Towards Autonomous Bootstrapping for Life-long Learning Categorization Tasks

Stephan Kirstein, Heiko Wersing and Edgar Körner

***Abstract*— We present an exemplar-based learning approach for incremental and life-long learning of visual categories. The basic concept of the proposed learning method is to subdivide the learning process into two phases. In the first phase we utilize supervised learning to generate an appropriate category seed, while in the second phase this seed is used to autonomously bootstrap the visual representation. This second learning phase is especially useful for assistive systems like a mobile robot, because the visual knowledge can be enhanced even if no tutor is present. Although for this autonomous bootstrapping no category labels are provided, we argue that contextual information is beneficial for this process. Finally we investigate the effect of the proposed second learning phase with respect to the overall categorization performance.**

I. Introduction

In the recent decades a wide variety of category learning paradigms have been proposed ranging from generative [10], [14] to discriminative models [6], [18]. However, most research on this topic focused so far on supervised learning. The major advantage of supervised over unsupervised learning is the higher categorization performance, where the time consuming and costly collection of accurately labeled training data is its fundamental drawback. In the context of assistive systems this means that whenever the system should enhance its category representation a tutor has to specify the corresponding labels. Although we consider the interaction with a tutor as a necessary part of the early learning phase, we want to enable the system to more and more autonomously bootstrap its acquired category representation. Therefore we investigate in this paper the combination of semi-supervised and life-long learning to reduce the necessity of tutor interactions.

The basic idea of semi-supervised learning is to combine supervised with unsupervised learning [12], [2]. The advantage of this combination is typically a considerably higher performance compared to purely data driven unsupervised methods, whereas the labeling effort can be strongly reduced. Typically for semi-supervised learning the initial representation is trained based on the labeled portion of the training data. Afterwards this initial representation is utilized to estimate the correct class labels for the unlabeled portion of the training data. Commonly only unlabeled training examples with high classifier confidence are used for the bootstrapping. This guaranties a low amount of errors in the estimated labels, but this data most probably is less useful to enhance the classifier performance, because it is already well represented [17]. To overcome this limitation semi-supervised learning can be extended by active learning [13], [15], where the learning system requests the tutor-driven labeling for the currently worst represented training data.

In contrast to this we propose to use temporal context information to overcome this limitation rather than requesting additional user interactions. To use the temporal context, object views that belong to the same physical object have to be identified first. In offline experiments this typically can be easily achieved. For an autonomous system this requires the tracking of the object over a longer period, so that it is most probable that the corresponding views belong to the same physical object. Based on this object view list a majority voting can be applied. The advantage of such voting is that not only already well represented views are added to the training ensemble, but also currently wrong categorized views of the same object. We believe that such a combination has the highest potential effect with respect to an increasing categorization performance.

Although semi-supervised learning is a common learning technique (see [19] for an overview), in the context of incremental and life-long learning it has gained so far much less interest. We consider the ability of increasing the visual knowledge in a life-long learning fashion as a basic requirement for an autonomous system. Nevertheless combining semi-supervised with life-long learning is more challenging compared to typical semi-supervised learning approaches. This is because for life-long learning tasks the learning method commonly has only access to a limited amount of training data, so that the bootstrapping is normally purely based on the unlabeled training views and their autonomously assigned label information. This is in contrast to typical semi-supervised approaches, where the labeled and unlabeled training views are combined to one single training set. Furthermore to cope with the "stability-plasticity dilemma" [1] of life-long learning tasks on the one hand stability considerations are required to avoid the "catastrophic forgetting effect" [3] of the learned representation, while for the plasticity the allocation of new network resources is necessary. It is obvious that this resource allocation is considerably more difficult if the label information is unreliable as this is the case for the unsupervised training data.

The paper is structured in the following way. In the next Section II we briefly explain our category learning vector quantization (cLVQ) framework. Afterwards the modifica-

Stephan Kirstein is with the Honda Research Institute Europe GmbH, Carl-Legien-Strasse 30, 63073 Offenbach, Germany; (email: stephan.kirstein@honda-ri.de).

Heiko Wersing is with the Honda Research Institute Europe GmbH, (email: heiko.wersing@honda-ri.de).

Edgar Körner is with the Honda Research Institute Europe GmbH, (email: edgar.koerner@honda-ri.de).

