQ representative l_{min} with minimal distance to the wrongly classified elements in S_o according to:

$$l_{min} = \arg \min_{l|\mathbf{r}^l \in S_o} ||\mathbf{r}^l - \mathbf{w}^{k_{\min}}(\mathbf{r}^l)||, \tag{3.8}$$

where $\mathbf{w}^{k_{\min}}(\mathbf{r}^l)$ is the winning iLVQ node for view $\mathbf{r}^l$. Insert a new iLVQ node with $\mathbf{w}^{K+1} = \mathbf{r}^{l_{min}}$. Reset $G = 0$ and $S_o = \emptyset$ for all o.

5 Start a new training step (goto step 1) until sufficient convergence is reached.

Our proposed LTM model defines extensions to the LVQ network architecture, which are necessary to fulfill the given incremental and life-long learning object recognition task. Especially the definition of an individual node learning rate or the definition of a node insertion rule are methods also used by Hamker (2001) and Furao & Hasegawa (2006). They propose node insertion based on accumulated errors of each individual node, whereas we only observe the wrong classification itself. If some classification errors occur, nodes are inserted for every wrongly classified object class. Also the initialization of the new nodes differs, we add nodes near class borders but based on a wrongly classified training vector, whereas Hamker and Furao & Hasegawa insert a new node in the neighborhood of an already existing node, for which activation does not occur necessarily. On the contrary, this slows down the learning algorithm, because such a node may not contribute to the representation. Based on the proposed node deletion criteria of both authors the detection of such useless nodes requires several training steps.

3.3 Experimental Results

In the following we describe experiments on using the coupled STM and LTM architecture in a recognition scenario for freely rotated objects. We describe the resulting image ensemble shown in Fig. 3.5 and specify how we do the preprocessing for segmenting the objects.

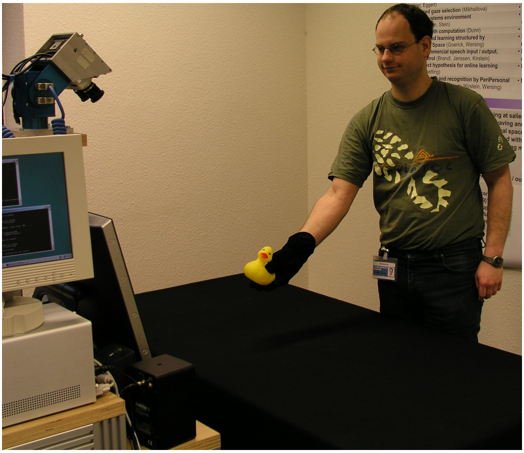
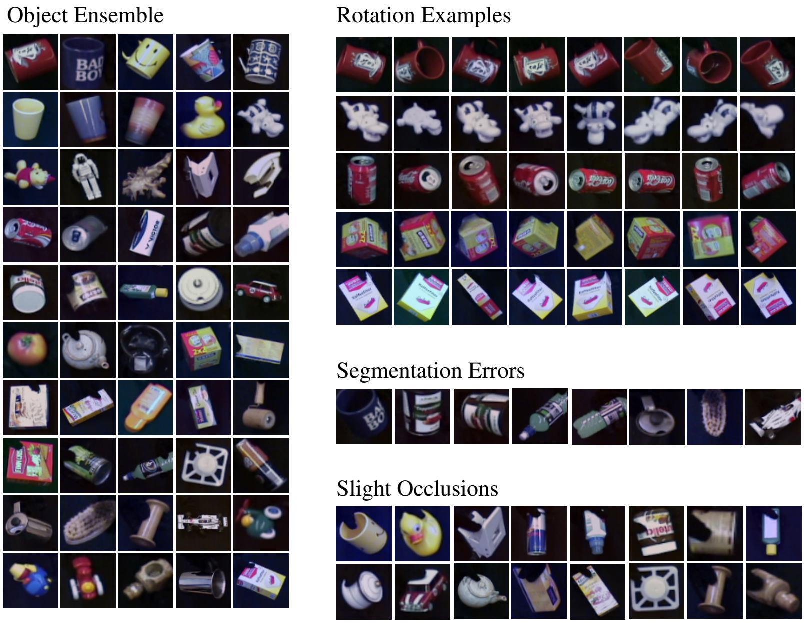


Figure 3.4: **Experimental Setup.** Objects are rotated freely by hand in front of a camera. Additionally we use a black glove and show the objects in front of a black table to simplify the foreground-background separation, which is not the focus of this chapter. Using our short-term memory model the recognition system can be trained online to recognize 50 different objects.

3.3.1 Experimental Setup

For our experiments we use a setup, where we show objects, held in hand and freely rotate them around three axes (see Fig. 3.4). To ease figure-ground segmentation we use a black glove and rotate the objects in front of a black background. The color images are taken with an analog camera and are segmented with a simple local entropy-thresholding (Kalinke & von Seelen, 1996) method. In Chapter 5 a larger integrated system is introduced that relaxes the strong constraints on the background using more advanced segmentation methods (Steil et al., 2007; Denecke et al., 2009), allowing object identification based on hand-held objects in cluttered office environments.

After the segmentation of the object view we normalize it in size (64x64 pixels). For collecting the database we rotated every object freely by hand for some minutes, such that 750 training views $\mathbf{J}^i$ for each object are collected. Another independently taken set of 750 images for each of the objects is recorded as validation database. Figure 3.5 shows all 50 different objects of our HRI50 database. The difficulty of this



Object Ensemble

Rotation Examples

Segmentation Errors

Slight Occlusions

Figure 3.5: **HRI50 Object Ensemble**. On the left side all 50 freely rotated objects are shown, taken in front of a dark background and using a black glove for holding. Additionally some rotation examples, some segmentation, and minor occlusion effects are shown. The main difficulties of this training ensemble are the high appearance variation of objects during rotation around three axes, and shape similarity among cans, cups and boxes, combined with segmentation errors, and slight occlusions.

database results from the high appearance variation of objects during rotation around three axes. The database also contains a lot of objects which are similar in shape or color, e.g. the different cups, boxes or cans. Furthermore some rotation examples for different objects, some segmentation errors and minor occlusion effects are shown in Fig. 3.5.

3.3.2 Feature Extraction

As a feature representation for the incremental learning of complex-shaped objects a feed-forward feature extracting hierarchy (Wersing & Körner, 2003) is used. This feature extraction method illustrated in

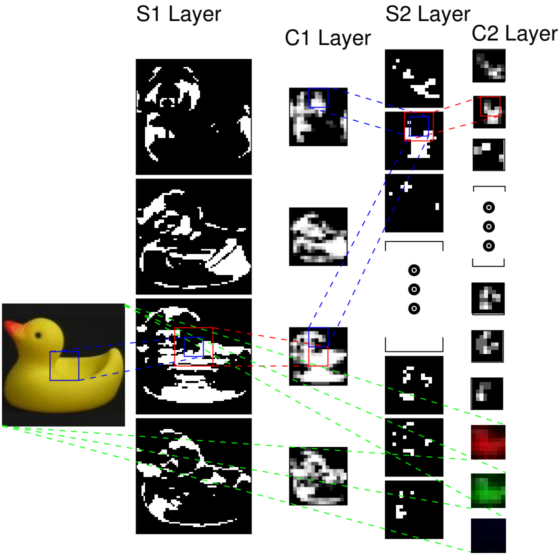


Figure 3.6: **Feature Extraction.** For our object identification task a feed-forward feature extraction hierarchy is used. This hierarchy is composed of a succession of feature detection and pooling layers. The output of the final C2 layer are 50 topographically organized feature maps, where a single feature responds to a local patch in the input image. Furthermore for some experiments coarse color information is added, based on down-sampled RGB maps.

Fig. 3.6 is a model of the human ventral pathway and is based on weight-sharing and a succession of feature-sensitive and pooling stages (see Appendix A.1.1 for details). Wersing & Körner (2003) could show that this feature extraction method allows robust object recognition that is competitive with other state-of-the-art models on benchmark data sets. The used hierarchy is composed of four layers. The first feature extraction layer S1 is composed of four orientation sensitive Gabor filters. Based on these filter responses a winners-take-most operation between features at the same position is applied to suppress submaximal responses. This activity is then passed through a threshold function. Afterwards a pooling operation is performed in the second layer C1 to increase the invariance of feature responses. In layer three feature combinations like corners and T-junctions are detected. Overall 50 different detectors are used in the S2 layer that were determined through a sparse coding method. The final C2 layer again performs a pooling operation. The output of this feature extraction hierarchy are 50 different topographically organized C2 feature maps, while a single C2 feature responds to a local patch in the input image. For our life-long learning method these feature maps are concatenated into a high-dimensional but sparsely activated feature vector $\mathbf{x}^i(\mathbf{J}^i)$. Typically only about 10-30% of all C2 features are active for a given input stimulus allowing an efficient handling of these high-dimensional vectors $\mathbf{x}^i$. For some experiments additional coarse color information is added to the feature vector $\mathbf{x}^i$, based on three down-sampled RGB maps, where each map has the same dimensionality as one of the C2 shape feature maps.

3.3.3 Online Vector Quantization to Build a Short-Term Memory

In the first experiment we investigate the time necessary for training the template-based oVQ short-term memory with up to 50 objects, and evaluate the recognition performance. The training speed is limited by i) the frame rate of the used analog camera for image capturing (12.5 Hz), ii) the computation time needed for the entropy segmentation, iii) the extraction of the corresponding sparse C2 feature vector $\mathbf{x}^i$ with 3200 shape dimensions and 192 optional color dimensions, and iv) the calculation of similarities A^{il} (see Section 3.2). As a good compromise between the representational accuracy and the required learning time the similarity threshold was set to $\epsilon_{stm} = 0.85$ for this experiment with the HRI50 database. Additionally there was no limit imposed on the number of STM representatives. Altogether we achieve an average frame rate of 7 Hz on a 3GHz Xeon processor. Figure 3.7 shows how long it takes until a newly added object can be robustly separated from all other objects. For the shown curves of a cup and a can from our database we trained 9 and 49 objects, respectly, and incrementally added the cup or can as an additional object. Every three seconds ($\approx$ 20 training views) the correct classification rate of the current object is computed using the 750 views from the disjoint test ensemble. Additionally we show the learning curves, averaged over 20 randomly chosen object selections. On average, training of one object can be done in less than 2 minutes, with rapid convergence.

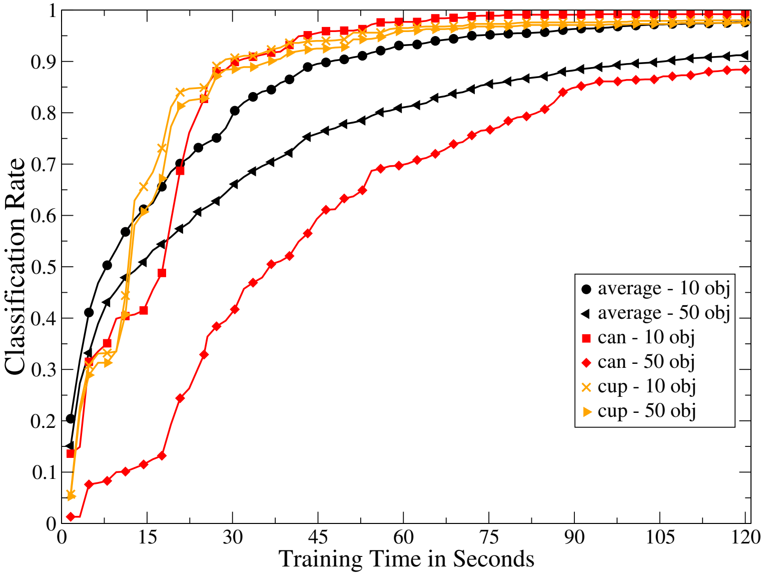


Figure 3.7: **Identification Performance Related to the Training Time.** Classification rate of two selected objects dependent on the training time for learning the 10th and 50th object, and the same learning curves averaged over 20 object selections. While training proceeds, every three seconds ($\approx$ 20 training views) the classification rate is measured based on all 750 available test views of the current object. Good recognition performance can be achieved within two minutes, also for the 50th object.

To evaluate the quality of the feature representation obtained from the visual hierarchy, we performed a systematic comparison with three different types of feature vectors. The first kind of feature vectors contains only shape information of the objects and has 8x8x50 dimensions (8x8 activations for each of the 50 extracted shape feature maps). The second type of vectors with a dimension of 8x8x(50+3) features contains shape and additional coarse color information. Finally we used plain 64x64x3 pixel RGB images as input vectors $\mathbf{x}^i$ for the oVQ model. Due to the high dimensionality and lack of sparsity we can only represent up to 17.000 representatives in this case, otherwise the memory limit of the used operating system was exceeded. This plain image setting also captures the baseline similarity of this ensemble, and can serve as a reference point, since there are currently no other established standard methods for online learning available. Additionally we varied the similarity threshold ϵ_{stm} to investigate the tradeoff between representation accuracy and classification errors. The results are shown in Fig. 3.8. Each symbol of the STM graphs in Fig. 3.8 corresponds to a particular threshold ϵ_{stm}. For a given ϵ_{stm} we let our short-term memory model decide, which training vectors are necessary and calculate the classification rate based on the selected representatives. For a fair comparison, error rates for roughly equal numbers of chosen representatives should be compared. Using the hierarchical shape features reduces the error rates considerably, compared to the plain color images, especially for small numbers of representatives. The addition of the three coarse RGB feature maps additionally reduces error rates by about one third. For a complete training of all 50 objects with a real camera, accomplished within about three hours, the remaining classification error is about 6% using color and shape features and 8% using only shape information.

3.3.4 Incremental LVQ to Build a Long-Term Memory

The performance of the proposed iLVQ long-term memory model is shown in Fig. 3.8 in relation to the results obtained from the STM model. We compare the effect of using only a limited STM memory history (limited to the recent 10 objects) for the transfer into the LTM representation, compared to the usage of unlimited STM. For the experiments with the iLVQ networks we used a similarity threshold $\epsilon_{stm} = 0.85$ for the STM model and applied this threshold to the STM training with shape features and also combined shape and coarse color features. This threshold was chosen as a compromise between the resulting generalization performance for both feature representations and the number of selected STM representatives.

With our LTM model we are able to strongly reduce the necessary number of representatives from about 27000 STM representatives to less than 3800 LTM iLVQ nodes using shape and color features. However this is achieved at the price of a slightly reduced performance of 91.1% correct classification, compared to the performances of the STM representatives which reaches a classification performance of 94.2% at the given value of ϵ_{stm}. If we compare the STM setting, where the classification rate matches approximately 91%, which corresponds to a lower similarity threshold of $\epsilon_{stm} = 0.7$, the number of representatives is still three times

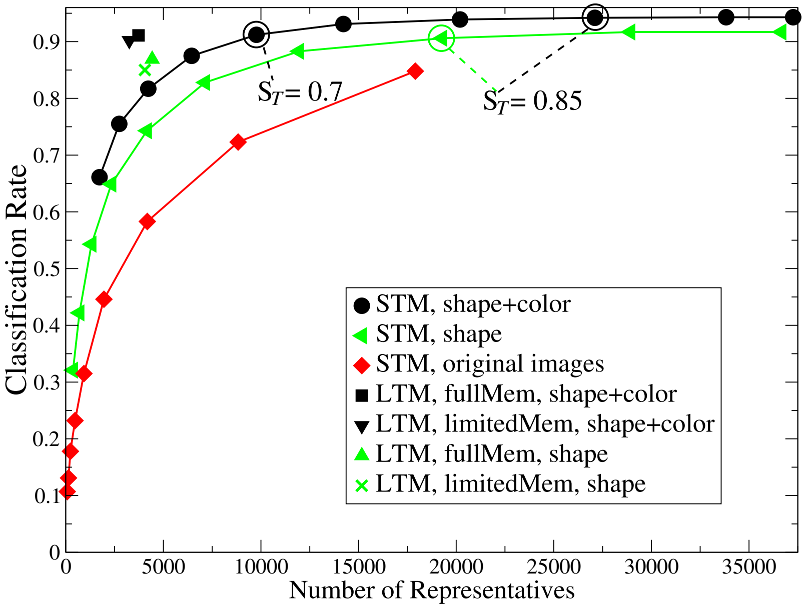


Figure 3.8: **Comparison of the STM and LTM Model.** Comparison of object identification performance of the STM and LTM model for the HRI50 database. The performance measurements of the STM model are calculated for different similarity thresholds ϵ_{stm} and different feature sets, whereas the LTM model was trained with limited or unlimited STM using shape and coarse color information. It can be seen that the use of the visual hierarchy shape features reduces the error rate, compared to the plain color images. The additional use of coarse color features again reduces the error rates of the STM model considerably. For the LTM model tests a similarity threshold of $\epsilon_{stm} = 0.85$ was used for training the STM model, where its representatives $\mathbf{r}^l$ serve as input for the LTM. It can be seen that the LTM model reduces the required resources from about 27000 STM representatives to less than 3800, with a slightly reduced classification performance. Further it should be mentioned that the iLVQ reaches nearly the same classification performance for the limited STM compared to the unlimited case.

larger than for the LTM, as can be seen in Fig. 3.8.

For a better comparison of our LTM model to other state-of-the-art approaches, experiments with the well-known COIL-100 database (Nayar et al., 1996) are performed. This database consists of 100 different objects rotated around one axis, where the 72 different views for each object are taken at pose intervals of 5^o. For our experiments we resized the original images to 64x64 pixels to allow a better comparison to our own HRI50 database. For all experiments with the COIL-100 database, 36 object views (10^o apart) are used for training and the remaining views for testing.

Additionally we compare our architecture to a Single Layer Perceptron (SLP) and the SNOW (Roth et al., 2002) approach. The SLP network architecture consists of an input and an output layer, without hidden layers. For every object o we used one output node, whereas the output p_o^i of each node for feature vector $\mathbf{x}^i$ is calculated based on a linear scalar product activation and a Fermi transfer function:

$$p_o^i = \frac{1}{1 + \exp(-\mathbf{w}_o * \mathbf{x}^i)}, \tag{3.9}$$

where $\mathbf{w}_o$ are the weights of node o. The SNOW approach is specially designed for a sparse feature representation as used in our experiments. It is also better suited for incremental and life-long learning compared to the SLP due to its conservative learning schema. The SNOW model is based on a multiplicative Winnow update rule (Littlestone, 1988), which is applied to wrongly classified training vectors only. Furthermore exclusively weights of currently activated input dimensions are modified at a training step, which theoretically provides more life-long learning stability than sigmoidal networks where typically all weights are updated at each learning step. For SNOW we used the same network size as for the sigmoidal networks, i.e. one output node for each object.

For the comparison of the iLVQ, SLP and SNOW approach we performed a systematic analysis using all available training data of the used image ensemble, compared to the use of the proposed STM model and a limited STM, where only the recent 10 objects are available for training. For the build up of the corresponding STM representation we have chosen for the COIL-100 database a similarity threshold of $\epsilon_{stm} = 0.9$, while for the HRI50 image ensemble $\epsilon_{stm} = 0.85$ was selected. These values where opted as a compromise between the total number of representatives and the identification performance of each database. Furthermore we compare the results achieved with two different feature ensembles based on the C2 shape features and the use of additional coarse color features. The results of this comparison are shown in Table 3.1 for the COIL-100 database and Table 3.2 for the HRI50 database.

	shape			shape and coarse color		
	all	STM	lim. STM	all	STM	lim. STM
iLVQ	98.6%	97.9%	96.0%	99.5%	99.3%	98.4%
SLP	99.9%	99.5%	28.0%	99.9%	99.8%	27.6%
SNOW	96.5%	94.2%	59.2%	97.6%	96.7%	50.0%

Table 3.1: **Comparison of the iLVQ, SLP, and SNOW Identification Performance Using the COIL-100 Database.** Classification rates of all three approaches are shown based on C2 shape features and the combination of shape and coarse color features. Additionally we compare the results using all available training data, the use of the proposed STM with $\epsilon_{stm} = 0.9$ and a capacity limited STM (restricted to the recent 10 objects).

For the COIL-100 database (see Table 3.1) it can be seen that the Single Layer Perceptron achieves better classification results as our proposed iLVQ method for the cases where no limit on the training data was imposed. The SNOW network is slightly worse than iLVQ and SLP, but the classification rate is still comparable to other state-of-the-art approaches applied to this database. It should be noted that the performance we achieved with our C2 shape features representation is superior to the results published by (Roth et al., 2002) (one-against-all: 90.52%), which highlights the quality of the hierarchical feature representation. For all three models, the introduction of the STM model with approximately 30% reduction of training data causes only minor increase in errors. For the experiments using only a limited STM of 10 objects, it can be seen that only the iLVQ method can handle this with almost no performance loss. Although the performance decrease of the SNOW approach is distinctly less than for SLP, both methods quickly fail to distinguish objects from earlier training phases, resulting in low recognition rates. This is the well-known "catastrophic forgetting effect" (Hamker, 2001).

The results obtained with the HRI50 database are shown in Table 3.2. In comparison to the COIL-100 results, the iLVQ method achieves better results on this more difficult database than the SLP approach, which is most distinct for the use of shape features only. This better performance is mainly caused by the incremental learning of the iLVQ approach allowing an adaptation to the difficulty of the identification task, while the SLP approach does not allow incremental learning. It can also be seen that the SNOW approach cannot capture the higher appearance

	shape			shape and coarse color		
	all	STM	lim. STM	all	STM	lim. STM
iLVQ	88.5%	86.9%	85.8%	91.6%	91.1%	90.2%
SLP	84.1%	80.7%	21.9%	91.2%	91.1%	21.7%
SNOW	52.8%	51.9%	20.3%	55.6%	54.2%	20.7%

Table 3.2: **Comparison of the iLVQ, SLP, and SNOW Identification Performance Using the HRI50 Database.** Classification rates of all three approaches are shown based on C2 shape features and the combination of shape and coarse color features. Additionally we compare the results using all available training data, the use of the proposed STM with $\epsilon_{stm} = 0.85$ and a limited STM.

variation of the HRI50 database, which results in poor identification performance. For the training with the limited STM the iLVQ also achieves good results on the HRI50 database. In contrast to the COIL-100 database the SNOW approach is also worse than SLP for the limited STM experiments, which is mainly due to the overall poor performance of SNOW on the HRI50 database.