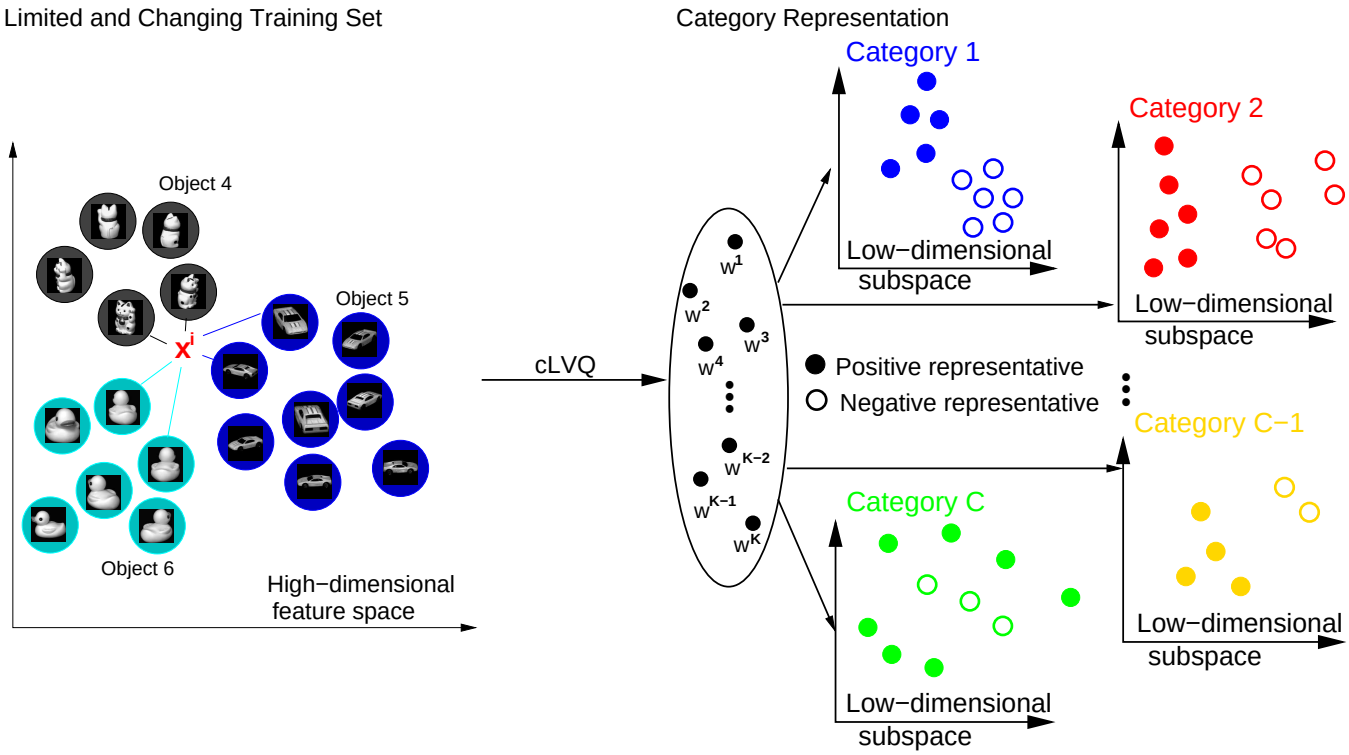

Fig. 1. **Illustration of the Category Learning Framework.** The learning with our proposed category learning vector quantization (cLVQ) approach is based on a limited and changing training set. Based on the currently available training vectors $\mathbf{x}^i$ and the corresponding target labels $\mathbf{t}^i$ the cLVQ incrementally allocates new representation nodes and category-specific features. The selected features sets for each category c enables an efficient separation of co-occurring categories (e.g. if an object belongs to several categories, which is the standard setting in our experiments) and the definition of various 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.

tions of the basic cLVQ approach and the context dependent estimation of category labels is described in Section III. In Section IV the experimental results are summarized and are discussed in Section V.

II. Category Learning Vector Quantization

Our proposed category learning approach [8] enables interactive and life-long learning and therefore can be utilized for autonomous systems, but so far we only considered supervised learning based on interactions with an human tutor. In the following we briefly describe the learning framework as illustrated in Fig.1. In the presented paper we utilized this framework for creating the category seed in a purely supervised fashion. The proposed learning approach is basically based on an exemplar-based incremental learning network combined with a forward feature selection method to enable incremental and life-long learning of arbitrary categories. Both parts are optimized together to find a balance between the insertion of features and allocation of representation nodes, while using as little resources as possible. 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. Thus we define a node insertion rule that automatically determines the number of required representation nodes. The final number of allocated nodes $\mathbf{w}^k$ and the assigned category labels $\mathbf{u}^k$ 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 based on an individual node learning rate Θ^k as proposed in [7].

Additionally 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 representation nodes of the exemplar-based network. During the learning process it selects low-dimensional subsets of features by predominantly choosing features that occur almost exclusively for this particular category. Furthermore only these selected category-specific features are used to decide whether a particular category is present or not as illustrated in Fig.1. 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.

A. Distance Computation and Learning Rule

The learning in the cLVQ architecture is based on a set of high-dimensional and sparse feature vectors $\mathbf{x}^i = (x_1^i, \ldots, x_F^i)$, where F denotes the total number of features. Each $\mathbf{x}^i$ is assigned to a list of category labels $\mathbf{t}^i = (t_1^i, \ldots, t_C^i)$. We use C to denote the current number of represented color and shape categories, 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 $\mathbf{x}^i$ with $t_c^i = 0$ have no influence on the representation of category c.

The cLVQ representative nodes $\mathbf{w}^k$ with $k = 1, \ldots, K$ are built up incrementally, where K denotes the current number of allocated vectors $\mathbf{w}$. Each $\mathbf{w}^k$ is 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. The winning nodes $\mathbf{w}^{k_{\min}(c)}(\mathbf{x}^i)$ are calculated independently for each category c, where $k_{\min}(c)$ is determined in the following way:

$$k_{\min}(c) = \arg\min_k \sum_{f=1}^{F} \lambda_{cf}(x_f^i - w_f^k)^2, \quad \forall k \text{ with } u_c^k \neq 0. \tag{1}$$

where the category-specific weights λ_{cf} are updated continuously inspired by the generalized relevance LVQ proposed by [4] . 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. Each $\mathbf{w}^{k_{\min}(c)}(\mathbf{x}^i)$ is updated based on the standard LVQ learning rule [9], but is restricted to feature dimensions $f \in S_c$:

$$w_f^{k_{\min}(c)} := w_f^{k_{\min}(c)} + \mu\, \Theta^{k_{\min}(c)}(x_f^i - w_f^{k_{\min}(c)}) \quad \forall f \in S_c, \tag{2}$$

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 $\mathbf{x}^i$. Additionally $\Theta^{k_{\min}(c)}$ is the node-dependent learning rate as proposed by [7]:

$$\Theta^{k_{\min}(c)} = \Theta_0 \exp\left(-\frac{a^{k_{\min}(c)}}{\sigma}\right). \tag{3}$$

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

B. Feature Scoring and Category Initialization

The learning dynamics of the cLVQ learning approach is organized in training epochs, where at each epoch only a limited amount of objects and their corresponding views are visible to the learning method. 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 object. Therefore for each training epoch the scoring values h_{cf}, used for guiding the feature selection process, are updated in the following way:

$$h_{cf} = \frac{H_{cf}}{H_{cf} + \bar{H}_{cf}}. \tag{4}$$

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:

$$H_{cf} := H_{cf} + 1 \text{ if } x_f^i > 0 \text{ and } t_c^i = +1, \tag{5}$$

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

$$\bar{H}_{cf} := \bar{H}_{cf} + 1 \text{ if } x_f^i > 0 \text{ and } t_c^i = -1. \tag{6}$$

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.

C. Learning Dynamics

All changes of the cLVQ network are only based on the limited and changing set of training vectors $\mathbf{x}^i$. During a single learning epoch of the cLVQ method an optimization loop is performed iteratively as illustrated in Fig. 2. The basic concept behind this optimization loop is to apply small changes to the representation of erroneous categories by testing new features v_c and representation nodes $\mathbf{w}^k$ that may lead to a considerable performance increase for the current set of training vectors. A single run through the optimization loop is composed of the following processing steps:

Step 1: Feature Testing. For each category c with remaining errors a new feature is temporally added and tested. If a category c is not present in the current training set or is error free then no modification to its representation is applied. The feature selection itself is based on the observable training vectors $\mathbf{x}^i$, the feature scoring values h_{cf} and the e_{cf}^+ values. The e_{cf}^+ is defined as the ratio of active feature entries ($x_f^i > 0.0$) for feature f among the positive training errors E_c^+ of class c. The E_c^+ is calculated in the following way:

$$E_c^+ = \{i | t_c^i = +1 \wedge t_c^i \neq u_c^{k_{min}}(\mathbf{x}^i)\}, \tag{7}$$

where the $t_c^i \in \{-1, 0, +1\}$ is defined as target signal for $\mathbf{x}^i$ and $u_c^{k_{min}}$ is the label assigned to the winning node $\mathbf{w}^{k_{min}(c)}(\mathbf{x}^i)$ of category c.