3.4 Discussion

In this chapter we have proposed a biologically motivated approach for the learning of visual object representations. It is based on a hierarchical feature extraction model serving as the input for a coupled short-term and long-term memory. Our main focus was to demonstrate the capability of online learning of many complex-shaped objects in combination with a model for a consolidation of the fast but limited short-term memory into a condensed long-term memory representation. In the following we discuss the components of our model with reference to related work.

Our feature detection approach is different from most of the related work on online learning for object recognition (Garcia et al., 2000; Steels & Kaplan, 2001; Roy & Pentland, 2002; Arsenio, 2004; Bekel et al., 2004), because the representation is not based on a dimension reduction of the high-dimensional visual input. Due to the receptive-field-based topographical representation of the used C2 features, we obtain multiple shape feature-map representations with a resulting dimensionality that is of the same order as the visual input. Within the maps, however, only sparse activation is present, which is caused by the coding strategy in the hierarchical network.

The short-term memory model is defined as a template-based representation that adds new object representatives using a Euclidean metrics within the high-dimensional space of shape and color feature map responses. Due to the purely incremental nature of this learning method we can perform online learning of objects by capturing sufficient appearance variation of the object under investigation. Adaptive resonance (ART) networks are another common approach to perform one-shot and online learning. Many applications of ART and its relative Fuzzy ARTMAP have so far concentrated on representation spaces with much lower dimensionality (Carpenter et al., 1992). The necessity of complement coding (see discussion in Section 3.2.1), doubling the input space dimensionality, and problems with sparse vectors make ART networks not very suitable for representing the feature activations of the visual hierarchy we use here.

For the application to online learning, using only the STM model achieved good generalization in combination with a large storage capacity of 50 objects, compared to other work on online learning of objects, which usually did not consider more than 10-12 objects (Bekel et al., 2004; Arsenio, 2004). This capacity is a direct consequence of the high-dimensional representation space, and is also achieved if only shape representations are used. The STM model enables learning in direct interaction with a human teacher, whereas the long training time of most current recognition architectures does not allow this user interaction. However, the representational effort of storing a large number of high dimensional feature maps can be large. To overcome this limitation we introduced a long-term memory model.

Our long-term memory model has to satisfy the two main requirements: It has to incrementally add and consolidate representational resources dependent on the complexity of the objects to be learned. Furthermore it has to care for the “stability-plasticity dilemma” caused by using only a limited STM memory of the previous object presentations. Due to the problems of standard architectures like MLPs, which suffer from catastrophic forgetting in such a scenario, most previous work on online object learning does not consider incremental learning, but rather collects the training data and then performs a standard batch learning procedure (Bekel et al., 2004).

As a demonstration of the "catastrophic forgetting effect" we performed experiments with the SLP and SNOW approach and could show a strong degradation of classification performance for our desired interactive and life-long learning task. Additionally we performed experiments with the COIL-100 database for a better evaluation of our HRI50 image ensemble. We could show that the LTM model can reach state-of-the-art recognition performance for the COIL-100 database. In direct comparison the HRI50 image ensemble is more challenging due to distinctly lower classification rates. The difficulty of the HRI50 database is caused by object rotation around three axes, whereas the COIL-100 objects are only rotated around one axis. This results in much higher appearance variations, which pose problems for the SNOW approach, while the iLVQ approach automatically scales to the difficulty of the recognition tasks resulting in good recognition rates for more challenging databases.

We have based our LTM architecture on a Learning Vector Quantization (LVQ) model, which we have extended by methods of incremental node insertion, and flexible adaptation of the local node learning rates. Our approach can be compared to recent work on life-long learning for incremental neural architectures (Hamker, 2001; Furao & Hasegawa, 2006), targeting learning for non-stationary distributions without destruction of previously learned representations (see Section 3.2.2). Our iLVQ algorithm differs from the work of Hamker and Furao & Hasegawa mainly in the node insertion rule. We insert neurons only if classification errors during the training phase occur and do not utilize the accumulated error of the nodes themselves. We assume that this leads to a smaller number of allocated resources compared to the distance-based insertion mechanism, especially in high-dimensional spaces. Hamker has demonstrated the efficiency of his proposed LLCS networks based on several low dimensional non-stationary benchmark datasets. How this network architecture performs on more realistic problems with high-dimensional input spaces can, however, only be speculated until now. Furao & Hasegawa (2006) applied the proposed method to a setting of face clustering, but it seems that the unsupervised learning method is not efficient in high-dimensional input spaces with strong variation. This may be the reason for the use of smoothed input images in their experiments.

Hamker and Furao & Hasegawa propose utility measurements to detect rarely activated nodes or to decide if the insertion of a node was ineffective and does not cause a decreasing error rate. The drawback of the proposed methods is that they tend to delete nodes representing rarely occurring data with only very few feature vectors, which are typically quite important in our scenario where objects are rotated freely by hand. Especially the LLCS (Hamker, 2001) utility measurements delete nodes that are not supported by other nodes in their direct neighborhood. The deletion of such nodes slows down the learning process and can also destroy parts of the representation, which infrequently occur again. Although we did not care for an explicit node deletion procedure in our iLVQ model, we think that similar mechanisms of utility measurements could be advantageous for reducing the representational effort in the LTM model.

Chapter 4

Life-Long Learning for Categorization Tasks

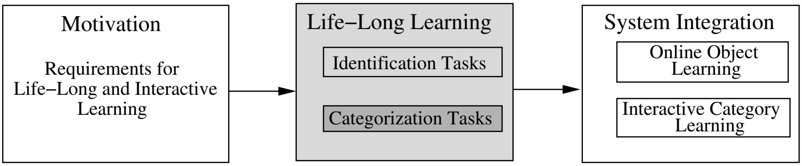


Figure 4.1: **Life-Long Learning for Categorization Problems.** In this chapter a life-long learning method for interactive learning of categories is proposed. This learning method enables the memory consolidation from the object-specific short-term memory proposed in Chapter 3 into a category-specific long-term memory. The basic idea of the incremental build up of the LTM representation is to combine an extended LVQ network with a category-specific feature selection.

In Chapter 3 we proposed a memory model for life-long learning of arbitrary identification tasks. In the following chapter we concentrate on more challenging categorization tasks. Therefore based on the previously developed STM (see Chapter 3) a memory consolidation from an object-specific STM into a category-specific LTM representation (Kirstein et al., 2008b; Kirstein et al., 2009) is proposed. The proposed LTM learning model combines an incremental exemplar-based neural network with a dynamic feature scoring and selection technique to enable life-long learning of arbitrary categories. The target of this learning method is a flexible category representation that is capable to deal with complex-shaped objects with high appearance variations. The presented experimental results are based on (Kirstein et al., 2009), where fast learning combined with good generalization performance could be shown.

4.1 Comparison between Identification and Categorization Tasks

The major target when dealing with categorization tasks is to achieve a higher generalization performance compared to identification tasks. For identification tasks commonly the mapping from all entities (e.g. object views) of an instance (e.g. a physically object) to a given class label is learned. Furthermore the learning algorithm tries to optimally separate each instance from all other classes as illustrated in Fig. 4.2. Therefore a well trained identification system normally can generalize to unseen entities of the learned instances, but typically the generalization to similar but novel instances is very limited. In contrast to this for categorization tasks normally a group of instances with common properties (e.g. the basic shape) are assigned to a single label. To achieve this mapping the learning method must be able to extract the reoccurring properties to decide if a category is present in the current entity or not. This means that instance-specific information should be neglected, whereas for identification tasks typically these details are used to distinguish the different classes. Due to the extraction of reoccurring activation patterns categorization architectures enable good generalization to other instances sharing the properties of the corresponding category.

4.2 Related Work

For the life-long learning of arbitrary categories we combine an exemplar-based neural network with a category-specific feed-forward feature selection method, where the incremental and life-long learning of both parts is the major novelty of our proposed method. Although our approach is applicable to any kind of categories we concentrate in this chapter on a challenging visual categorization task, where we apply our methods to rotated and complex-shaped objects. Besides incremental and life-long learning we are additionally targeting for fast interactive learning that

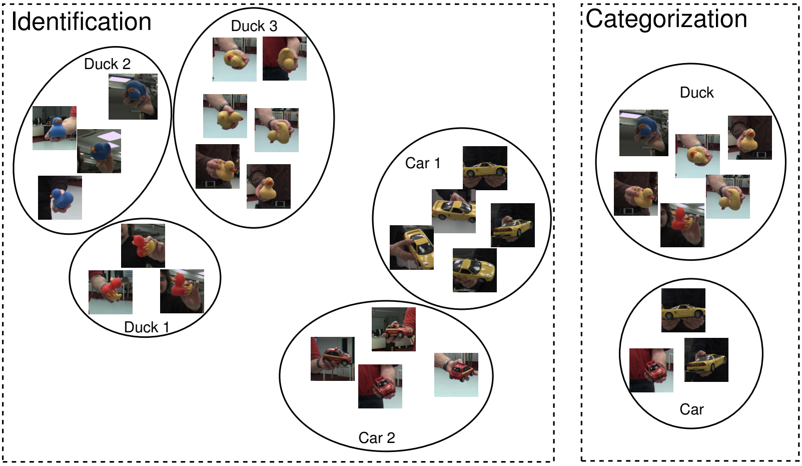


Figure 4.2: **Comparison between Identification and Categorization Tasks.** For identification tasks commonly the mapping from all entities of a physical instance (e.g. an object) to a given class label is learned. Additionally each instance is separated from all other instances. Therefore learning methods for identification problems try to extract the most distinctive features to achieve this learning target. In contrast to this for categorization tasks normally a group of instances, sharing common properties, are assigned to a label. For such learning tasks therefore the extraction of these reoccurring properties is targeted. Due to the suppression of instance-specific characteristics, for categorization problems a higher generalization performance compared to identification tasks can be achieved.

allows learning in direct interaction with a human tutor. In the following, related work addressing life-long learning, feature selection, visual categorization, and online learning is discussed in more detail.

4.2.1 Life-Long Learning Architectures

Life-long learning architectures (Hamker, 2001; Polikar et al., 2001; Kirstein et al., 2008), as discussed in Section 3.1 are typically based on exemplar-based learning techniques like the Learning Vector Quantization (LVQ) (Kohonen, 1989) or the Growing Neural Gas (GNG) (Fritzke, 1995). Such neural architectures are beneficial for life-long learning, because for a specific input vector the learning methods modify only small portions of the overall network. Thus stability can be better achieved compared to the Multi Layer Perceptron (MLP), where all weights are modified at each learning step. Furthermore the learning of exemplar-based networks is commonly based on a similarity measurement (e.g. Euclidean distance), where the chosen metric has a strong impact on the generalization performance. To relax this dependency, metrical adaptation methods can be used that individually weight the different feature dimensions as proposed for the Generalized Relevance Learning Vector Quantization (GRLVQ) (Hammer & Villmann, 2002) algorithm.

A common strategy for life-long learning architectures is the usage of a node specific learning rate combined with an incremental node insertion rule (Hamker, 2001; Furao & Hasegawa, 2006; Kirstein et al., 2008). This permits plasticity of newly inserted neurons, while the stability of matured neurons is preserved. The major drawback of these architectures is the inefficient separation of co-occurring visual categories, because typically the complete feature vectors are used to represent the different classes and no assignment of feature vector parts to different classes is considered. To overcome this limitation we propose a category-specific feature selection that enable an efficient separation of co-occurring categories.

4.2.2 Feature Selection Methods

In the context of text categorization, feature selection methods are a common technique to enhance the performance (Yang & Pedersen, 1997), while for visual categorization tasks feature selection gained distinctly less interest. One exception are approaches based on boosting (Viola & Jones, 2001), where the feature selection is an integrated part of the learning method. In contrast to this, category-specific feature selection is considered to be an important part for our categorization approach. This is because commonly only a small subset of extracted features is relevant for a specific category, while the other features are irrelevant or even can cause confusions. Furthermore small category-specific feature subsets are beneficial with respect to the computational costs to allow fast interactive learning. Therefore in the following a brief overview of different feature selection techniques is given.

There are basically three groups of feature selection methods, namely filter, wrapper and embedded methods (Guyon & Elissee, 2003). Filter methods (see Forman (2003) for an overview) are independent of the used classifier and commonly select a subset of features as a preprocessing step. The corresponding feature selection is typically based on some feature ranking method (Furey et al., 2000; Kira & Rendell, 1992), but also the training of single variable classifiers is used. The second group of feature selection methods are wrapper methods (Kohavi & John, 1997). Similar to the filter approaches these wrapper methods are independent of the underlying recognition architecture but they use the learning algorithm as a “black box” to weight different feature subsets (e.g. based on the training error). Due to the incorporation of the learning method to guide the feature selection process and to evaluate the different feature subsets, wrapper methods are considered to select more class-specific features sets compared to filter methods (Guyon & Elissee, 2003). Wrapper methods furthermore can be categorized into backward and forward selection methods, where the backward selection starts with a full set of features and iteratively eliminates irrelevant features. In contrast to this, forward selection methods start with an empty set of features and incrementally add new features. Such methods are beneficial with respect to interactive learning, because they enable fast learning. Therefore we propose a novel forward selection wrapper method for our categorization task. The last group of feature selection methods are the so-called embedded methods. Here the feature selection is an integrated part of the recognition architecture and is optimized together with the network parameters, so that these methods usually can not be transferred to other learning approaches. One strategy of this group is to add sparsity constraints to the error function (Perkins et al., 2003) resulting in a pruning of irrelevant features.

4.2.3 Visual Category Learning Approaches

In the recent years many architectures dealing with categorization tasks have been proposed in the computer vision research field. Such category learning approaches can be partitioned into generative and discriminative models (Fritz, 2008). Generative probabilistic models, as proposed by Leibe et al. (2004), Fei-Fei et al. (2003), Fergus et al. (2003) or Mikolajczyk et al. (2006), first model the underlying joint probability $P(\mathbf{x}, t_c)$ for each category t_c and all training examples $\mathbf{x}$ individually and afterwards use the Bayes theorem to calculate the posterior class probability $p(t_c|\mathbf{x})$ (Bishop, 2006). The advantages of generative models are that expert knowledge can be incorporated as prior information and that those models usually require only a few training examples to reach a good categorization performance. In contrast to this, discriminant models directly learn the mapping from $\mathbf{x}$ to t_c based on a decision function $\Phi(\mathbf{x})$ or estimate the posterior class probability $P(t_c|\mathbf{x})$ in a single step (Ng & Jordan, 2001). Common approaches for this group of categorization models are based on Support Vector Machines (Heisele et al., 2001), boosting (Viola & Jones, 2001; Opelt et al., 2004) or SNOW (Agarwal et al., 2004). Such discriminant models tend to achieve a better categorization performance compared to generative models if a large ensemble of training examples is available (Ng & Jordan, 2001).

In general most categorization approaches are robust against partial occlusions, scale changes, and are able to deal with cluttered scenes. However, many models have only been demonstrated to work with data sets restricted to canonical views of categories (e.g. side views of cars). Thomas et al. (2006) try to overcome this limitation by training several pose-specific implicit shape models (ISM) (Leibe et al., 2004) for each category. After the training of these ISMs, detected parts from neighboring pose-dependent ISMs are associated by so-called "activation links". These links then allow the detection of categories from many viewpoints. Additionally categorization architectures are commonly designed for offline usage only, where the required training time is not important. This makes them unsuitable for our desired interactive training. Recent work of Fritz et al. (2007) and Fei-Fei et al. (2007) addresses this issue by proposing incremental clustering methods, which in general allow interactive category learning, but still these approaches are restricted to the canonical views of the categories.

4.2.4 Online and Interactive Learning

The development of online and interactive learning systems has become increasingly popular in the recent years (Roth et al., 2006; Steels & Kaplan, 2001; Arsenio, 2004; Wersing et al., 2007a). Most of these methods were not applied to categorization tasks, because their learning methods are unsuitable for a more abstract and variable category representation. The work of Skočaj et al. (2007) is of particular interest with respect to online and interactive learning of categories. It enables learning of several simple color and shape categories by selecting a single feature that describes the particular category most consistently. Finally the corresponding category 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). Although this category learning architecture shares some common targets with our proposed learning method, the restriction to a single feature only allows the representation of categories with little appearance changes. This is basically because more complex categories typically require several features to adequately represent all category instances. To avoid this limitation we propose a forward feature selection process that incrementally selects an arbitrary number of features if they are required for the representation of a particular category.

4.3 Life-Long Learning of Categories

Our categorization memory architecture illustrated in Fig. 4.3 is based on a limited and object-specific short-term memory (STM) (see Chapter 3) that is transferred into a category-specific long-term memory (LTM) representation. This LTM model is based on an exemplar-based incremental learning network combined with a forward feature selection method to allow life-long learning of arbitrary categories. Both parts are optimized together to find a balance between insertion of features and allocation of representation nodes, while using as little resources as possible. This is crucial for interactive learning with respect to the required computational costs. In the following we refer to this architecture as category Learning Vector Quantization (cLVQ).

To achieve the interactive and incremental learning capability the exemplar-based network part of the cLVQ method is used to approach the "stability-plasticity dilemma" of life-long learning problems. Commonly for LVQ networks the number of nodes for each class has to be predefined. Thus experiments normally are repeated with different numbers of nodes to find a network size adequate for the difficulty of the corresponding learning problem. Such a repetition of experiments is unsuitable for interactive learning. Thus we define a node insertion rule that automatically determines the number of required nodes. The final

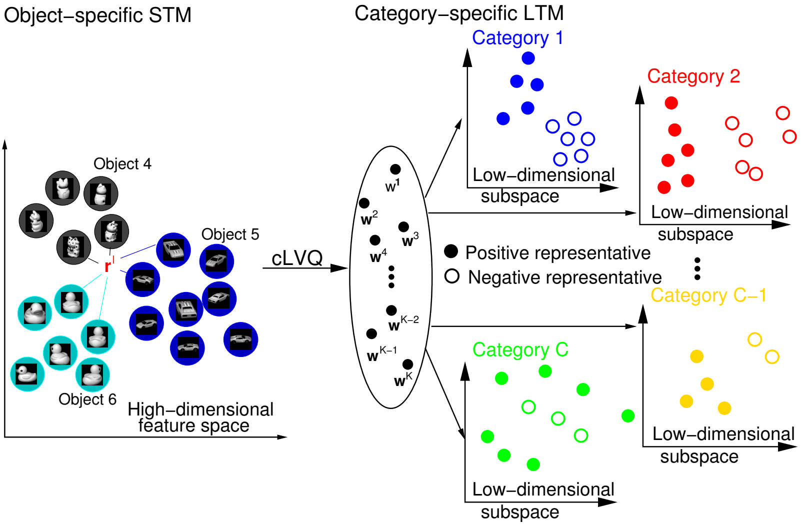


Figure 4.3: **Illustration of the Life-Long Learning Model for Categorization Tasks.** For the category learning model the same limited short-term memory (STM) representation can be used as proposed in Chapter 3. Therefore the STM is not considered anymore in this chapter. In contrast to the previous model the proposed category Learning Vector Quantization (cLVQ) method allows the transition from a object-specific STM to a category-specific long-term memory (LTM). This is achieved by combining an exemplar-based neural network approaching the "stability-plasticity dilemma" with a category-specific feature selection. This allows the separation of co-occurring categories (e.g. if an instance belongs to several categories) and the definition of different metrical "views" to a single node $\mathbf{w}^k$. The categorization decision itself is based on the allocated cLVQ nodes $\mathbf{w}^k$ and the low-dimensional category-specific feature spaces.

number of allocated nodes $\mathbf{w}^k$ corresponds to the difficulty of the different categories itself but also to the within-category variance. Finally the long-term stability of these incrementally learned nodes is considered as proposed by (Kirstein, Wersing, & Körner 2008).

Additionally for our learning approach a category-specific forward feature selection method is used to enable the separation of co-occurring categories, because it defines category-specific metrical "views" on the nodes of the exemplar-based network. During the learning process it selects low-dimensional subsets of category-specific features by predominantly choosing features that occur almost exclusively for a certain category. Furthermore only these selected category-specific features are used to decide whether a particular category is present or not. For guiding this selection process a feature scoring value h_{cf} is calculated for each category c and feature f. This scoring value is only based on previously seen exemplars of a certain category, which can strongly change if further information is encountered. Therefore a continuous update of the h_{cf} values is required to follow this change.

4.3.1 Distance Computation and Learning Rule

Learning in the cLVQ architecture is based on a set of high-dimensional and sparse feature vectors $\mathbf{x}^i = (x_1^i, \dots, x_F^i)$, where F denotes the total number of features. Additionally each vector $\mathbf{x}^i$ is assigned to a list of category labels $\mathbf{t}^i = (t_1^i, \dots, t_C^i)$. We use C to denote the current number of represented color and shape categories, whereas 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.

The cLVQ representative vectors $\mathbf{w}^k$ with $k = 1, \dots, K$ are built up incrementally, where K denotes the current number of allocated vectors $\mathbf{w}$. Each $\mathbf{w}^k$ is attached 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 weighted Euclidean distance with specific weight factors λ_{cf} similar to the Generalized Relevance Learning Vector Quantization (GRLVQ) method proposed by Hammer & Villmann (2002):

$$d_c(\mathbf{x}^i, \mathbf{w}^k) = \sum_{f=1}^{F} \lambda_{cf} (x_f^i - w_f^k)^2, \tag{4.1}$$

where the category-specific weights λ_{cf} are updated continuously. We denote the set of selected features for an active 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 $\mathbf{w}^{k_{\min}(c)}(\mathbf{x}^i)$ are calculated independently for each category c, where $k_{\min}(c)$ is deter-

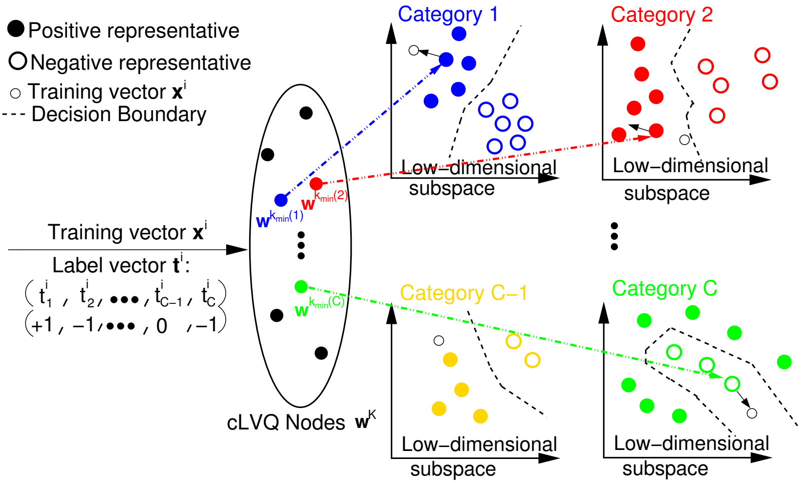

● Positive representative
○ Negative representative
○ Training vector $\mathbf{x}^i$
--- Decision Boundary
Category 1
Category 2
Low−dimensional subspace
Low−dimensional subspace
Training vector $\mathbf{x}^i$
Label vector $\mathbf{t}^i$:
$(t^i_1, t^i_2, \bullet\bullet\bullet, t^i_{C-1}, t^i_C)$
$(+1, -1, \bullet\bullet\bullet, 0, -1)$
$\mathbf{w}^{k_{min}(1)}$
$\mathbf{w}^{k_{min}(2)}$
$\mathbf{w}^{k_{min}(C)}$
Category C−1
Category C
cLVQ Nodes $\mathbf{w}^K$
Low−dimensional subspace
Low−dimensional subspace

Figure 4.4: **Illustration of the cLVQ Learning Rule.** Based on a training vector $\mathbf{x}^i$ and the corresponding target vector $\mathbf{t}^i$ the winning nodes $\mathbf{w}^{k_{\min}(c)}$ are calculated for each category c independently. For this calculation only the selected features $f \in S_c$ are used, so that the categorization decision is based on the different low-dimensional feature subsets. If the categorization decision was correct, the winning node $\mathbf{w}^{k_{\min}(c)}$ is shifted into the direction of the training vector. Otherwise $\mathbf{w}^{k_{\min}(c)}$ is moved into the opposite direction. If for an $\mathbf{x}^i$ the membership of a category is unknown ($t^i_c = 0$) no adaptation of the prototype node $\mathbf{w}^{k_{\min}(c)}$ is performed.

mined in the following way:

$$k_{\min}(c) = \arg\min_k \; d_c(\mathbf{x}^i, \mathbf{w}^k) \quad \forall k \text{ with } u^k_c \neq 0. \tag{4.2}$$

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

$$w^{k_{\min}(c)}_f := w^{k_{\min}(c)}_f + \mu\, \Theta^{k_{\min}(c)} (x^i_f - w^{k_{\min}(c)}_f) \quad \forall f \in S_c, \tag{4.3}$$

where $\mu = 1$ if the categorization decision for $\mathbf{x}^i$ was correct, otherwise $\mu = -1$ and the winning node $\mathbf{w}^{k_{\min}(c)}$ will be shifted away from $\mathbf{x}^i$. This node adaptation is illustrated in Fig. 4.4. Additionally $\Theta^{k_{\min}(c)}$ is the node-dependent learning rate as proposed by Kirstein et al. (2008):

$$\Theta^{k_{\min}(c)} = \Theta_0 \exp\left(-\frac{a^{k_{\min}(c)}}{d}\right). \tag{4.4}$$

Here Θ_0 is a predefined initial value, d is a fixed scaling factor, and a^k is an iteration-dependent age factor. Similar to the iLVQ approach proposed in Chapter 3 the age factor a^k is incremented every time the corresponding $\mathbf{w}^k$ becomes the winning node.

4.3.2 Feature Scoring and Category Initialization

The incremental category learning of our model is organized in training epochs, where only a limited number of category entries (e.g. object views) are visible to the learning method, emulating a limited short-term memory (STM). After each epoch some of the training vectors $\mathbf{x}^i$ and their corresponding target category values $\mathbf{t}^i$ are removed and replaced by vectors of a new instance. Therefore for each training epoch the scoring values h_{cf}, used for guiding the feature selection process, are updated in the following way:

$$h_{cf} = \frac{H_{cf}}{H_{cf} + \bar{H}_{cf}}. \tag{4.5}$$

The variables 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 ($x_f > 0$). For each newly inserted object view, the counter value H_{cf} is updated in the following way:

$$H_{cf} := H_{cf} + 1 \text{ if } x_f^i > 0 \text{ and } t_c^i = +1, \tag{4.6}$$

where $\bar{H}_{cf}$ is updated as follows:

$$\bar{H}_{cf} := \bar{H}_{cf} + 1 \text{ if } x_f^i > 0 \text{ and } t_c^i = -1. \tag{4.7}$$

The score h_{cf} defines the metrical weighting in the cLVQ representation space. We then choose $\lambda_{cf} = h_{cf}$ for all $f \in S_c$ and $\lambda_{cf} = 0$ otherwise.

For our learning architecture we assume that not all categories are known from the beginning, so that new categories can occur in each training epoch. Therefore if category c with the category label $t_c^i = +1$ occurred for the first time in the current training epoch, 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 scoring value and initialize $S_c = \{v_c\}$. The training vector $\mathbf{x}^i$ is selected as the initial cLVQ node, where the selected feature v_c has the highest activation, i.e. $\mathbf{w}^{K+1} = \mathbf{x}^q$ with $x_{v_c}^q \geq x_{v_c}^i$ for all i. The attached label vector is chosen as $u_c^{K+1} = +1$ and zero for all other categories.

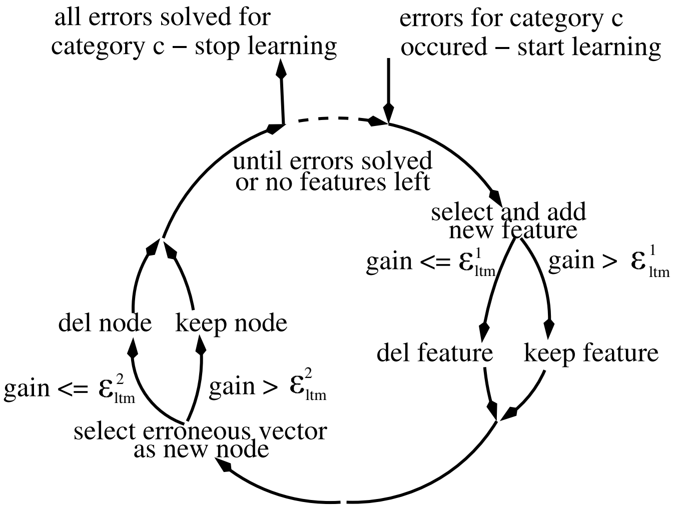

all errors solved for category c – stop learning
errors for category c occured – start learning
until errors solved or no features left
select and add new feature
gain <= $\varepsilon^1_{\mathrm{ltm}}$
gain > $\varepsilon^1_{\mathrm{ltm}}$
del feature
keep feature
del node
keep node
gain <= $\varepsilon^2_{\mathrm{ltm}}$
gain > $\varepsilon^2_{\mathrm{ltm}}$
select erroneous vector as new node

Figure 4.5: **Illustration of the cLVQ Optimization Loop.** The basic idea of this optimization loop is to make small modifications to the representation of categories where categorization errors on the available training vectors occur. 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 it is retracted.

4.3.3 Learning Dynamics

During a single learning epoch of the cLVQ method an optimization loop is performed iteratively (see Fig. 4.5). This loop applies small changes to the representation of erroneous categories by testing new features and representation nodes. A single run through the optimization loop is composed of the following steps:

Step 1: Feature Testing. The target of this step is the addition of features for the category-specific metrics, based on the observable training vectors $\mathbf{x}^i$ and the corresponding categorization errors. Additionally in rare cases also the removal of already selected features is possible. For each category c we determine the set of positive errors E_c^+ as:

$$E_c^+ = \{i | t_c^i = +1 \wedge t_c^i \neq u_c^{k_{min}(c)}(\mathbf{x}^i)\} \tag{4.8}$$

and negative errors E_c^- as:

$$E_c^- = \{i | t_c^i = -1 \wedge t_c^i \neq u_c^{k_{min}(c)}(\mathbf{x}^i)\}. \tag{4.9}$$

Afterwards we compare the total number of positive errors $\#E_c^+$ with the corresponding number of negative ones $\#E_c^-$. If the total number of $\#E_c^+ \geq \#E_c^-$ then we compute:

$$e_{cf}^+ = \sum_{i \in E_c^+} \Phi_{ltm}(x_f^i) / \sum_{i \in E_c^+} 1, \tag{4.10}$$

where for the category-specific LTM the Φ_{ltm} is defined as a Heaviside function.

The score e_{cf}^+ is the ratio of active feature entries for feature f among the positive training errors of class c. We now want to add a feature to the category feature set S_c that potentially improves the categorization performance of category c by having a high scoring value h_{cf} which is also very active for the encountered error set E_c^+. Therefore we choose:

$$v_c = \arg\max_{f \notin S_c} (e_{cf}^+ + h_{cf}) \tag{4.11}$$

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 errors is calculated based on the newly added feature v_c. If the performance increase for category c is larger than threshold ϵ_{ltm}^1, then v_c is permanently added and otherwise it is removed. An analog step is performed if the number of negative errors is larger than the number of positive errors ($\#E_c^+ < \#E_c^-$). The only difference is that a feature is removed and then again the performance gain is computed for the final decision on the removal.

Step 2: cLVQ Node Testing. Similarly to Step 1, we test new cLVQ nodes only for erroneous categories. In previous work concerning identification tasks (see Chapter 3) nodes were inserted for training vectors with smallest distance to wrong winning nodes (Kirstein et al., 2008). In contrast to this, we here insert new cLVQ nodes based on training vectors $\mathbf{x}^i$ with most categorization errors $t_c^i \neq u_c^{k_{min}(c)}(\mathbf{x}^i)$ for all categories C, until for each erroneous category c at least one new node is inserted (see Fig. 4.6). This leads to very compact representations, 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 thresh-

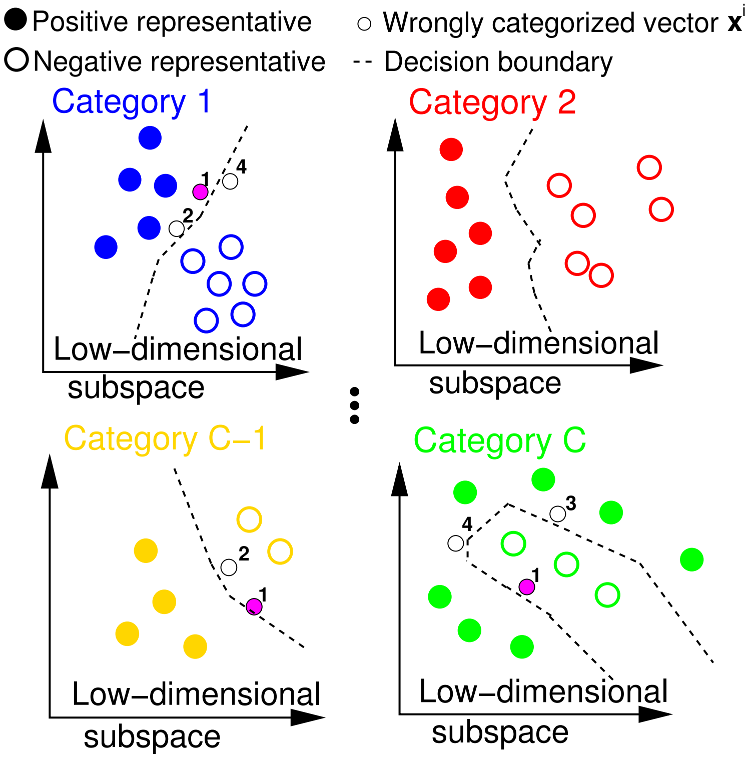


Figure 4.6: **Illustration of the Node Insertion Rule.** We incrementally add new cLVQ prototype nodes $\mathbf{w}^k$ based on wrongly categorized training vectors illustrated with small circles. The positions of these vectors are different in each category-specific subspace and also the number of erroneously labeled categories varies. Thus the wrongly categorized training vectors are labeled with a number to ease this mapping. In the following for each erroneous category at least one new node is inserted. For the insertion we prefer training vectors where the most categorization errors occurred. For the illustrated example only one training vector (highlighted with the small filled circle) causes errors in three different categories. Therefore at the corresponding vector position a new node is inserted. This insertion rule leads to a compact representation, because a single node $\mathbf{w}^k$ potentially improves the representation of several categories.

old 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_c^k of each newly inserted node $\mathbf{w}^k$ to zero, so that node k does not further contribute to the representation of category c. Finally we remove nodes where all u_c^k are zero, which means that no erroneous category for which the node $\mathbf{w}^k$ was originally inserted reached a performance gain above ϵ_{ltm}^2.

Step 3: Stop condition. If all remaining categorization errors are resolved or all possible features f of erroneous categories c are tested then start a new training epoch. Otherwise we iterate the optimization Steps 1 and 2 to test further features and nodes.

4.3.4 Insertion Thresholds

Similar to the a priori definition of the optimal number of LVQ nodes also the insertion thresholds ϵ_{ltm}^1 for the feature testing or ϵ_{ltm}^2 for the node testing are difficult to predetermine. Large insertion thresholds minimize the number of allocated resources, but the learning progress is slow, which is unsuitable for our desired interactive learning capability. Additionally the learning approach may even fail to learn an appropriate representation for more difficult categories due to the fact that no feature candidate or node reaches the insertion threshold and therefore all of them are rejected. On the other hand, small insertion thresholds considerably increase the learning speed, because it is typically much easier to resolve small numbers of errors iteratively, but the amount of allocated network resources is much higher. Especially for the feature selection process this has the effect that also many irrelevant or object-specific features are selected, so that the generalization performance to new category instances is poor.

As a compromise between these two extremes we propose to start each learning epoch with high insertion thresholds ϵ_{ltm}^1 and ϵ_{ltm}^2, so that predominantly category-specific resources are allocated. During each iteration of the optimization loop illustrated in Fig. 4.5 a decrement of both thresholds is calculated based on the prespecified values ϵ_{ltm}^{max} and ϵ_{ltm}^{min}:

$$\Delta\epsilon_{ltm} = \frac{\alpha(\epsilon_{ltm}^{max} - \epsilon_{ltm}^{min})}{F} \tag{4.12}$$

to gradually relax the insertion constraint, where F corresponds to the total number of extracted features and α is a constant that controls the slope of the linear decrement. The final insertion thresholds of the current learning iteration are calculated in the following way:

$$\epsilon_{ltm} := \begin{cases} \epsilon_{ltm} - \Delta\epsilon_{ltm} & : \quad \text{if } \epsilon_{ltm} - \Delta\epsilon_{ltm} > \epsilon_{ltm}^{min} \\ \epsilon_{ltm}^{min} & : \quad \text{else} \end{cases} . \tag{4.13}$$

For the cLVQ architecture this gradual decrement of insertion thresholds has the benefit that at the beginning of a learning epoch many allocated object-specific network resources are rejected, but also allows the representation of categories for which no category-specific features are available. In such rare cases the categorization performance to new category members is most probably poor, but at least already known exemplars of such a category can be robustly detected. Furthermore all features that where initially below the insertion threshold ϵ_{ltm}^1 are retested if ϵ_{ltm}^1 meanwhile is below the previously measured performance increase.

4.4 Experimental Results

In the following section our proposed cLVQ life-long learning architecture is compared with a Single Layer Perceptron (SLP) and two modified cLVQ versions cGRLVQ and cLVQ*. The comparison of the exemplar-based networks is done to measure the effect of the feature weighting, and feature selection methods with respect to categorization performance, number of allocated resources and required training time. For this comparison the cLVQ* is the most simplified exemplar-based network, where nodes are incrementally added and tested, but no feature weighting and selection is performed. In contrast to this, the cGRLVQ additionally applies a feature weighting based on the GRLVQ method proposed by Hammer & Villmann (2002). The GRLVQ weighting is based on the distance d_c^{co} to the nearest correctly labeled prototype $\mathbf{w}^{k_{co}(c)}$ and d_c^{inco} to the nearest prototype $\mathbf{w}^{k_{inco}(c)}$ with incorrect label:

$$\Delta\lambda_{cf} = \Theta^{\lambda}\Phi'_G \left(\frac{d_c^{co}}{d_c^{co} + d_c^{inco}} (x_f^i - w_f^{inco})^2 - \frac{d_c^{inco}}{d_c^{co} + d_c^{inco}} (x_f^i - w_f^{co})^2 \right), \tag{4.14}$$

where Θ^{λ} is the learning rate for the λ_{cf} weighting values and Φ'_G is the first derivative of a Fermi-function. Although similar to the proposed cLVQ this dynamical feature weighting enables the cGRLVQ to suppress irrelevant features, but no explicit feature selection is performed.

The comparison with the SLP network architecture is done because this is the simplest neural network model that fulfills the requirements of the categorization task. Therefore SLPs are used to measure the baseline performance. For each category one output node is used. The output p_c^i of each node is defined as:

$$p_c^i = \frac{1}{1 + \exp(-\mathbf{w}_c * \mathbf{x}^i)}, \tag{4.15}$$

where $\mathbf{w}_c$ is a single linearly separating weight vector for each category c. The training is based on 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 Characteristics (ROC) curves, we estimate the rejection thresholds during the learning process, based on the average activation strength of the network output. This is necessary for interactive learning tasks to allow categorization of new object views at any time.

4.4.1 Experimental Setup

For the comparison of our cLVQ architecture with other learning approaches we use a challenging categorization database composed of 56 different training objects and 56 distinct objects for testing (see Fig. 4.7), which were never used during the training phase. For each object 300 color views of dimensionality 128x128 pixels were taken in front of a black background while rotating the object around the vertical axis.

Overall our object ensemble contains ten different shape categories and five different color categories as shown in Fig. 4.7. It should be mentioned that several objects are multi-colored (e.g. the cans) where not only the base color should be detected, but also all other prominent colors covering at least 30% of the visible object view. 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 all experiments performed with this database we trained the different network architectures with a limited and changing training ensemble

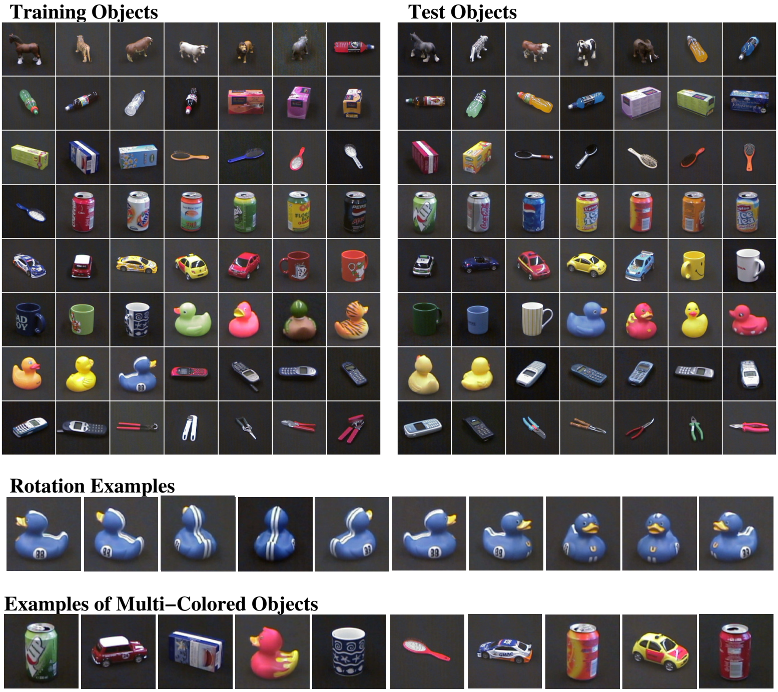


Figure 4.7: **Object Ensemble.** Examples of all training (left) and test objects (right) used for our categorization task, where 15 different categories are trained. As color categories red, green, blue, yellow and white are trained. The shape categories are animal, bottle, box, brush, can, car, cup, duck, phone and tool. Each object was presented in front of a black background and is rotated around the vertical axis (bottom), resulting in 300 color images per object.

composed of a visible "window" of only three objects to test the life-long learning ability of the different approaches. For each epoch only these three objects are visible to the learning algorithm. At the beginning of each epoch a randomly selected object is added, while the oldest one is removed. This scheme is repeated until all training objects are presented once to the network architectures. Additionally all experiments are repeated ten times with identical parameter set but random order of object presentation. The corresponding results shown in Fig. 4.9 and Fig. 4.10 are the average values over these runs.

4.4.2 Feature Extraction

The proposed cLVQ life-long learning method is a general categorization framework but we investigate the learning capabilities of our method based on a visual categorization task. Therefore different feature extraction methods are used to provide shape and color information as illustrated in Fig. 4.8. We propose to combine all these multiple feature cues, where for such learning tasks typically only a single feature extraction method is used (Mikolajczyk et al., 2006; Opelt et al., 2004; Fritz et al., 2007). Furthermore this qualitative separation of the extracted features is not given to the learning system as a priori information. For our categorization task we are particularly interested in discovering the structure of the categories 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 the category “fire engine” from other cars and trucks, 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 $\mathbf{x}^i$.

4.4.2.1 Extraction of Color Features

In contrast to the identification task in Chapter 3, we use for the considered visual categorization tasks color histograms, which combine robustness against view and scale changes with computational efficiency (Swain & Ballard, 1991). The histograms are used, because for the feature selection method a representation is required, where a single feature represents a specific color, which can not be achieved with the previously used down-sampled RGB-maps. Furthermore for our experimental setup the color histograms are commonly sufficient sparse, so that also the sparsity constraint of the proposed cLVQ feature scoring method is fulfilled.

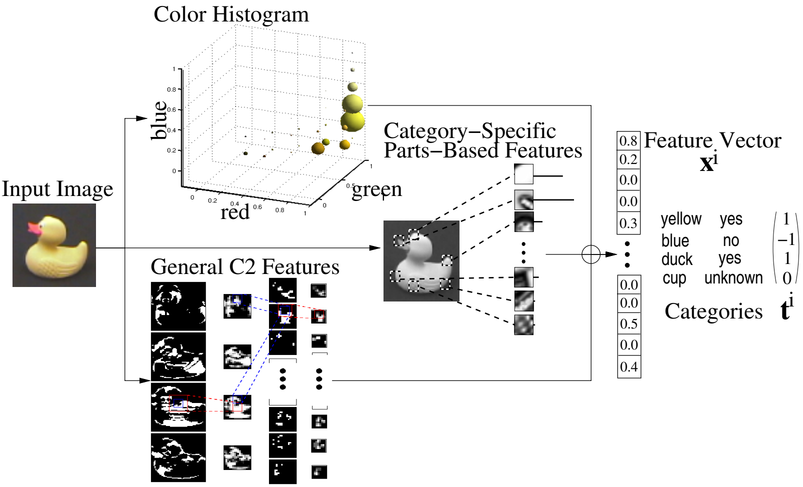


Figure 4.8: **Feature Extraction.** Color features are extracted as histogram bins in the RGB space. Shape features are obtained from parts-based feature detectors and a feed-forward feature extraction hierarchy. Shape and color features are concatenated into a single "flat" vector representation. Furthermore the target categories are represented in a category vector $\mathbf{t}^i$ for each feature vector $\mathbf{x}^i$.

4.4.2.2 Extraction of Shape Features

For the extraction of shape features two different methods are combined. The first method extracts category-specific parts-based features (see Hasler et al. (2007) or Appendix A.1.2 for details). This feature extraction is based on a learned set of category-specific feature detectors that are based on SIFT descriptors (Lowe 2004). Commonly these descriptors are only determined around some highly structured interest points, while the used feature extraction method applies them at all image positions. This especially allows the representation of structureless categories. For the final feature response only the maximum detector value is selected, so that all spatial information is neglected. The second method is the same hierarchical feed-forward feature extraction hierarchy (Wersing & Körner, 2003) as used for the object identification in Chapter 3. In contrast to the parts-based features the C2 features from this hierarchy are more general and less category-specific. Furthermore the C2 features are topographically organized, where a single feature responds to a local patch in the input image. 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 uncommon because most categorization methods rely on parts-features only (Willamowski et al., 2004; Agarwal et al., 2004).

4.4.3 Categorization Performance

Although no prior information is given during the learning process with respect to the kind of trained categories, we distinguish between color and shape categories in the performance measurement to discuss the different quality of extracted features and the corresponding behavior of all network architectures. We also investigate the effect of different shape features by performing experiments with parts-based features only or the combination of these features with less category-specific C2 features.

4.4.3.1 Color and Parts-based Features

The overall performance of the cLVQ architecture for this feature setting is good for all categories as can be seen on the left of Fig. 4.9. For the color categories it performs much better than the simpler cGRLVQ and cLVQ*. Thus for categories with a few stable and category-specific features a feature selection method and the suppression of irrelevant features is beneficial with respect to the generalization performance. On the contrary for shape categories the cGRLVQ method performs at intermediate training epochs better than cLVQ and cLVQ*, while at the end of the overall learning process it is only slightly better compared to cLVQ* and cLVQ. This slightly higher performance of GRLVQ and also cLVQ* compared to our cLVQ approach is most probably due to the much higher number of allocated nodes (see Fig. 4.10 for details). Although cLVQ is slightly worse for shape categories compared to the other tested vector quantization methods it is still able to capture most category information even for categories with higher appearance variations.

The SLP network architecture is distinctly worse for the color categories

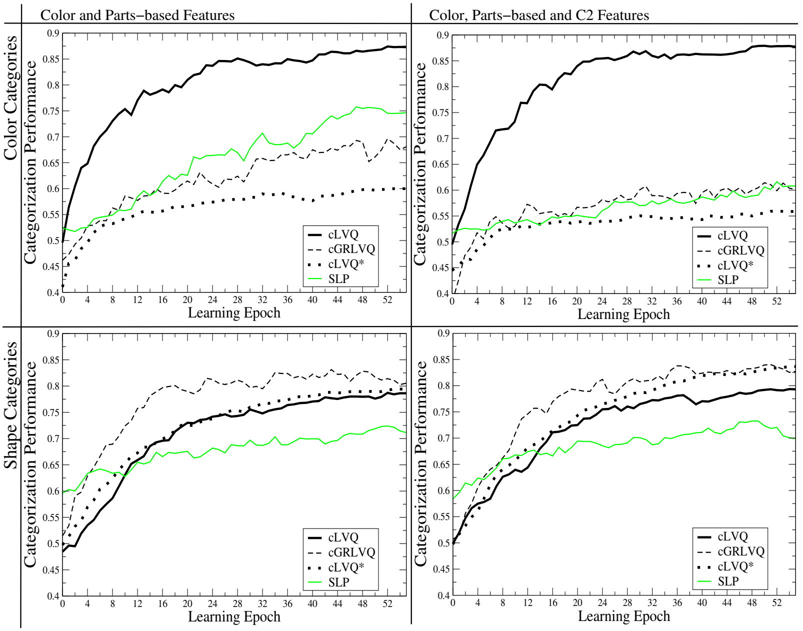


Figure 4.9: **Comparison of Categorization Performance for Color Categories.** For this comparison of our proposed cLVQ with cGRLVQ, cLVQ* and SLP we performed ten different runs with an identical parameter set but random object order. The categorization performance is calculated after each training epoch, based on all test objects. This means that the performance is calculated based on the representation of the objects seen so far, simulating an interactive learning session. Additionally different feature sets are used to investigate their impact on the categorization performance. In general the cLVQ method is superior for color categories compared to all other learning methods, while for the shape categories cLVQ is slightly worse than cGRLVQ and cLVQ*. Nevertheless the proposed cLVQ algorithm uses much fewer memory and computational resources as shown in Fig. 4.10.

than the proposed cLVQ method. For the combination of color and parts-based features SLP is able to suppress irrelevant features better than cGRLVQ and cLVQ*. Therefore the SLP achieves a distinctly higher performance among all tested methods using every feature. For the shape categories the SLP network architecture is only superior at earlier learning epochs, but is worse if the learning process is continued.

Overall the SLP performance is surprisingly good, which is in contrast to classification tasks with a one-out-of-n class selection, where the SLP approach is known for the "catastrophic forgetting effect" (French, 1999; Kirstein et al., 2008). For our categorization task this effect is only slightly visible for the shape categories, where the performance increase for newly presented objects is distinctly less than for all other tested approaches. We therefore would expect that the performance difference between cLVQ and SLP increases if the learning process is continued.

4.4.3.2 Color, Parts-based and C2 Features

We also performed experiments with color histogram features, parts-based features and hierarchical C2 features. It can be seen on the right side of Fig. 4.9 that the cLVQ method reaches almost the same performance as in the previous feature setting. In contrast to this the performance of cGRLVQ, cLVQ* and also the SLP is distinctly worse for the color categories compared to the feature set using only color and parts-based features. Additionally for color categories the SLP is not better anymore than cGRLVQ, so that the performance difference to cLVQ is nearly 30%. For the shape categories the cGRLVQ architecture achieves a better performance in comparison to cLVQ even for the final learning stages. Overall it can be said that for color categories our proposed cLVQ is unaffected if the general but less category-specific C2 features are added, but these features only have a minor positive effect on the shape categories. Nevertheless we believe that C2 features can become beneficial by contributing to the fine tuning of the category representation if the learning process will be continued.

4.4.4 Comparison of Required Network Resources

In the following we compare the different learning approaches with respect to the required network resources (see Fig. 4.10). For interactive learning tasks the training time is most crucial. For the used vector quantization approaches cLVQ, cGRLVQ and cLVQ* this training time is basically determined by the overall feature dimensionality and the capability to iteratively solve remaining errors by allocating new prototype nodes. Furthermore the number of used features and the total number of these allocated nodes are important with respect to the learning speed.

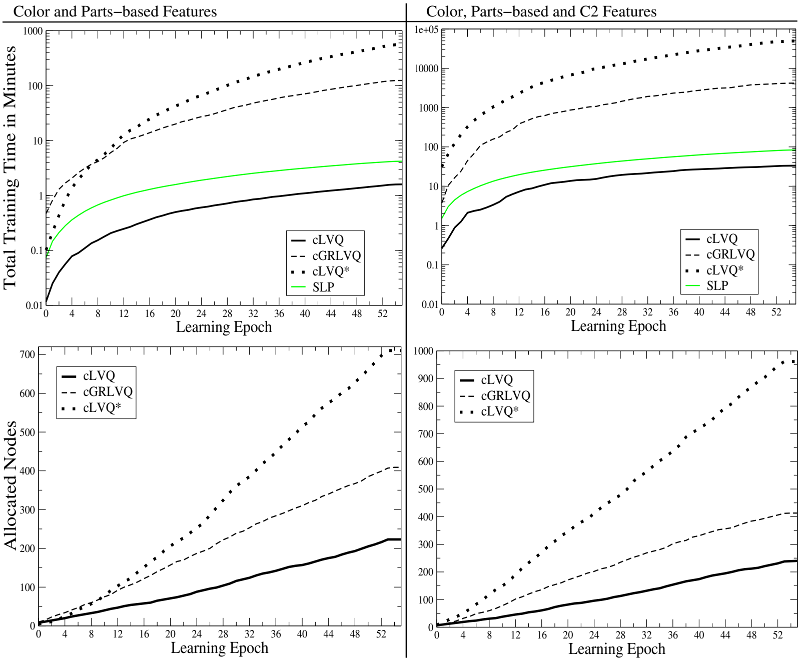


Figure 4.10: **Comparison of Network Resources.** The total training time of the proposed cLVQ and also cGRLVQ, cLVQ* and SLP is most crucial with respect to interactive learning. Therefore for both feature sets and all tested approaches the corresponding training time is shown at the top of this figure. It can be seen that especially in later learning epochs the simpler vector quantization methods cGRLVQ and cLVQ* require more than two orders of magnitudes more training time compared to SLP and cLVQ, while the cLVQ is even two times faster than the SLP. This computational efficiency of the cLVQ method is caused by the small number of selected features, but also by the smaller number of allocated LVQ nodes as shown at the bottom of this figure.

In contrast to this the training time of the SLP network architecture only depends on the feature vector dimensionality and the number of preselected iterations per training epoch (100 in our case). Additionally we are also interested in the scalability of the different approaches with respect to an increasing feature dimensionality.

The proposed cLVQ method theoretically has the highest computational costs if almost all feature dimensions were used, because of the iterative insertion and testing of features and nodes. Effectively it is more than two orders of magnitudes faster compared to cGRLVQ and cLVQ*. For the combination of all three feature extraction methods the proposed iLVQ requires about 30 min training time to acquire the category representation, while the iGRLVQ already need about three days for the same task. Compared to this, the simpler cLVQ* requires overall five weeks training time for this learning task, which strongly highlights the necessity for a dynamic feature weighting. In our experiments the proposed cLVQ is even more than two times faster than the simple SLP network architecture as shown in the upper part of Fig. 4.10. This computational efficiency is basically caused by the proposed feature selection method, which typically selects less than 5% of all available feature dimensions. Also the number of allocated neurons is much smaller compared to the other vector quantization methods as shown at the bottom of Fig. 4.10. This is another positive side effect of this small amount of selected category-specific and stable features. This smaller number of nodes again enhances the learning speed and is also beneficial with respect to the representational capacity for storing many different categories.

4.4.5 Qualitative Evaluation of the cLVQ Feature Selection Method

Apart from categorization performance and network resources we are also interested in how good the feature selection method of our proposed cLVQ learning algorithm is able to find reasonable category-specific features. Therefore ten different training runs of the cLVQ method were performed and all selected features for each category are saved together with the corresponding feature scoring values. The selected features for each category c are sorted based on the total number of occurrence in these ten runs, where frequent features are most probably critical for the representation of this particular category. Additionally each feature is visualized with a small patch, to allow a visual inspection of its usefulness for the corresponding category. We use the RGB value of the histogram bin center for color features, while for the parts-based features the grey-value patch corresponding to the highest detector activity is chosen (Hasler et al., 2007). We also consider the final scoring value h_{cf} of each selected feature. This value is identical for all learning runs

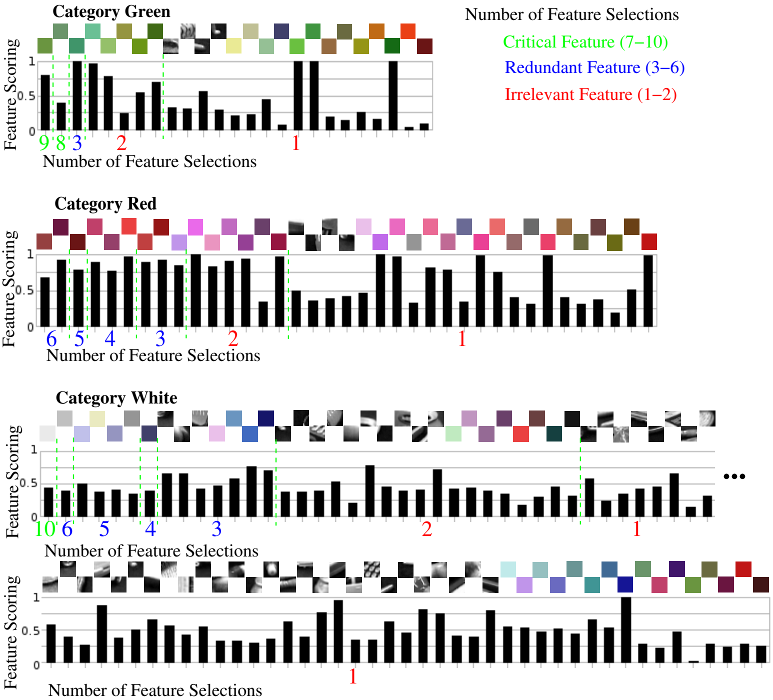


Figure 4.11: **Qualitative Evaluation of the Feature Selection Method for Color Categories.** Illustration of the selected features of three representative color categories, where one easy, one average and one difficult category was selected. For this visualization ten different cLVQ networks are trained and the selected features of each category together with the scoring values are saved. The selected features of each category are sorted based on the total number of occurrences in these ten runs, while the bar height correspond to the feature score of these selected features. All features that occurred at least 7 times are considered as critical for the representation of this category, while feature occurrence of less than 3 time are probably irrelevant or even wrong. Finally features that are selected 3-6 times indicate redundant feature sets, with similar representational capacity. Besides the occurrence of each feature the total number of selected features indicate the difficulty of the category. This is especially visible for the worst color category "white". Nevertheless even for this category the correct color feature is selected in all test runs.

and provides information about the category specificity of this feature. The results of this investigation are shown in Fig. 4.11 for three representative color categories and in Fig. 4.12 for three shape categories.

Due to the fact that the training objects are presented iteratively to the cLVQ, its wrapper feature selection method can never be perfect. This is because a certain feature at a particular learning state might be useful, but with more experience it can become obsolete. This especially occurs for the first object presentation of a shape category, where often a color feature is selected, because due to the object rotation it is more stable than all shape features. As a consequence, features that are selected only once in Fig. 4.11 and Fig. 4.12 are most probably not category-specific and in many cases unrelated to the most exemplars of the category. But such erroneous features often also have low scoring values, so that the impact of these features for the category representation is minimized. Interestingly the number of features selected once and also their total number positively correlates with the categorization performance. Therefore both numbers indicate the difficulty of each category. Furthermore the categorization performance over different runs is more stable if the set of different selected features is small. In contrast to this a larger number of selected features which occurred 3-6 times during the different runs, indicates that several redundant feature sets with roughly the same representational power exist.

It is somehow surprising with respect to the difficulty of categories that the color categories are not in general easier compared to shape categories. This is especially visible for the category “white” shown in Fig. 4.11 and the category “cup” illustrated in Fig. 4.12. Although in all runs the correct histogram bin for white was selected, the corresponding scoring value of this feature is quite low. This small scoring value is most probably caused by reflections on glossy objects, because such spots typically cause activations of this histogram bin that are independent of the actual color of the object. Additionally “white” is the only color category for which only few training objects are completely white but many of them contain smaller fractions of white. Therefore for this category the separation from other co-occurring shape and color categories becomes more difficult. Finally it should be mentioned that among the most frequently reoccurring features a considerable amount have relatively small scoring values, even if some features with higher scoring values are available. This effect is best visible for the category “animal” in Fig. 4.12. Basically this can occur if features with higher scoring values are rarely activated and thus are rejected because

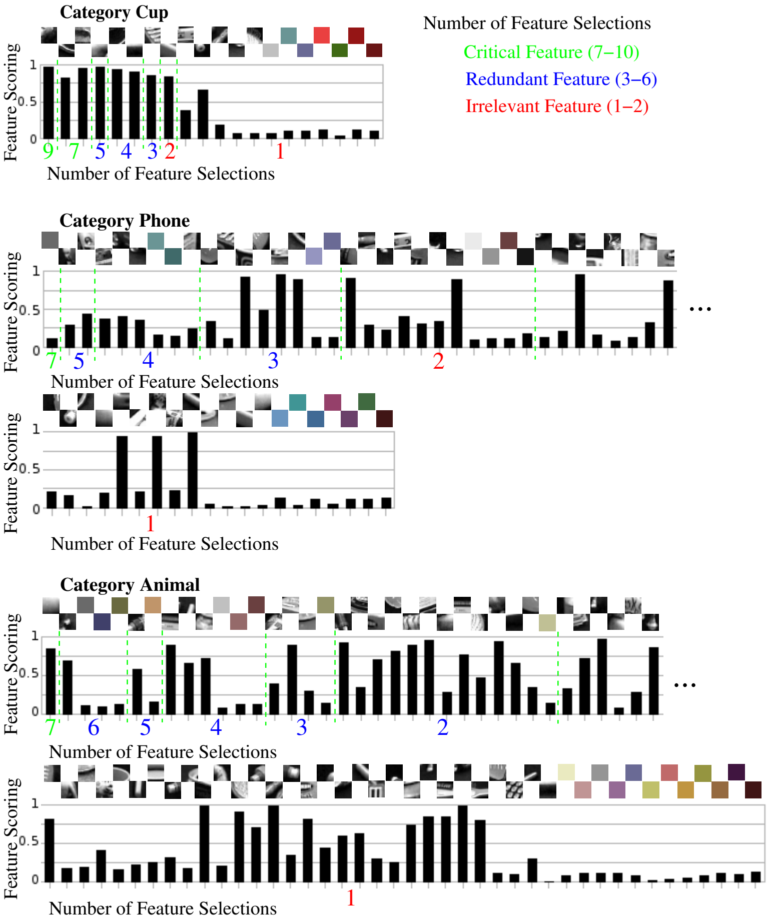


Figure 4.12: **Qualitative Evaluation of the Feature Selection Method for Shape Categories.** Illustration of the selected features of three representative shape categories, analogous to Fig. 4.11. Surprisingly not all shape categories are more difficult than the color categories. This becomes clear if one compares the category "cup" with "white" depicted in Fig. 4.11. Furthermore many stable selected features have low scoring values, which indicate only little category specificity. This suggests that for shape categories only the combination of several features allows a stable category representation.

the measured performance gain is below the feature insertion threshold. Additionally it is probable that at least for the shape categories the combination of several features is important, so that a single feature might be general and less category-specific, but in combination with other features allows a robust category detection.

4.5 Discussion

We have proposed an architecture for interactive life-long learning of arbitrary categories that is able to perform an incremental allocation of cLVQ nodes, automatic feature selection and feature weighting. This automatic control of the architecture complexity is crucial for interactive and life-long learning, where an exhaustive parameter search is not feasible. Additionally we use our proposed wrapper method for incremental feature selection, because the representation of categories should use as few feature dimensions as possible. This can not be achieved with simple filter methods, where typically only a small amount of redundant or noisy features are eliminated. The used feature selection method enables the cLVQ to separate co-occurring categories and allows a resource efficient representation of categories, which is beneficial for fast interactive and incremental learning of categories. Recently a variant of an embedded feature selection method for LVQ networks was proposed by Kietzmann et al. (2008) based on the GRLVQ method (Hammer & Villmann, 2002) which was called iGRLVQ. This method iteratively removes features with small weighting values λ. For our categorization task this proposed backward feature selection method is not suitable because a low λ value at a certain learning epoch does not imply that this feature can not become useful at a later learning stage. Unfortunately a removed feature can not be readded to the corresponding iGRLVQ network at a later learning stage, especially if the reduction of computational costs is targeted. Additionally the definition of a stopping condition for the feature pruning is difficult to determine a priori, so that Kietzmann et al. (2008) prespecified the final feature dimensionality. Finally the required computational costs are considerably higher compared to forward feature selection methods.

In contrast to many other categorization approaches our model is able to learn multiple categories at once, while commonly the categories are trained individually (Fritz et al., 2005; Fei-Fei et al., 2007). We applied our learning method to a challenging categorization task, where the objects are rotated around the vertical axis. This rotation causes much higher appearance changes compared to many other approaches dealing with canonical views only (Leibe et al., 2004). In contrast to this our exemplar-based method can deal with a larger within-category variation, which we consider crucial for complex categories. Furthermore we recently could show that our proposed cLVQ learning method can be integrated into a larger vision system that allows online learning of categories based on hand-held and complex-shaped objects under full rotation (Kirstein et al., 2008a). This means our cLVQ approach does not only scale well to higher feature dimensionalities, but also to more complex categorization tasks in unconstrained environments.

Chapter 5

Interactive and Life-Long Learning in Unconstrained Environments

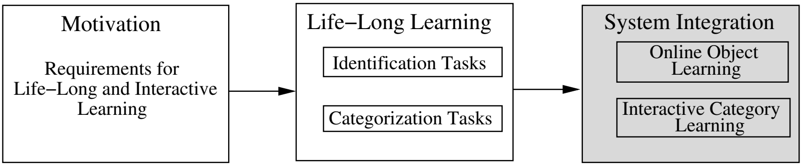


Figure 5.1: **System Integration.** In the following chapter we propose a modular active vision system that utilizes the previously developed life-long learning approaches. This integrated system enables interactive learning for identification and categorization tasks based on complex-shaped objects held in hand. We first explain the different building blocks of this system, before we show the applicability of the different approaches for challenging unconstrained experimental setups.

Life-long learning is one necessary precondition for an assistive system, acting autonomously in an unconstrained environment. Such a system must be able to continuously update and increase its knowledge to fulfill useful tasks in scenarios like an office environment or the flat of its owner. The learning capability is basically required because a pre-training of all necessary objects and categories it may require for its future tasks is unfeasible due to the richness and complexity of natural environments. Therefore the following chapter considers how the proposed learning methods can be integrated into a larger vision system to allow interactive learning of hand-held objects in cluttered office environments. Unfortunately, the iLVQ method proposed as LTM model for object identification tasks is too slow for interactive learning, therefore in the following only the STM model and the cLVQ LTM model for categorization tasks is considered. Furthermore we experimentally show the difficulty of interactive learning in this unconstrained experimental setup. We therefore compare the recognition performance of this setup with a more restricted setting using a black background as used in Chapter 3 and Chapter 4. For this comparison the same set of training and test objects for both experimental setups was used.

The proposed online vector quantization (oVQ) was the first methods that was integrated into different interactive learning systems. The starting point was a simple learning framework that was composed of an entropy-based segmentation, a feature extraction hierarchy and the proposed online vector quantization (oVQ) method (Kirstein 2004). This system already provides the capability of online learning of object-specific STM representations. Nevertheless the experimental setting at this stage was strongly restricted, because objects could only be learned using a black glove and a black background. As a next development stage we integrated the online learning approach into a brain-like active vision system. This system allowed based on a more sophisticated figure-ground segregation (Steil & Wersing 2006) learning of objects held in hand (Wersing, Kirstein, Götting, Brandl, Dunn, Mikhailova, Goerick, Steil, Ritter, & Körner 2006; Wersing, Kirstein, Götting, Brandl, Dunn, Mikhailova, Goerick, Steil, Ritter, & Körner 2007a). For this system we also proposed the sensory-memory concept (see Section 5.1.4.1) and the user interaction with temporal integration (see Section 5.1.5) to enhance the identification performance of this more challenging learning scenario. Furthermore we developed a system that enables learning of complex-shaped objects and faces into a single STM representation (Wersing, Kirstein, Götting, Brandl, Dunn, Mikhailova, Goerick, Steil, Ritter, & Körner 2007b). The proposed oVQ method was also utilized for the humanoid robot ASIMO, where interactive online learning was coupled with autonomous behavior generation (Goerick, Mikhailova, Wersing, & Kirstein 2006; Goerick, Bolder, Janssen, Gienger, Sugiura, Dunn, Mikhailova, Rodemann, Wersing, & Kirstein 2007). Finally also the category Learning Vector Quantization (cLVQ) was integrated into an active vision system enabling interactive learning of visual categories based on natural and arbitrary rotated objects (Kirstein et al., 2008a; Kirstein et al., 2009).

In the following chapter the fundamental building blocks required for interactive learning of visual representations are summarized in Section 5.1. The combination of these building blocks with our proposed learning methods enables interactive learning based on complex-shaped objects presented by hand in unconstrained office environments. Additionally several offline and interactive learning experiments are performed in Section 5.2 and Section 5.3 to show the corresponding learning capabilities of the proposed vector quantization approaches under this more challenging experimental setup.

5.1 An Integrated Vision Architecture for Identification or Categorization Tasks

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 it 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 the human body. We utilize this concept as an attention mechanism, where everything in this short distance range is of particular interest to the system with respect to interaction and learning. To allow interactive learning of arbitrary objects and visual categories in a cluttered office environment this near range depth information and an additional foreground-background segregation method are used to isolate the object from the scene. Based on the defined foreground region robust color and shape features are extracted, which are used as input for the corresponding learning module. Finally an user interaction is required to allow a natural communication with the human tutor. All this processing steps are illustrated in Fig. 5.2 and are described in more detail in the following.

5.1.1 Object Hypothesis Generation

For the generation of the initial object hypothesis 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 cor-

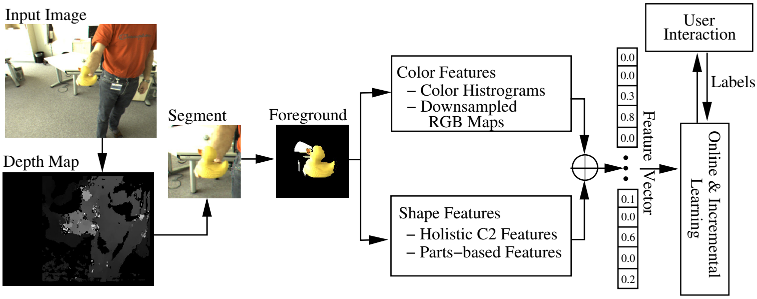


Figure 5.2: **Building Blocks of the Interactive Learning System.** Based on an object hypothesis extracted from the depth map a figure-ground segregation is performed. The detected foreground is used to extract color and shape features to represent objects or visual categories. All extracted features are concatenated into a single structureless vector $\mathbf{x}^i$. This feature vector together with the label information provided by a human tutor, is the input to the incremental and interactive learning module. Furthermore also the recognition results are communicated to the tutor based on a state-based user interaction.

rection of lens distortions. This depth information is used to generate an interaction hypothesis in cluttered scenes that 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.

5.1.2 Figure-ground Segregation

The extracted segment $\mathbf{J}^i$ contains the object view, but also a substantial amount of background clutter as can be seen in Fig. 5.3. For the incremental build-up of visual 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. 5.3 is to train for each segment $\mathbf{J}^i$ a Learning

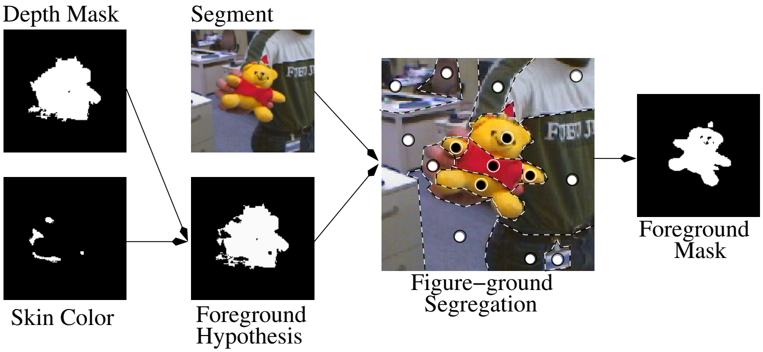


Figure 5.3: **Figure-ground Segregation.** Based on the extracted segment, the corresponding depth mask and a skin color removal, a foreground hypothesis is generated. This hypothesis includes a considerable amount of noise and clutter, which the applied figure-ground segregation method strongly reduces. 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.

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 edges of the corresponding object silhouette. Furthermore also "holes" at textureless object parts are common. Due to the fact that the objects are presented manually, 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). Therefore faces and gestures can not be learned with this preprocessing. Nevertheless with a modified preprocessing as proposed in Wersing et al. (2007a) a combined learning of objects and faces can be achieved. For the figure-ground segregation the learning of the representing subsegments 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 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 purpose of figure-ground segregation such local matrices lead to a significantly better foreground classification (Denecke et al., 2009), which directly enhances the interactive learning process. 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.

5.1.3 Feature Extraction

For the learning of objects and visual categories we use different feature extraction methods providing shape and color information. All extracted features of the corresponding object view are concatenated into a structureless feature vector $\mathbf{x}^i$, where especially for the category learning we do not give this qualitative separation of color or shape features to the learning system as a priori information. This is done, because in general such a priori knowledge (e.g. red is a color category) given to the different learning methods should be minimal to make as few assumptions for the learning as possible. Additionally the learning system should make efficient use of all features, which partly also includes the combination of color and shape features for our object identification and categorization tasks. Finally we use for all experiments in this chapter the same feature sets as described in Chapter 3 and Chapter 4 to allow a better comparison of the different experimental setups.

5.1.3.1 Feature Extraction Methods for Object Identification Tasks

For all object identification tasks C2 features (Wersing & Körner 2003) obtained with a hierarchical feed-forward feature extraction architecture are used to provide high-dimensional but sparse shape features. Additionally coarse color features are used, which are based on downscaled RGB-maps.

5.1.3.2 Feature Extraction Methods for Visual Categorization Tasks

For the representation of shape features we combine the C2 features (Wersing & Körner, 2003) with a parts-based features extraction method (Hasler et al., 2007; Hasler et al., 2009). The feature detectors of the hierarchical feature extraction architecture are obtained by unsupervised learning, resulting in general and less category-specific features, while the parts-based feature extraction methods are trained supervised 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. Additionally the histogram binning method (Swain & Ballard, 1991) is used to provide color information.

5.1.4 Learning of Object and Category Representations

Inspired by the human brain we use different memory concepts (see Fig. 5.4) to interactively learn object and category representations. These memory concepts are basically used to combine the online learning capability of the oVQ approach with iterative learning methods that require more computational resources but are able to extract more resource-efficient representations. Therefore labeled training vectors are first stored into an intermediate and object-specific sensory and short-term memory (STM) representation and are finally transferred into a category-specific long-term memory (LTM) representation. In the following the basic concepts and the used learning methods are explained in more detail.

5.1.4.1 Sensory Memory Concept

For interactive learning scenarios usually only very few object views are seen by the system. Additionally learning systems are typically strictly separated into a train and test phase, where commonly two distinctive sets of views are used. To relax this separation and to make the most efficient use of object views, we introduce a sensory memory concept (Wersing et al., 2006) for temporarily remembering views of the currently attended object, by using the online vector quantization (oVQ) method (Kirstein et al., 2005a). The basic assumption behind

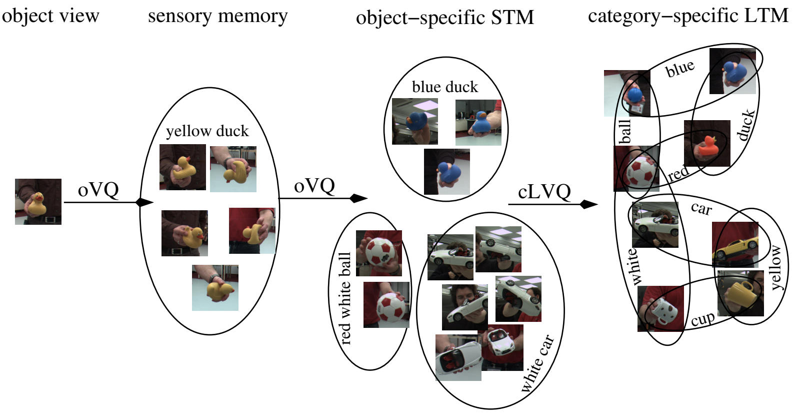


Figure 5.4: **Memory Concepts for Interactive Learning.** Object views are first buffered into the sensory memory until label information is provided by the tutor. Due to our assumption that only views of a single object are represented into this memory, all collected views have the same label information, even if they are collected before the labeling. After labeling this knowledge is transferred into the STM using the same oVQ learning method as for the sensory memory. The object-specific STM is limited in capacity allowing only to store few different objects. Additionally to the STM we can apply the life-long learning cLVQ method to approach the “stability-plasticity dilemma” and iteratively transfer the STM information into the category-specific LTM.

this memory concept (see Fig. 5.4) is, that only views of a single object are inserted and that the memory is cleared if the object identity changes. Therefore every time the attended object disappears from the peri-personal space this memory is erased. The advantage of this memory concept is that object views first can be used to test the STM and LTM representation. After providing confirmed labels the same views can also be used to enhance both memory representations, even if these object views were recorded before the confirmation.

5.1.4.2 Short-Term Memory Concept

For the transfer from the sensory memory to the STM representation the same oVQ model is used, which provides fast appearance-based learning of several complex-shaped objects. The proposed STM model stores template-based representatives $\mathbf{r}^l$ with $l = 1, \ldots, L$. The representatives

$\mathbf{r}^l$ are providing a limited and changing object-specific memory by applying a kind of novelty detection. Compared to the naive approach, where each $\mathbf{x}^i$ is stored as representative $\mathbf{r}^l$, we could show that the number of representatives $\mathbf{r}^l$ can be considerable reduction by about 30% without losing generalization performance (Kirstein, Wersing, & Körner 2008).

5.1.4.3 Long-Term Memory Concept for Category Learning

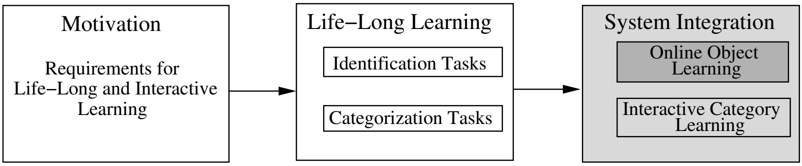
For the knowledge transfer from the object-specific STM to the category-specific LTM the cLVQ learning method (Kirstein et al., 2008b) is used that combines an incremental exemplar-based network and a forward feature selection method. The proposed cLVQ allows life-long learning and also enables a separation of co-occurring visual categories, which most exemplar-based networks can not efficiently handle. Both parts are optimized together to ensure a compact category representation that is necessary for fast and interactive learning.

The exemplar-based network part of the cLVQ method is used to approach the "stability-plasticity dilemma" of life-long learning problems. It basically represents the variations of different category members (e.g. normal car and cabriolet) and object poses (e.g. front and side view of cars). The included forward feature selection method is used to find low dimensional subsets of category-specific features. For the category decision only these selected feature sets together with the LVQ nodes are considered, which make the cLVQ method computationally efficient. To achieve a high generalization performance the selection process should predominately select features, which occur almost exclusively for a certain category. For guiding this process a feature scoring value h_{cf} for each category c and feature f is calculated. Similar to the development of category concepts of children, this scoring is only based on previously seen exemplars of a certain category and can strongly change if further information is encountered. Therefore the h_{cf} values are continuously updated.

5.1.5 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. To also allow the labeling of objects and categories for which no specific label is defined, additionally some wild card labels (e.g. “property one”) are included in this list. All these 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 recognition decisions of detected object identities or categories over 5 seconds ($\approx$ 20-30 segments). This temporal integration of recognition results is beneficial, because it strongly reduces errors which typically occur in the challenging task of object identification or categorization based on hand-held objects in cluttered scenes. These individual decisions are used to generate a hypothesis of the currently present categories or the shown object. It is communicated to the interacting person, where additionally also a confidence value (sure, maybe, unknown) is added based on predefined rejection thresholds. The detected classes are repeatedly communicated to the user (in 5 second intervals), while newly acquired segments are also used to refine the recognition outcome. 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 object or category representation. Furthermore it is also possible for the user to directly provide class labels, in order to label previously unknown objects and categories.

5.2 Object Identification in Unconstrained Environments



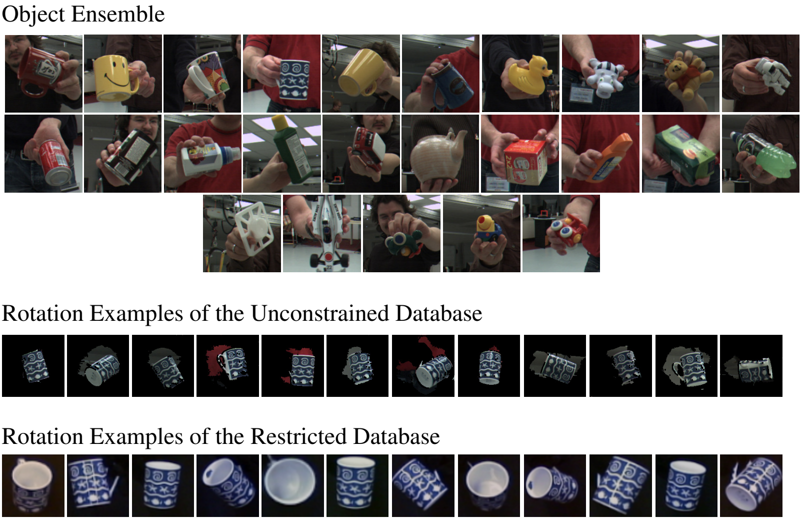
In the following section we show the learning capability of our proposed online vector quantization (oVQ) method for several offline and interactive learning experiments. For the interactive learning experiment complex-shaped objects are freely rotated by hand in front of our active camera system. Based on the extracted segment and the corresponding foreground mask, 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 class label o are used to interactively learn natural objects under real-world conditions. For the offline experiments two databases with the same objects but different experimental setups are used. The major difference between the offline and interactive learning experiments is that for the offline experiments no sensory memory is required.

5.2.1 Offline Object Identification Experiments

For the offline experiments we investigate the identification performance of our proposed online vector quantization (oVQ) approach for three different feature sets. As the simplest feature representation the original RGB color images are used. This feature set is compared with hierarchical C2 features, while for the third feature representation C2 and coarse color features are combined. Furthermore we analyze the effect of temporal integration that is defined as a majority voting schema. This means the classifier responses are accumulated over a predefined number of input vectors (20 in these experiments), where the most occurred class in this list is assigned to the current object view. This enhances the identification performance if the total number of misclassifications is considerably low.

5.2.1.1 Experimental Setup

The experiments in this sections are based on two different databases, where the same training and test objects, as shown in Fig. 5.5, are collected using different experimental setups. Overall 25 natural and complex-shaped objects are used for both databases. The objects of the first database are presented by hand using a black glove. Additionally the objects are freely rotated in front of a black background, so that no figure-ground segregation is required. Based on this experimental setup we collected 600 views of each object. Due to the presentation of objects in front of a black background we refer to this database in the following as *restricted database*. In contrast to this for the *unconstrained database* the same objects are presented held in hand in a cluttered environment. Although the number of training views and also the object presentation



Object Ensemble

Rotation Examples of the Unconstrained Database

Rotation Examples of the Restricted Database

Figure 5.5: **Identification Database.** Illustration of the 25 objects used for the offline identification experiments. For the *unconstrained database* the objects views are acquired in a cluttered office environment and are presented held in hand. In contrast to this for the *restricted database* the same objects are presented in front of a black background. For both databases the objects are freely rotated in depth as shown in the rotation examples. Furthermore the number of collected segments and the object size (for the *unconstrained database* larger segments are extracted) are nearly identical.

is nearly identical to the previous database, the *unconstrained database* is more challenging. This is due to larger brightness variations that are caused by the active movement of the camera head. Furthermore fluctuation and errors in the foreground mask increase the difficulty of this database. All these effects cause additional fluctuations in the feature responses, so that the learning of the object representations is more complicated.

Furthermore experiments based on the *object and face database* are performed (see Fig. 5.7). These experiments were done to show the generality of the used feature representation and the proposed oVQ method to different visual identification tasks. Therefore objects and faces are sharing the same feature representation and are collected into a single short-term memory. Similar to the *unconstrained database*, the 15 objects of this database are presented by hand in a cluttered office environment. For the preprocessing and figure-ground segregation the mechanisms explained in Section 5.1.1 and Section 5.1.2 are used. In contrast to this for the views of 10 different persons we combine depth and skin color information to generate the foreground hypothesis. Additionally, we proposed a saliency mechanism to dynamically switch between the identification of objects and faces (Wersing et al., 2007b). Overall 100 training and 100 distinct test views are collected for each of the 25 classes.

5.2.1.2 Object Identification Results

For this comparison we investigate the tradeoff between the representation accuracy and the identification performance. Therefore the similarity threshold ϵ_{stm} of our proposed oVQ method was varied from 0.5 to 0.95, where each symbol in Fig. 5.6 corresponds to a particular ϵ_{stm} value. The selected ϵ_{stm} influences how many representatives $\mathbf{r}^l$ are selected by the oVQ method. Based on these $\mathbf{r}^l$ the corresponding classification rate is calculated. Similar to the online learning experiments in Chapter 3.3.3 the usage of the original images leads to the worst identification performance, while for the combination of C2 shape features and coarse color the best results are achieved. This is consistent for all similarity thresholds and both databases.

For this investigation both identification databases use the same objects. Additionally also the total number of training views, the overall object size in the segment and the 3D in-depth rotation is almost identical for both object ensembles. Nevertheless the identification performance of the *unconstrained database* is distinctly worse. The fundamental difference between both image ensembles is the strongly varying background in the *unconstrained database* as shown in Fig. 5.5. Therefore a figure-ground segregation is essential to enable efficient online learning with the proposed oVQ, because the one-shot learning method directly stores the feature vectors as representatives. Otherwise the background would cause strong additional fluctuations in the feature responses, so that only very poor generalization can be achieved. The used figure-ground segregation method already suppresses large portions of the strongly varying background, but considerable parts of this image proportion are still included in the foreground masks. Additionally sometimes parts of

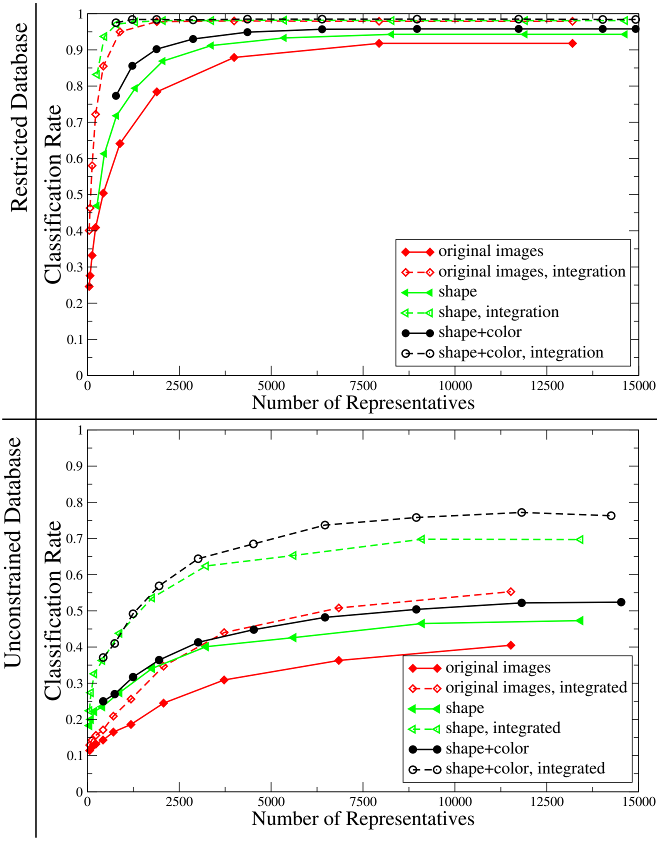


Figure 5.6: **Comparison of the Identification Experiments.** We compare the identification performance based on three different feature sets. Based on the given feature set and the similarity threshold ϵ_{stm} the oVQ automatically selects a set of representatives that are used for the evaluation with the test views. Furthermore we investigate the effect of temporal integration with respect to the overall performance. Comparing the results with a similar number of selected representatives commonly the original images perform worse, while the combination of C2 features with coarse color features performs best. Additionally temporal integration enhances the identification performance for all considered test cases. Finally it can be seen that the identification for the unconstrained database is distinctly worse although the same objects as for the restricted database are used.

the object views are missing in this mask. Both effects together with stronger brightness variations most probably cause the distinctly worse performance for the *unconstrained database.*

Furthermore we investigated the effect of the proposed temporal integration with respect to the identification performance. If this mechanism is applied for the *restricted database* nearly perfect identification performance can be achieved independent of the selected feature representation. This high performance is furthermore also achieved for a large variety of similarity thresholds. Although the utilization of temporal integration for the *unconstrained database* leads to a considerable performance increase (up to 20%) the performance gain is largely determined by the raw classifier performance. Therefore good generalization for this difficult database can only be reached if the object views are densely sampled, so that the stronger variations in the feature responses are captured.

5.2.1.3 Object and Face Identification Results

Commonly for identification problems a priori knowledge is utilized in the feature representation and learning approach. This guarantees for this particular learning problem optimal performance, but the applicability of the learning system to other tasks is strongly reduced. In contrast to this for the proposed learning systems as little prior-knowledge as possible should be used. To experimentally show the generality of the learning system the oVQ is utilized for a unified representation of objects and faces. This means the different classes are collected using the same feature and STM representation.

Overall also for this learning problem good classification results are reached as can be seen in Fig. 5.7. Nevertheless the performance gain with respect to the temporal integration of the classifier responses is much weaker compared to the *unconstrained database.* This is basically related to the coarse sampling of the different classes, where only 100 training views are used. Therefore large portions of the viewing sphere of the presented classes is missing in the training ensemble, so that many test views connected to such areas are consistently wrong classified.

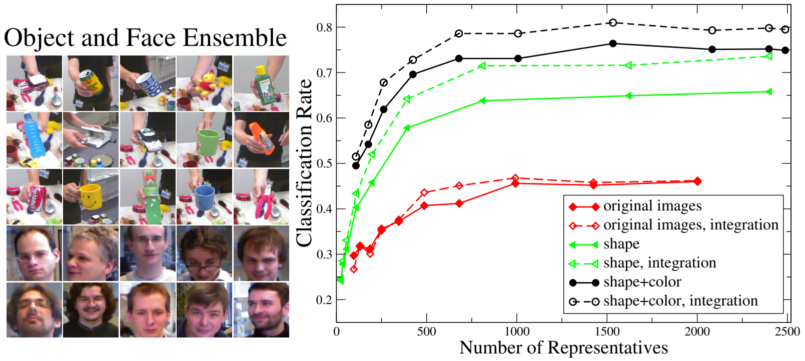


Figure 5.7: **Combined Object and Face Identification.** For the combined online learning of objects and faces 25 different classes are used. The different objects of this database are similar to the *unconstrained database* freely rotated in front of a cluttered background. In contrast to this the different faces are basically collected in frontal poses. Similar to the experiments shown in Fig. 5.6 a high identification performance can be reached if the temporal integration is applied. The only difference to earlier experiments is that for this experimental condition the performance gain is distinctly weaker. This effect is basically caused by the low number of training views (100 per class), so that considerable parts of the viewing sphere are not present in the training set. Therefore test views related to such viewing regions are consistently classified wrongly.

5.2.2 Interactive Learning of Object Representations

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 to 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 three 3 GHz CPUs. The overall system including preprocessing, figure-ground segregation, feature extraction, online 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 active vision system.

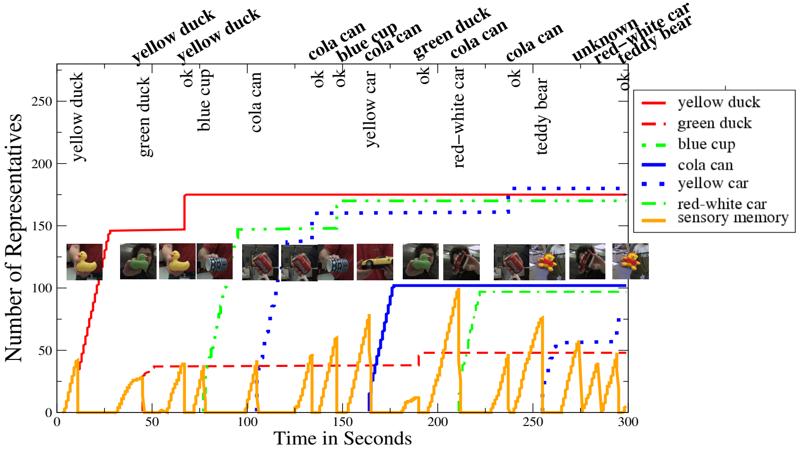
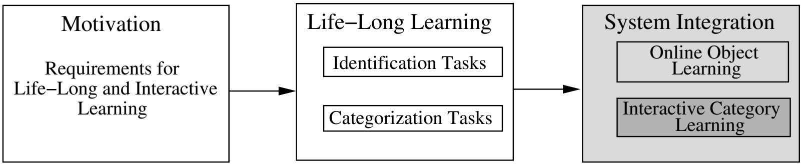


Figure 5.8: **Incremental Learning of Objects.** The oVQ model starts from a completely blank memory and incrementally learns seven different objects in approximately five minutes. The objects are selected in a way that they are similar in color or in shape. To show the ability of the proposed method to recognize and refine already known objects each of the trained objects are presented to the system immediately after the first presentation or later in this learning session. Furthermore we plotted the allocated representatives for the sensory memory and each object over time. It can be seen that the number of representatives for each object is different, depending on the object complexity and on how strong it was rotated. We also added the identification 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 communicated decisions on top of the figure and that not every time a confirmation is given.

In Fig. 5.8 a typical online learning session is shown that is trained based on complex-shaped objects. We claimed in Chapter 3 that the proposed oVQ can online train, refine and recognize many natural objects independent of the presentation order. This means that a new object can be immediately identified after the first presentation, which we shown in Fig. 5.8 for the cola can. Furthermore for the considered learning session each object is at least presented once again to highlight the identification capability of the learned representation, but also to show that the presented objects can easily be refined. Although in the presented learning session only seven different classes are trained in about five minutes, we selected the objects in a way that they are similar in color or in shape. This emphasize that also visually similar objects can be easily separated with the proposed oVQ approach. We especially want to mention that this identification capability is achieved based on freely rotated objects held in hand. Furthermore the average training time per object is approximately only 30 seconds. Finally we want to note that the proposed sensory memory is very useful to make the most efficient use of the presented object views, because up to 100 representatives are collected until confirmed label information is provided by the tutor. Compared to the total number of representatives of each object in the STM (< 200 representatives) this refers to a considerable amount of the learned object representation.

5.3 Category Learning in Unconstrained Environments



In the following section several offline and interactive category 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 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. Similar to the object identifications experiments performed in Section 5.2 we also use for the category learning two databases with the same objects but different experimental setups. For all offline experiments a simplified STM concept is used, where all collected object views are stored into the limited STM, similar to the experiments described in (Kirstein et al., 2008b).

5.3.1 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. 5.10 and Fig. 5.11. We use the SLP for comparison because it is the simplest architecture for this category learning task, characterizing the difficulty and the baseline performance for 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 in size (Kirstein, Wersing, & Körner 2008). The SLP output for each category c is calculated according to Eq. 4.15. The SLP as described in Section 4.4 is trained based on gradient descent. Furthermore the rejection thresholds are estimated 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.

Furthermore we investigated the effect of different shape feature sets to the categorization performance of the proposed cLVQ approach. The first set is composed of color and parts-based features, while for the second set C2 features are added, obtained with the feature extraction hierarchy, so that the overall feature dimensionality is considerably larger.

5.3.1.1 Experimental Setup

For the offline experiments two databases of the same training and test objects shown in Fig. 5.9 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.

For the second database, called *unconstrained database* in the following,

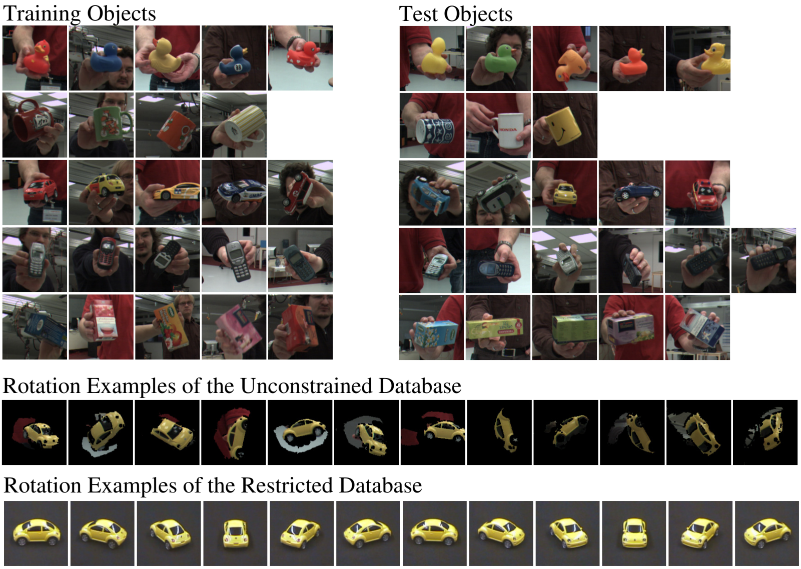

Training Objects
Test Objects
Rotation Examples of the Unconstrained Database
Rotation Examples of the Restricted Database

Figure 5.9: **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 *unconstrained database* the objects are shown in a cluttered office environment as depicted in the upper part of this figure. All objects are freely rotated by hand covering almost the complete viewing sphere. In contrast to this for the *restricted database* the same objects where shown in front of a black background and are rotated only around the vertical axis. 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.

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 described in Section 5.1.1 and Section 5.1.2. 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* this object ensemble 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 in the feature responses. These instable responses combined with a very small amount of training examples pose a considerable problem for any kind of category learning approach. 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.

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 4.3.3. 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 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.

5.3.1.2 Categorization Results

Restricted Database. The comparison of cLVQ and SLP for the *restricted database* is shown in the upper row of Fig. 5.10 and Fig. 5.11. 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 consistent to earlier experiments in Section 4.4.3. It seems that also for this 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., 2008b).

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.

Unconstrained Database. Also for the *unconstrained database* (see lower row of Fig. 5.10 and Fig. 5.11) 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

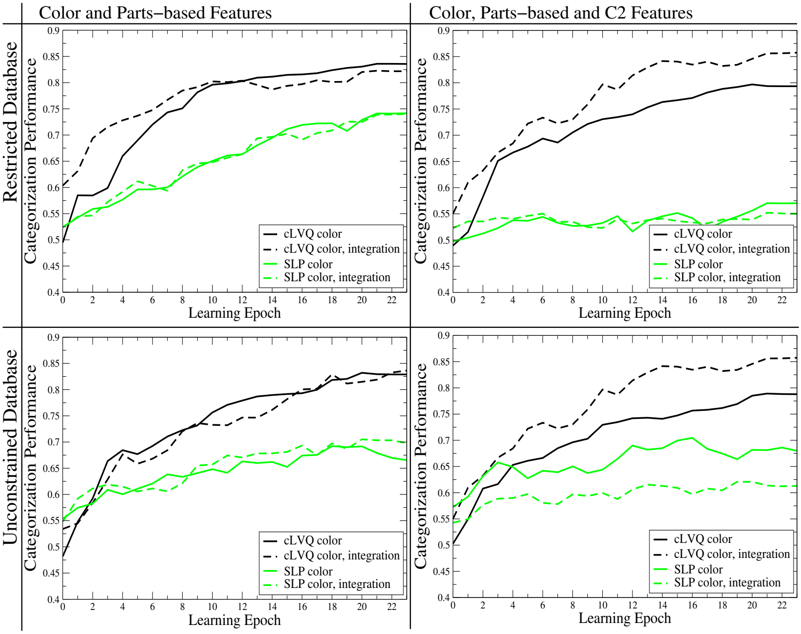


Figure 5.10: **Average Performance for Color Categories.** The performance of our proposed cLVQ method and the SLP networks are compared for the *restricted* and *unconstrained database* using the same set of training and test objects (averaged over 10 runs). All results show the categorization performance on the test set, which was never seen during the training. Additionally we tested the effect of C2 features with respect to the categorization performance. Similar to Section 5.2 also the effect of temporal integration with respect to the categorization results is investigated. After the presentation of all training objects the cLVQ method performs distinctly better for the color categories compared to the SLP networks. For all offline experiments the SLP method is superior at earlier learning stages, while the cLVQ is better at later learning steps. Finally temporal integration is only partially beneficial although a larger integration window (200 segments) was used. Especially for the SLP the categorization performance is distinctly worse for the *unconstraint database* and the usage of C2 features. In contrast to this, for the cLVQ utilizing the same setting a gain of more than 5% correct categorization performance is reached.

harder, because some parts of the objects are missing in those object views. Additionally also imprecise foreground masks cause problems for

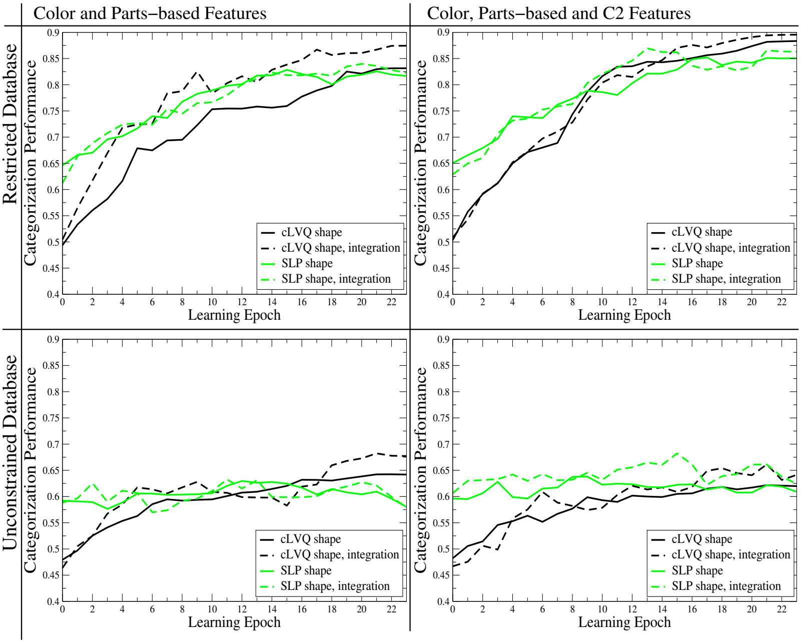


Figure 5.11: **Average Performance for Shape Categories.** Also for the shape categories the same experiments as shown in Fig. 5.10 are performed. For the shape categories our proposed cLVQ approach is compared to the SLP only slightly better after the incremental presentation of all training objects. 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 performance enhancement could be measured. In contrast to the color categories, temporal integration is at least for the final learning stages beneficial for shape categories with respect to the categorization performance. Nevertheless the performance gain compared to the identification experiments (see Fig. 5.6) is only minor.

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 further 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 detected 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 requires 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.

Temporal Integration. Similar to the identification experiments in Section 5.2 also the effect of temporal integration was investigated with respect to the enhancement of the categorization performance. In contrast to the results shown in Fig. 5.6 the required integration window must be distinctly larger to have a measurable effect on the categorization performance (integration over 200 instead of 20 classifier responses). This is basically related to fact that for categorization problems often many succeeding input vectors are misclassified, while for identification tasks typically only few succeeding views are wrongly identified. This effect is also reflected in the performance gain, where compared to the identification tasks ($\approx$ 20 %) only up to 5% for the color categories and up to 8% for the shape categories could be reached. Furthermore partially also a performance decrease was measured, which is especially visible for the color categories shown in Fig. 5.10.

5.3.2 Interactive Category Learning

Similar to the interactive learning experiment shown in Section 5.2.2 we additionally can utilize the proposed active vision system for interactive category learning with the developed cLVQ approach. Also for the category learning the system runs roughly at the frame rate of our current digital camera system of approximately 6-8 Hz, but compared to the identification system requires one additional CPU. Still the achieved frame rate is fast enough to show the desired ability of incremental and life-long learning of visual categories.

In Fig. 5.12 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 respond to 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 was 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 achieve a higher generalization per-

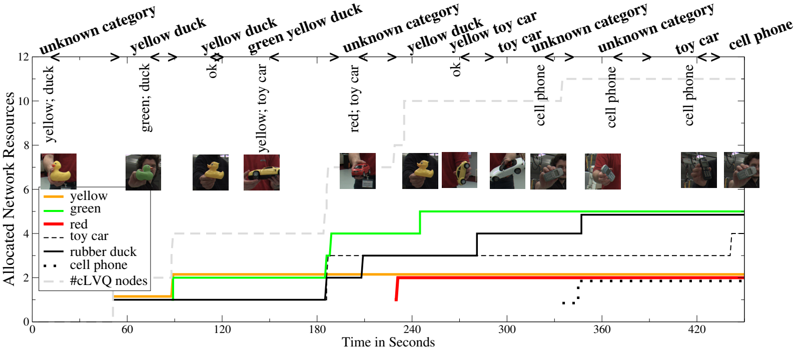


Figure 5.12: **Incremental Learning of Visual Categories.** The incremental selection of features for each category is shown over time, while presenting different objects. The model starts from a completely blank memory. Additionally the total number of cLVQ representatives is plotted, which are allocated 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.

formance 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.

5.4 Discussion

We could show that our integrated vision 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, with typically a few thousand feature dimensions, is necessary to allow interactive and incremental learning. Commonly for such feature representations additional dimension reduction techniques like the Principal Component Analysis (PCA) are required to allow online learning. Furthermore the proposed active vision system is designed in a modular fashion, so that the overall system architecture can easily be adapted to other learning tasks and scenarios.

With respect to the integration of the proposed oVQ approach, we could show that also in difficult experimental settings many natural objects can be efficiently learned online. Furthermore we showed the difficulty of learning object representations in unconstrained environments based on the comparison between the *restricted* and the *unconstrained database*. This difficulty is directly reflected in the distinctly lower identification performance for the *unconstrained database* although the same objects are used. Nevertheless with the proposed temporal integration even for such difficult learning scenarios good generalization performance can be reached. Furthermore we could show that the underlying feature representation and the similarity-based oVQ are general with respect to visual identification tasks, so that our online learning system can be easily applied also to a combined object and face recognition task. Finally the online learning system was also applied to the humanoid robot ASIMO, where people interacting the first time with the integrated system could successfully train arbitrary natural objects. This achievement especially highlights the quality of the proposed system architecture.

We furthermore 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 interactive 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. Furthermore the high feature dimensionality is also challenging for the used feature selection method, because of a large amount of possible feature candidates. However, the learning system is still able to extract small sets of category-specific features out of many possible feature candidates. Although category learning under real-world conditions is challenging, we are able to learn categories 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 visual stimuli 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.

Chapter 6

Summary and Outlook

6.1 Summary

The major topic of this dissertation is the life-long acquisition of representations. Therefore three different life-long learning methods were proposed that in combination allow learning of challenging visual identification and categorization tasks. Typically artificial neural networks are optimized for a defined function, where also the incorporation of a priori knowledge is common. In contrast to this for life-long learning such information is rare, because what kind of objects or categories are acquired is a priori unknown. Therefore the learning methods are designed in a way that many performance relevant parts are self-adapting and the total number of prespecified parameters is reduced to a minimum. This guarantees that the proposed life-long learning learning methods can be applied to a large variety of difficult recognition tasks.

Additionally we also targeted for fast interactive learning. This fast learning requirement is difficult to achieve for the target of solving challenging visual recognition tasks. Commonly online and interactive learning systems have a strongly limited representational capacity. On the other hand life-long learning approaches typically are only utilized for offline training. To overcome this limitation we propose the usage of high-dimensional and sparse feature representations. We could show that based on the sparseness of the feature representations fast interactive learning can be achieved. Furthermore the representational capacity is high enough to achieve good generalization performance for difficult experimental settings, where fully rotated and complex-shaped objects are presented in hand.

Another aspect with respect to interactive learning is the proposal of an intermediate STM representation that is incrementally collected with the similarity-based online vector quantization (oVQ) approach. This STM representation already achieves a good generalization performance but the limiting factor is the high representational cost for storing a large number of high-dimensional feature vectors. This high memory requirements led to the development of memory consolidation models that are able to incrementally build up a LTM representation. These proposed life-long learning methods are the major contribution of the presented dissertation and can in general be applied to arbitrary identification and categorization problems.

With respect to these memory consolidation approaches we first concentrated on life-long learning for identification tasks and proposed the incremental Learning Vector Quantization (iLVQ) model. The iLVQ is able to strongly reduce the representation load by iteratively consolidating the limited and changing STM into a LTM representation. This reduction of representational resources is reached without a significant drop in identification performance, which we could show in several experiments. Although the iLVQ does not enable interactive learning, it is still very useful for autonomous acting agents, because the memory consolidation could be done during periods where the system is not on duty.

To achieve the desired life-long learning capability of the iLVQ approach we proposed to extend the standard Learning Vector Quantization (LVQ) network architecture with an incremental learning of prototype nodes and a node-dependent learning rate. These modifications are the basic requirements to approach the “stability-plasticity-dilemma”. In contrast to related life-long learning architectures, where commonly the accumulated quantization error is utilized, we propose to incrementally add new prototype nodes based on wrongly classified training vectors. Especially for high-dimensional features spaces we believe that this leads to a more compact representation. Furthermore we also could reduce the number of predefined parameters compared to related work of Hamker (2001). This is especially beneficial for assistive agents operating in unconstrained environments, because typically it is unpredictable what knowledge is required to fulfill the agents duties. Therefore an a priori selection of optimal parameters is commonly very difficult.

Furthermore we proposed the category Learning Vector Quantization (cLVQ) approach for the consolidation of the object-specific STM into a category-specific LTM. In general, this knowledge transfer is considered to be more difficult compared to identification tasks. This is related to the fact that natural objects typically belong to several categories. To achieve a higher generalization performance compared to identification problems such co-occurring categories have to be independently represented. This separation is especially difficult for exemplar-based neural networks that are commonly used for life-long learning problems. To overcome this limitation we proposed a category-specific feature selection method. Combined with the incremental allocation of prototype nodes and the node-dependent learning rate this allows efficient learning of arbitrary categories. The proposed forward feature selection does not only enable an efficient separation of categories, but also enables fast interactive learning of categories.

In addition to the independent representation of categories also the incremental presentation of objects causes problems for the learning process. Basically this is because the representation of categories can undergo fundamental changes if further examples of a category are presented. Therefore in contrast to identification tasks a balance between the stability and the correction of erroneous representations is fundamental. Nevertheless we have experimentally shown that the proposed cLVQ method is able to interactively extract compact category representations.

To show the applicability of the proposed life-long learning methods in changing and cluttered environments different integrated active visions systems have been developed. All these systems combine several building blocks to enable interactive learning for object identification and categorization in common office environments. The most important parts of all systems are the figure-ground segregation, feature extraction, the life-long learning method and a state-based user interaction. Overall we could show that the oLVQ and cLVQ interactive learning approaches scale well to these more challenging and unpredictable experimental settings. Furthermore we could show that under these difficult conditions high generalization performance can be achieved, where even many offline learning methods would fail. Finally, it should be mentioned that the modular designed vision system enables a simple adaptation to different learning tasks, but also to different platforms including the humanoid robot ASIMO.

6.2 Outlook

The proposed iLVQ approach enables a resource efficient storage of many complex-shaped objects. It also can be applied to many different learning problems, where especially the low number of required parameters enhances the usability of this approach. Nevertheless this method can only be used for offline learning. Therefore one possible extension with respect to the computational cost could be a feature selection method. This would similar to the proposed categorization approach enable interactive learning. However, the feature selection process for identification tasks is more difficult. This is basically caused by the fact that for the distinction of many similar objects the combination of many different features is required. To allow time efficient learning under these conditions the feature selection methods must already strongly reduce the number of possible candidate sets. Otherwise this would cause a combinatorial explosion of feature combinations and the effect with respect to the required learning speed is most probably contra-productive.

The cLVQ model enables interactive learning of complex visual categories and can easily be applied to other problem domains like visually guided driver assistance systems. Nevertheless currently all categories are treated equally. One possible extension would be the learning of categorization hierarchies. Such a differentiation could be inspired by the proposed psychological distinction into superordinate, basic level, and subordinate categories (Rosch et al., 1976). A similar hierarchical organization could also be achieved with the proposed cLVQ method, where at the lowest level general visual attributes like “cylindrical” or “cubical” are represented. The next higher hierarchical level could then be utilized to store categories like “cup” or “cell phone”, where the highest level could further distinguish these categories into different groups like “tea cup” or “coffee pot”. This would allow to represent a large number of different categories based on an efficient reuse and combination of representations. Although such representation of categories is beneficial it is unclear how this hierarchical representation can automatically evolve without an explicit assignment of categories to the desired hierarchy level.

Furthermore the long-term memory (LTM) representations for object identification and categorization are currently distinct. The combination of both representations is also an interesting research direction. Typically for visual identification tasks each object is separated from all other trained objects. With the utilization of the category representation this differentiation could be divided into several subproblems. The basic idea is that based on the categorization decisions only similar objects are distinguished (e.g. only objects belonging to the category "cup"). We believe that this combination considerably increase the representational capacity compared to currently available object identification architectures. Additionally information from other visual modalities could ease the learning of many similar objects. This means that knowledge about the detected colors categories is most probably a good cue to efficiently separate objects belonging to a specific shape category. This combined representation would also provide the basis for many different behavioral tasks. Such tasks can range from object search to grasping, where in many cases coarse visual properties are sufficient.

As already mentioned the proposed life-long learning methods are not only restricted to visual recognition tasks. Therefore also the application to other sensory modalities or multi-modal feature representations is conceivable. Here especially audio-visual representations or the learning of visio-motor skills are of particular interest. The combination of the auditory and visual modality enables the learning of audio-visual events. Furthermore this would enable a more flexible labeling of visual representations, whereas so far these mappings are commonly predefined. The visio-motor skills are interesting with respect to the acquisition of task dependent representations. On research direction could be to extract the essential visual properties to select the most efficient grasping movement. This ability could be learning in a trial and error fashion, where typically the grasping patterns are largely predefined.

Appendix A

Feature Extraction Methods

A.1 Shape Feature Extraction

For the training of the incremental and life-long learning methods two different kinds of shape feature extraction methods are used. The first one is a feed-forward feature extracting hierarchy developed by Wersing & Körner (2003), which is based on weight-sharing and a succession of feature detection and pooling stages. The feature detectors of this architecture are obtained by unsupervised learning, providing a set of general but less object or category-specific features. The second extraction methods are so-called analytic features obtained by parts-based feature detectors. In contrast to the previous feature extraction method are the parts-based features trained in a supervised manner with respect to object or category specificity.

The input of the feature extraction methods are RGB color segments $\mathbf{J}^i = (\mathbf{J}^i_R, \mathbf{J}^i_G, \mathbf{J}^i_B)$ and the corresponding foreground mask $\boldsymbol{\xi}^i$. Note that for all image ensembles with black background all pixels in $\boldsymbol{\xi}^i$ are assigned to the foreground, therefore the complete segment is used for the extraction of shape features. For the shape feature extraction $\mathbf{J}^i$ is converted into gray-value intensity images, obtained by a weighted addition of the RGB channels:

$$\hat{\mathbf{J}}^i = \frac{1}{3}\mathbf{J}^i_R + \frac{1}{3}\mathbf{J}^i_G + \frac{1}{3}\mathbf{J}^i_B. \tag{A.1}$$

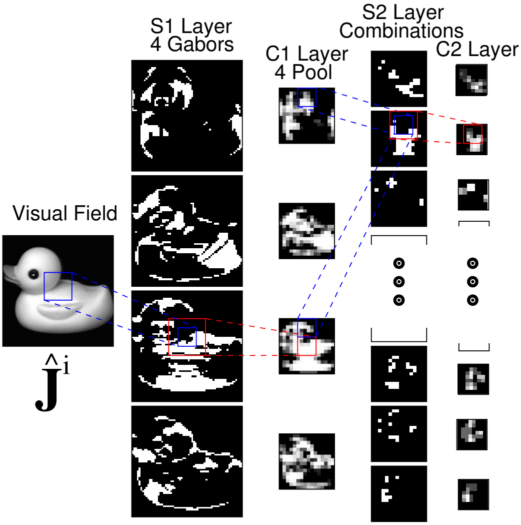


Figure A.1: **Feed-forward feature Extraction Hierarchy.** Based on a gray-scale input image $\hat{\mathbf{J}}^i$ the first feature-sensitive layer S1 performs a coarse orientation estimation using Gabor filters. Additionally a winner-take-most mechanism and a final threshold function is applied. The S1 features are pooled down to a quarter in each direction in layer C1. Neurons in the S2 layer are sensitive to local combinations of the features of the C1 layer. The C2 layer again reduces the resolution by a half in each direction.

A.1.1 Feed-Forward Shape Feature Extracting Hierarchy

We use a feed-forward feature extraction architecture (Wersing & Körner, 2003) inspired by the human ventral visual pathway as one method to extract shape features. Our feed-forward feature extraction architecture is based on weight-sharing and a succession of feature detection and pooling stages. The feature detectors are obtained through unsupervised learning based on invariant sparse coding. Figure A.1 shows an overview of this feature extracting architecture providing a high-dimensional and sparse set of feature responses.

The first feature-matching layer S1 is composed of four orientation sensitive Gabor filters $\mathbf{z}_{s1}^m$ with $m = 1, \ldots, 4$ performing a local orientation estimation. To compute the response $q_{s1}^{mi}(x, y)$ of a simple cell of this layer, responsive to feature type m at position (x, y) first the image vector $\hat{\mathbf{J}}^i$ is convolved with a Gabor filter $\mathbf{z}^m_{s1}(x,y)$:

$$q^{mi}_{s1}(x,y) = \left\{ \begin{array}{rcl} |\hat{\mathbf{J}}^i * \mathbf{z}^m_{s1}(x,y)| & : & \xi^i(x,y) > 0 \\ 0 & : & else \end{array} \right. , \tag{A.2}$$

Additionally a winners-take-most (WTM) mechanism between features at the same position is performed:

$$r^{mi}_{s1}(x,y) = \begin{cases} 0 & \text{if } \frac{q^{mi}_{s1}(x,y)}{T} < \gamma_{s1} \text{ or } T = 0, \\ \frac{q^{mi}_{s1}(x,y) - T\gamma_{s1}}{1-\gamma_{s1}} & \text{else,} \end{cases} \tag{A.3}$$

where $T = \max_k q^k_{s1}(x,y)$ and $r^{mi}_{s1}(x,y)$ is the response after the WTM mechanism, which suppresses sub-maximal responses. The parameter $0 < \gamma_{s1} < 1$ controls the strength of the competition. The activity is then passed through a simple threshold function with a common threshold ϵ_{s1} for all cells in layer S1:

$$s^{mi}_{s1}(x,y) = \Phi_{s1}\big(r^{mi}_{s1}(x,y) - \epsilon_{s1}\big), \tag{A.4}$$

where $\Phi_{s1}(x) = 1$ if $x \geq 0$ and $\Phi_{s1}(x) = 0$ else. The $s^{mi}_{s1}(x,y)$ is the final activity of the neuron sensitive to feature m at position (x,y) in this S1 layer.

The C1 layer subsamples the S1 features by pooling down to a quarter in each direction (e.g. 64x64 S1 features are pooled down to 16x16 C1 features):

$$c^{mi}_{c1}(x,y) = \tanh\big(\mathbf{s}^{mi}_{s1} * \mathbf{z}_{c1}(x,y)\big), \tag{A.5}$$

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 selective features extracted from layer C1. The so-called combination features of this S2 layer (for all experiments n=1,...,50 different shape features are used) are obtained through sparse coding (see Wersing & Körner (2003) for details). The response $q^{ni}_{s2}(x,y)$ of one S2 cell is calculated in the following way:

$$q^{ni}_{s2}(x,y) = \sum_m \mathbf{c}^{mi}_{c1} * \mathbf{z}^{nm}_{s2}(x,y), \tag{A.6}$$

where $\mathbf{z}^{nm}_{s2}(x,y)$ 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 (see Eq. A.3) and a final threshold function (see Eq. A.4) is performed in this S2 layer.

The following C2 layer again performs a spatial integration and reduces the resolution by half in each direction (i.e. 16x16 S2 features are downsampled to 8x8 C2 features). The pooling is done with the same mechanism as in layer C1 (see Eq. A.5).

A.1.2 Parts-based Shape Feature Extraction

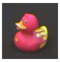
The parts-based feature detection (Hasler et al., 2007) is based on a preselected set of SIFT-descriptors (Lowe, 2004), which are designed to be invariant with regard to rotations in the image plane. Commonly in categorization frameworks such descriptors are only extracted at a small number of interest points detected e.g. by the Harris detector (Harris & Stephens, 1988) or the Kadir and Brady detector (Kadir & Brady, 2001). These interest point detectors usually respond to highly textured regions and typically ignore structureless regions. In contrast to this in the used approach these SIFT descriptors are extracted at any location in the segment $\mathbf{J}^i$, with foreground mask $\xi^i(x,y) > 0$, allowing for a greater variety of learnable objects and categories, which also includes visually less structured classes. For each segment $\hat{\mathbf{J}}^i$ the similarity $r_a^{mi}(x,y)$ $(m = 1,\ldots,500)$ between the stored feature detector $\mathbf{z}_a^m$ and the SIFT-response $\mathbf{q}_a^{mi}(x,y)$ corresponding to the segment $\hat{\mathbf{J}}^i$ at position (x,y) is calculated using the dot product:

$$r_a^{mi}(x,y) = \begin{cases} \mathbf{q}_a^{mi}(x,y) * \mathbf{z}_a^m & : \quad \xi(x,y) > 0 \\ 0 & : \quad else \end{cases} \tag{A.7}$$

The final response s_a^{mi} for the feature detector $\mathbf{w}^m$ and the current segment $\mathbf{J}^i$ is defined as:

$$s_a^{mi} = \max_{x,y}(r_a^{mi}(x,y)). \tag{A.8}$$

Thus for each feature only the maximum response is used, 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, which is unsuitable for representing a large amount of classes. Neglecting this information leads to a more compact representation with an efficient

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 A.2: **Illustration of Feature Candidate Scoring.** For each feature candidate $\mathbf{w}_a^n$ the corresponding response s_a^{ni} is calculated for each training image i. The threshold ϵ^n is chosen so that all $s_a^{ni} \geq \epsilon^n$ 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.

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 regard 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 $\mathbf{J}^i$.

A.1.2.1 Scheme for Selecting Optimal Parts-Based Feature Detectors

In the following we describe how the feature detectors $\mathbf{z}_a$ 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 class 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 $\mathbf{J}^i$ with $P_a^{ni} \geq \epsilon^n$ belong to the same category. Each image $\mathbf{J}^i$ satisfying this constraint is assigned to a constant scoring value $h_a^{ni} = 3$, which is illustrated in Fig. A.2 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 $\mathbf{w}_a^n$:

$$n = \arg\max_{n \in N} \left(\sum_i \Phi \left(h_a^{ni} + \sum_{m \in M} h_a^{mi} \right) \right), \tag{A.9}$$

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 $\mathbf{z}_a^{M+1} = \mathbf{w}^n$ and 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 classes best, while still being general enough to represent arbitrary unknown shape classes, that are not included in the set of training images.

Appendix B

Learned Feature Representation for Categorization Tasks

B.1 Selected Features of all Color and Shape Categories

In the following the selected features of all color and shape categories are shown, where in Fig. 4.11 and Fig. 4.12 only three representative color and shape categories are depicted. For this visualization ten different cLVQ networks are trained and the selected features of each category together with the scoring values are saved. The selected features of each category are sorted based on the total number of occurrences in these ten runs, while the bar height correspond to the feature score of these selected features. All features that occurred at least 7 times are considered as critical for the representation of this category, while a feature occurrence of less than 3 time are probably irrelevant or even wrong. Finally features that are selected 3-6 times indicate redundant feature sets, with similar representational capacity. Besides the occurrence of each feature the total number of selected features indicate the difficulty of the category.

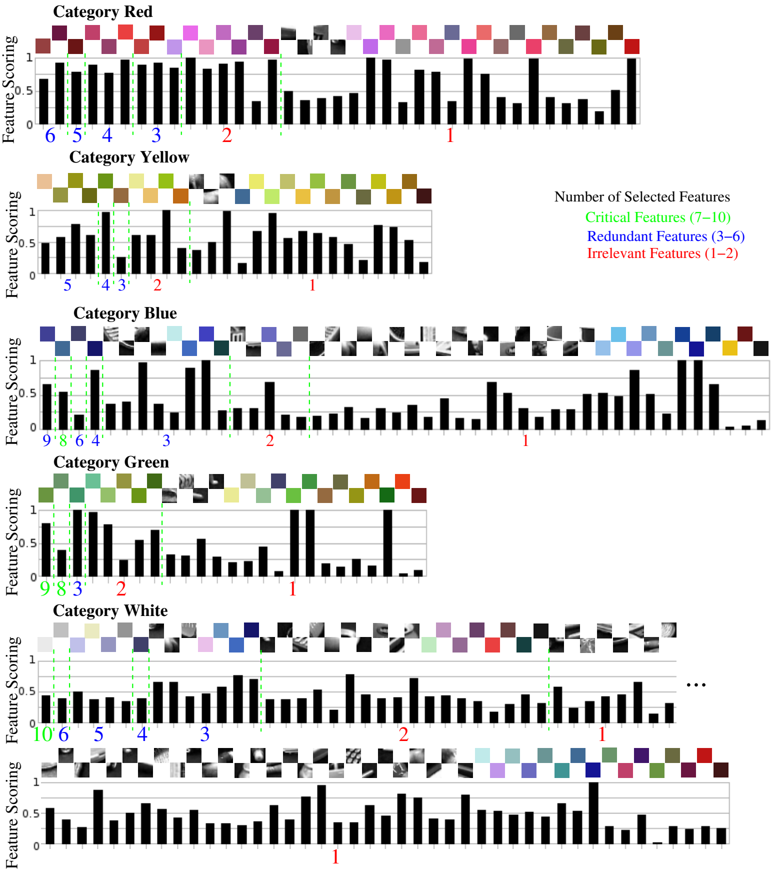


Figure B.1: Selected Features of all Color Categories ("Red", "Yellow", "Blue", "Green" and "White").

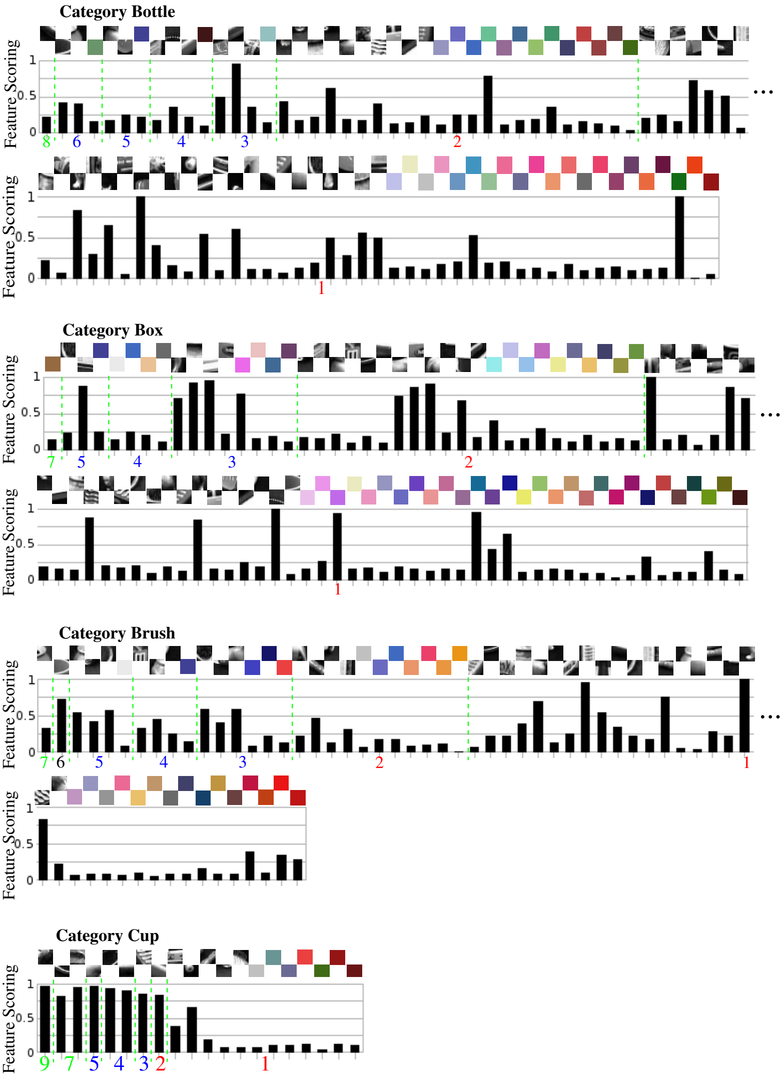
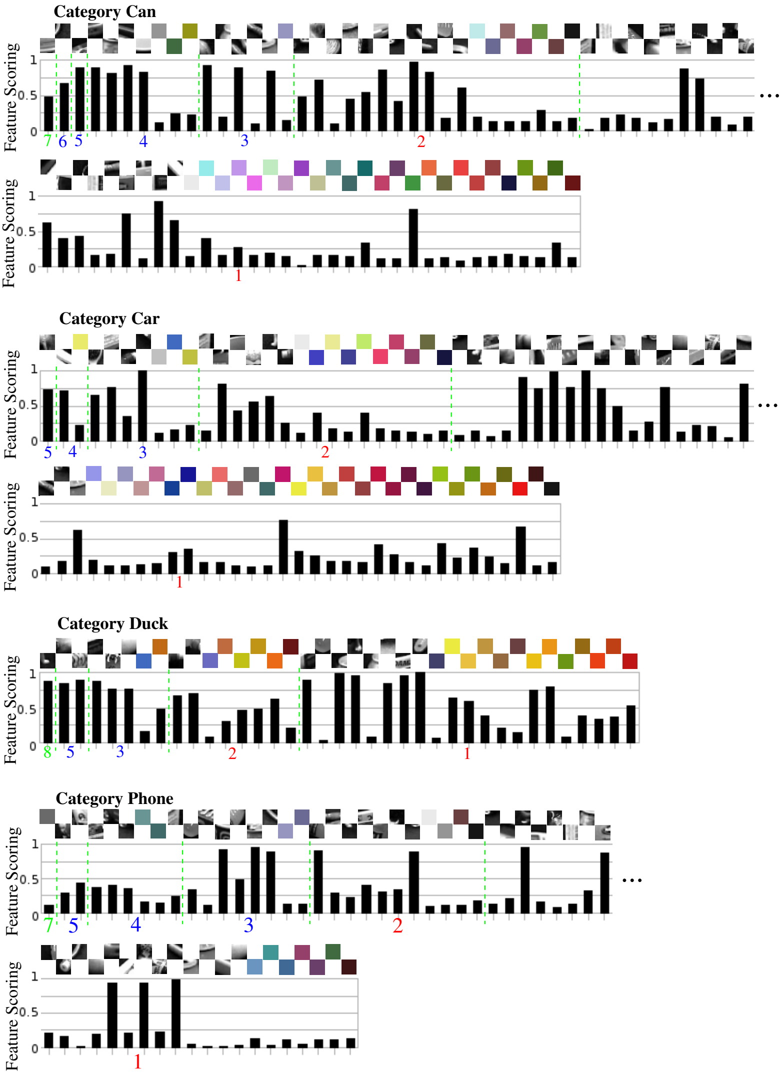


Figure B.2: Selected Features of Category “Bottle”, “Box”, “Brush” and “Cup”.


Category Can

Category Car

Category Duck

Category Phone

Figure B.3: Selected Features of Category “Can”, “Car”, “Duck” and “Phone”.

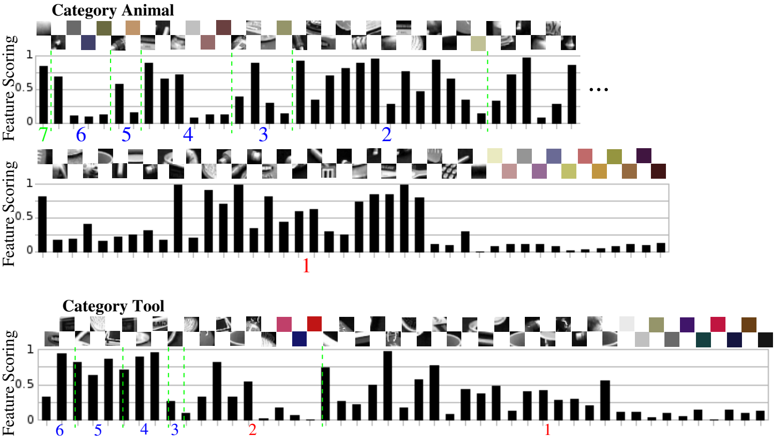


Figure B.4: Selected Features of Category "Animal" and "Tool".

Bibliography

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.

Bagnall, R. G. (1990). Lifelong education: The institutionalisation of an illiberal and regressive ideology? Educational Philosophy and Theory 22(1), 1–7.

Bekel, H., Bax, I., Heidemann, G., & Ritter, H. (2004). Adaptive computer vision: Online learning for object recognition. In Proc. German Association for Pattern Recognition (DAGM), pp. 447–453.

Biederman, I. (1987). Recognition by components - a theory of human image understanding. Psychological Review, 94, 115–147.

Bishop, C. M. (2006). Pattern Recognition and Machine Learning. Springer.

Bojer, T., Hammer, B., Schunk, D., & von Toschanowitz, K. T. (2001). Relevance determination in learning vector quantization. In Proc. of European Symposium on Artificial Neural Networks, pp. 271–276.

Bontempi, B., Laurent-Demir, C., Destrade, C., & Jaffard, R. (1999, August). Time-dependent reorganization of brain circuitry underlying long-term memory storage. Nature 400(6745), 671–675.

Broadbent, N. J., Squire, L. R., & Clark, R. E. (2004). Spatial memory, recognition memory, and the hippocampus. In Proc. of the National Academy of Sciences, pp. 14515–14520.

Broomhead, D. S. & Lowe, D. (1988). Multivariable functional interpolation and adaptive networks. Complex Systems, 2, 321–355.

Burges, C. (1998). A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery 2(2), 121–167.

Buzsáki, G. (1996). The hippocampo-neocortical dialogue. Cerebral Cortex 6(2), 81–92.

Carpenter, G. A. & Grossberg, S. (1987). ART 2: Stable self-organization of pattern recognition codes for analog input patterns. Applied Optics 26(23), 4919–4930.

Carpenter, G. A. & Grossberg, S. (1988). The ART of adaptive pattern recognition by a self-organizing neural network. IEEE Computer, 21, 77–88.

Carpenter, G. A. & Grossberg, S. (1998). Adaptive resonance theory (ART). In The Handbook of Brain Theory and Neural Networks, pp. 79–82.

Carpenter, G. A., Grossberg, S., & Iizuka, K. (1992). Comparative performance measures of Fuzzy ARTMAP, learned vector quantisation, and back propagation for handwritten character recognition. In Proc. International Joint Conference on Neural Networks (IJCNN), pp. 794–799.

Carpenter, G. A., Grossberg, S., Markuzon, N., Reynolds, J. H., & Rosen, D. B. (1992). Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps. IEEE Transaction on Neural Networks 3(5), 698–712.

Carpenter, G. A., Grossberg, S., & Rosen, D. B. (1991). Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system. Neural Networks 4(6), 759–771.

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.

Fahlman, S. E. (1988). Faster-learning variations on back-propagation: An empirical study. In Proc. of the 1988 Connectionist Models Summer School.

Fahlman, S. E. & Lebiere, C. (1990). The cascade-correlation learning architecture. Advances in Neural Information Processing Systems, 2, 524–532.

Fei-Fei, L., Fergus, R., & Perona, P. (2003). A Bayesian approach to unsupervised one-shot learning of object categories. In Proc. International Conference on Computer Vision (ICCV), pp. 1134–1141.

Fei-Fei, L., Fergus, R., & Perona, P. (2007). Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories. Computer Vission and Image Understanding 106(1), 59–70.

Fergus, R., Perona, P., & Zisserman, A. (2003). Object class recognition by unsupervised scale-invariant learning. In Proc. Computer Vision and Patern Recognition (CVPR), Volume 2, pp. 264–271.

Forman, G. (2003). An extensive empirical study of feature selection metrics for text classification. Journal of Machine Learning Research, 3, 1289–1305.

Frankland, P. W. & Bontempi, B. (2005). The organization of recent and remote memories. Nature Reviews Neurosience, 6, 119–130.

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. (2008). Modeling, Representation and Learning of Visual Categories. Ph. D. thesis, Technical University of Darmstadt.

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).

Fritz, M., Leibe, B., Caputo, B., & Schiele, B. (2005). Integrating representative and discriminative models for object category detection. In Proc. International Conference on Computer Vision (ICCV), Volume 2, pp. 1363–1370.

Fritzke, B. (1994a). Fast learning with incremental RBF networks. Neural Processing Letters 1(1), 2–5.

Fritzke, B. (1994b). 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.

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

Furao, S., Ogura, T., & Hasegawa, O. (2007). An enhanced self-organizing incremental neural network for online unsupervised learning. Neural Networks, 20, 893–903.

Furey, T. S., Cristianini, N., Duffy, N., Bednarski, D. W., Schummer, M., & Haussler, D. (2000). Support vector machine classification and validation of cancer tissue samples using microarray expression data. Bioinformatics 16(10), 906–914.

Garcia, L.-M., Oliveira, A. A. F., Grupen, R. A., Wheeler, D. S., & Fagg, A. H. (2000). Tracing patterns and attention: Humanoid robot cognition. IEEE Intelligent Systems 15(4), 70–77.

Goerick, C., Bolder, B., Janssen, H., Gienger, M., Sugiura, H., Dunn, M., Mikhailova, I., Rodemann, T., Wersing, H., & Kirstein, S. (2007). Towards incremental hierarchical behavior generation for humanoids. In Proc. International Conference on Humanoid Robots (Humanoids), pp. 248–255.

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.

Grossberg, S. (1976). Adaptive pattern classification and universal recoding, I: Parallel development and coding of neural feature detectors. Biological Cybernetics, 23, 121–134.

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.

Harris, C. & Stephens, M. (1988). A combined corner and edge detector. In Proc. Alvey Vision Conference, pp. 147–151.

Hasler, S., Wersing, H., Kirstein, S., & Körner, E. (2009). Large-scale real-time object identification based on analytic features. In Proc. International Conference on Artificial Neural Networks (ICANN), pp. 663–672.

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.

Hasselmo, M. E. & McGaughy, J. (2004). High acetylcholine sets circuit dynamics for attention and encoding; low acetylcholine sets dynamics for consolidation. Progress in Brain Research, 145, 207–231.

Hawickhorst, B. A., Zahorian, S. A., & Rajagopal, R. (1995). A comparison of three neural network architectures for automatic speech recognition. In Proc. of the Artificial Neural Networks in Engineering (ANNIE), pp. 221–226. ASME Press.

Haykin, S. (1994). Neural Networks: A Comprehensive Foundation. Macmillan.

Heinke, D. & Hamker, F. H. (1998). Comparing neural networks: A benchmark on growing neural gas, growing cell structures, and fuzzy artmap. IEEE Transactions on Neural Networks 9(6), 1279–1291.

Heinze, A., Gross, H.-M., & Surmeli, D. (2001). Integration of a Fuzzy ART approach in a biologically inspired sensorymotor architecture. In Proc. International Joint Conference on Neural Networks (IJCNN), pp. 1261–1266.

Heisele, B., Serre, T., Pontil, M., Vetter, T., & Poggio, T. (2001). Categorization by learning and combining object parts. In Proc. Advances in Neural Information Processing Systems (NIPS), pp. 1239–1245.

Izquierdo, I., Medina, J. H., Vianna, M. R., Izquierdo, L. A., & Barros, D. M. (1999). Seperate mechanisms for short- and long-term memory. Behavioral Brain Research 103(1), 1–11.

Jebara, T. & Pentland, A. (1999). Action reaction learning: Automatic visual analysis and synthesis of interactive behaviour. In Proc. International Conference on Computer Vision Systems (ICVS).

Kadir, T. & Brady, M. (2001). Saliency, scale and image description. International Journal of Computer Vision 45(2), 83–105.

Kalinke, T. & von Seelen, W. (1996). Entropie als Mass des lokalen Informationsgehalts in Bildern zur Realisierung einer Aufmerksamkeits-steuerung. In Proc. German Association for Pattern Recognition (DAGM), pp. 627–634.

Kietzmann, T. C., Lange, S., & Riedmiller, M. (2008). Incremental GRLVQ: Learning relevant features for 3D object recognition. Neurocomputing 71(13–15), 2868–2879.

Kira, K. & Rendell, L. A. (1992). The feature selection problem: Traditional methods and a new algorithm. In Proc. Association for the Advancement of Artificial Intelligence (AAAI), pp. 129–134.

Kirstein, S. (2004). Ansichtsbasierte Objektrepräsentation zur Objekterkennung. Diplomarbeit, Fakultät für Informatik und Automatisierung, Technische Universität Ilmenau. 2004-08-09/068/IN98/2233.

Kirstein, S., Denecke, A., Hasler, S., Wersing, H., Gross, H.-M., & Körner, E. (2009). A vision architecture for unconstrained and incremental learning of multiple categories. Memetic Computing 1(4), 291–304.

Kirstein, S., Wersing, H., Gross, H.-M., & Körner, E. (2008a). An integrated system for incremental learning of multiple visual categories. In Proc. International Conference on Neural Information Processing (ICONIP), pp. 811–818. Springer.

Kirstein, S., Wersing, H., Gross, H.-M., & Körner, E. (2008b). 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., Gross, H.-M., & Körner, E. (2009). A life-long learning vector quantization approach for the learning of multiple categories. Neural Networks, submitted.

Kirstein, S., Wersing, H., & Körner, E. (2005a). Online learning for object recognition with a hierarchical visual cortex model. In Proc. International Conference on Artificial Neural Networks (ICANN), pp. 487–492.

Kirstein, S., Wersing, H., & Körner, E. (2005b). Rapid online learning of objects in a biologically motivated recognition architecture. In Proc. German Association for Pattern Recognition (DAGM), pp. 301–308.

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

Kohavi, R. & John, G. (1997). Wrappers for feature subset selection. Artificial Intelligence, 97, 273–324.

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

Kohonen, T. (1990). The self-organizing map. In Proc. of the IEEE, Volume 78, pp. 1464–1480.

Kohonen, T. (1992). New developments of learning vector quantization and the self-organizing map. In Symposium on Neural Networks; Alliances and Perspectives in Senri (SYNAPSE).

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.

Linster, C., Maloney, M., Patil, M. M., & Hasselmo, M. E. (2003). Enhanced cholinergic suppression of previously strengthened synapses enables the formation of self-organized representations in olfactory cortex. Neurobiology of Learning and Memory 80(3), 302–314.

Lippmann, R. (1987). An introduction to computing with neural nets. IEEE ASSP Mag., 4, 4–22.

Littlestone, N. (1988). Learning when irrelevant attributes abound: A new linear-threshold algorithm. Machine Learning, 2, 285–318.

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

Maquet, P. (2001). The role of sleep in learning and memory. Science, 294, 1048–1052.

McClelland, J. L., McNaughton, B. L., & O'Reilly, R. C. (1995). Why there are complementary learning systems in the hippocampus and neocortex: Insights from the successes and failures of connectionist models of learning and memory. Psychological Review 102(3), 419–457.

McGaugh, J. L. (2000). Memory–A century of consolidation. Science 287(5451), 248–251.

Medina, J. H., Bekinschteina, P., Cammarotac, M., & Izquierdoc, I. (2008). Do memories consolidate to persist or do they persist to consolidate? Behavioural Brain Research 192(1), 61–69.

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).

Minsky, M. & Papert, S. (1969). Perceptrons. Cambridge, MA: MIT Press.

Moody, J. & Darken, C. J. (1989). Fast learning in networks of locally-tuned processing units. Neural Computation 1(2), 281–294.

Müller, G. & Pilzecker, A. (1900). Experimentelle beiträge zur lehre vom gedächtnis. Zeitschrift fur Psychologie, Ergänzungsband 1.

Nayar, S. K., Nene, S. A., & Murase, H. (1996). Real-time 100 object recognition system. In Proc. of ARPA Image Understanding Workshop, Palm Springs.

Ng, A. Y. & Jordan, M. I. (2001). On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes. In Proc. Advances in Neural Information Processing Systems (NIPS).

Opelt, A., Fussenegger, M., Pinz, A., & Auer, P. (2004). Weak hypotheses and boosting for generic object detection and recognition. In Proc. European Conference on Computer Vision (ECCV), Volume 2, pp. 71–84.

O'Reilly, R. C. & Norman, K. A. (2002). Hippocampal and neocortical contributions to memory: advances in the complementary learning systems framework. Trends in Cognitive Sciences 6(12), 505–510.

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.

Palmeri, T. J. & Gauthier, I. (2004). Visual object understanding. Nature Reviews Neuroscience, 5, 291–303.

Park, J. & Sandberg, I. W. (1991). Universal approximation using radial-basis-function networks. Naural Computation 3(2), 246–257.

Patil, M. M., Linster, C., Lubenov, E., & Hasselmo, M. E. (1998). Cholinergic agonist carbachol enables associative long-term potentiation in piriform cortex slices. Journal of Neurophysiology 80(5), 2467–2474.

Perkins, S., Lacker, K., & Theiler, J. (2003). Grafting: Fast, incremental feature selection by gradient descent in function space. Journal of Machine Learning Research, 3, 1333–1356.

Poggio, T. & Girosi, F. (1989). A theory of networks for approximation and learning. Technical Report 1140, Massachusetts Institute of Technology.

Polikar, R., Udpa, L., Udpa, S., & Honavar, V. (2001). Learn++: An incremental learning algorithm for supervised neural networks. IEEE Transactions on System, Man and Cybernetics (C) 31(4), 497–508.

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.

Powell, M. J. D. (1985). Radial basis functions for multivariable interpolation: A review. In Proc. IMA Conference on Algorihms for the Approximation of Function and Data, pp. 143–167.

Ranganath, C. & Blumenfeld, R. S. (2005). Doubts about double dissociations between short- and long-term memory. Trends in Cognitive Sciences 9(8), 374–380.

Reed, R. D. & Marks II, R. J. (1998). Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks. Cambridge, MA: MIT Press.

Ribot, T. (1881). Les maladies de la memoire. Appleton-Century-Crofts, New York.

Riedmiller, M. & Braun, H. (1993). A direct adaptive method for faster backpropagation learning: The rprob algorithm. In Proc. IEEE International Conference on Neural Networks, pp. 586–591.

Rigoutsos, I. & Wolfson, H. (1997). Geometric hashing: An overview. CSAE: Computational Science & Engineering, IEEE Computer Society, 4, 10–21.

Rosch, E., Mervis, C. B., Gray, W. D., Johnson, D. M., & Boyes-Braem, P. (1976). Basic objects in natural categories. Cognitive Psychology, 8, 382–439.

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. Psychological Review, 65, 386–408.

Rosenblatt, F. (1962). Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms. Washington DC: Spartan Books.

Roth, D., Yang, M.-H., & Ahuja, N. (2002). Learning to recognize 3d objects. Neural Computation 14(5), 1071–1104.

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).

Roy, D. & Pentland, A. (2002). Learning words from sights and sounds: a computational model. Cognitive Science 26(1), 113–146.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning internal representations by error propagation. In D. E. Rumelhart & J. L. McClelland (Eds.), Parallel distributed processing: explorations in the microstructure of cognition, vol. 1: foundations, Chapter 8, pp. 318–362. Cambridge, Mass.: MIT Press.

Sarter, M., Bruno, J. P., & Givens, B. (2003). Attentional functions of cortical cholinergic inputs: What does it mean for learning and memory? Neurobiology of Learning and Memory 80(3), 245–256.

Schapire, R. E. (1990). The strength of weak learnability. Machine Learning 5(2), 197–227.

Schiele, B. & Crowley, J. L. (2000). Recognition without correspondence using multidimensional receptive field histograms. International Journal of Computer Vision 36(1), 31–50.

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.

Scoville, W. B. & Milner, B. (1957). Loss of recent memory after bilateral hippocampal lesions. The Journal of Neuropsychiatry and Clinical Neurosciences, 20, 11–21.

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.

Squire, L. R. & Zola-Morgan, S. (1991). The medial temporal lobe memory system. Science, 253, 1380–1386.

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

Steil, J. J., Götting, M., Wersing, H., Körner, E., & Ritter, H. (2007). Adaptive scene-dependent filters for segmentation and online learning of visual objects. Neurocomputing 70(7-9), 1235–1246.

Steil, J. J. & Wersing, H. (2006). Recent trends in online learning for cognitive robotics. In Proc. European Symposium on Artificial Neural Networks (ESANN), Bruges, pp. 77–87.

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

Tarr, M. J. & Bülthoff, H. H. (1998). Image-based object recognition in man, monkey, and machine. Cognition, Special issue on "Images-based Recognition in Man, Monkey, and Machine", 67, 1–20.

Tetewsky, S. J., Shultz, T. R., & Takane, Y. (1995). Training regimens and function compatibility: Implications for understanding the effects of knowledge on concept learning. In Proc. of the Seventeenth Annual Conference of the Cognitive Science Society, pp. 304–309.

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.

Vijayakuma, S., D'Souza, A., & Schaal, S. (2005). Incremental online learning in high dimensions. Neural Computation 17(12), 2602–2634.

Viola, P. & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In Proc. IEEE Conference on Computer Vision and Pattern Recogntion (CVPR), pp. 511–518.

Wassermann, P. (1989). Neural Computing, Theory and Practice. Van Nostrand Reinhold.

Werbos, P. (1974). Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Science. Ph. D. thesis, Harvard University.

Wersing, H., Kirstein, S., Götting, M., Brandl, H., Dunn, M., Mikhailova, I., Goerick, C., Steil, J., Ritter, H., & Körner, E. (2006). A biologically motivated system for unconstrained online learning of visual objects. In Proc. International Conference on Artificial Neural Networks (ICANN), Volume 2, pp. 508–517.

Wersing, H., Kirstein, S., Götting, M., Brandl, H., Dunn, M., Mikhailova, I., Goerick, C., Steil, J., Ritter, H., & Körner, E. (2007b). Online learning of objects and faces in an integrated biologically motivated architecture. In Proc. International Conference on Computer Vision Systems (ICVS), pp. 383–392.

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

Wersing, H., Kirstein, S., Schneiders, B., Bauer-Wersing, U., & Körner, E. (2008). Online learning for boostrapping of object recognition and localization in a biologically motivated architecture. In Proc. International Conference on Computer Vision Systems (ICVS), pp. 383–392.

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.

Wirth, S., Yanike, M., Frank, L. M., Smith, A. C., Brown, E. N., & Suzuki, W. A. (2003). Single neurons in the monkey hippocampus and learning of new associations. Science, 300, 1578–1581.

Yang, Y. & Pedersen, J. O. (1997). A comparative study on feature selection in text categorization. In Proc. International Conference on Machine Learning, pp. 412–4120.

Zell, A. (1994). Simulation Neuronaler Netze. Addison-Wesley.

Theses:

1 The proposed learning methods automatically adapt to the difficulty of the recognition task

2 Node dependent learning rates combined with error-based incremental learning are substantial requirements for life-long learning tasks

3 The proposed iLVQ enables resource efficient life-long learning for a large variety of identification tasks

4 The cLVQ approach is suited for life-long learning of multiple categories using a single prototype memory

5 Dynamic feature weighting and selection enables fast and high performance categorization

6 The online learning STM enables interactive learning and immediate identification of many complex-shaped objects

7 The developed integrated vision system allows interactive identification or categorization of natural hand-held objects

8 The utilization of high-dimensional but sparse feature representations is beneficial for solving difficult recognition tasks

9 Exploitation of the sparsity in the feature activity allows interactive learning even in high-dimensional spaces