For the feature testing a candidate v_c should be added to the category-specific feature set S_c that potentially improves

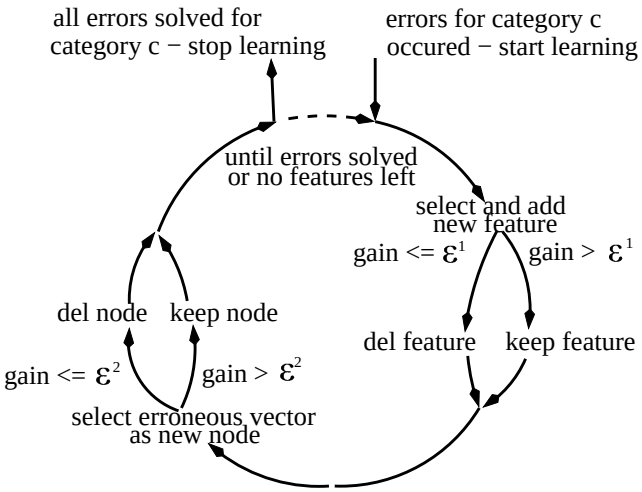

Fig. 2. **Illustration of the cLVQ Optimization Loop.** The basic idea of this optimization loop is to make small modifications to the representation of categories where categorization errors on the available training vectors occur. If the gain in categorization performance, based on all available training examples of category c, is above the insertion threshold the modification is kept and otherwise it is retracted.

the categorization performance of category c by having a high scoring value h_{cf}. Additionally the feature candidate should also be very active in the remaining training errors of this category to quickly resolve all remaining errors of this particular category. Therefore we choose:

$$v_c = \arg\max_{f \notin S_c} (e^+_{cf} + h_{cf}) \tag{8}$$

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. Thus based on all training vectors $\mathbf{x}^i$ we calculate the actual categorization performance of the erroneous categories. If the performance increase for category c is larger than the prespecified threshold ϵ^1 the v_c is permanently added and otherwise is removed and excluded for further training iterations of this epoch.

Furthermore in rare cases also the removal of already selected features is possible. This is done if the total number of negative errors $\#E^-_c > \#E^+_c$, where the E^-_c is analogous to E^+_c defined as:

$$E^-_c = \{i | t^i_c = -1 \wedge t^i_c \neq u^{k_{min}}_c(\mathbf{x}^i)\}. \tag{9}$$

The only difference is that in this case a feature $f \in S_c$ is removed from the set of selected features S_c and the performance gain is computed for the final decision on the removal.

Step 2: LVQ Node Testing. Similar to Step 1 we test new LVQ nodes only for erroneous categories. In contrast to the node insertion rule proposed in [7], where nodes are inserted for training vectors with smallest distance to wrong winning nodes, we propose to insert new LVQ nodes based on training vectors $\mathbf{x}^i$ with most categorization errors. This leads to a more compact representation, because a single node typically improves the representation of several categories. In this optimization step we insert new representation nodes $\mathbf{w}^k$ until for each erroneous category c at least one new node is inserted. As categorization labels $\mathbf{u}^k$ for these nodes only the correct targets labels for the categorization errors are assigned. For all other categories c the corresponding $\mathbf{u}^k_c = 0$, keeping all error free categories unchanged.

Again we calculate the performance increase based on all currently available training vectors. If this increase for category c is above the threshold ϵ^2, we make no modifications to LVQ node labels of the newly inserted nodes. Otherwise we set the labels u^k_c of this set of newly inserted nodes $\mathbf{w}^k$ to zero. If due to this evaluation step all u^k_c become zero then we remove the corresponding $\mathbf{w}^k$.

Step 3: Stop condition. If all remaining categorization errors for the current training set are resolved or all possible features f of erroneous categories c are tested then we start the next training epoch. Otherwise we continue this optimization loop and test further feature candidates and LVQ representation nodes.

III. Unsupervised Bootstrapping of Category Representations

Our focus is the life-long learning of visual representations. For such learning tasks normally it is unsuitable to store all previously seen training vectors. Thus we decided that the learning during the bootstrapping phase is only based on unlabeled training views and their estimated category labels, which is distinct from most commonly used semi-supervised learning methods. Before the cLVQ modifications are described in more detail, we first define the majority voting schema used for the autonomous estimation of category labels for the unlabeled training views.

A. Autonomous Estimation of Category Labels

For the autonomous estimation of category labels we first measure the network response for all available unlabeled training views based on the previously supervised trained category seed. For each individual object o in this current training set we calculate the detection rates $d^+_{oc} = D^+_{oc}/Q_o$ and $d^-_{oc} = D^-_{oc}/Q_o$, where the Q_o is defined as the number of unlabeled training views of object o. The measures d^+_{oc} indicates how reliable the category c can be detected in the views of object o, while the rate d^-_{oc} indicates how probable the category c is not present in these views. Furthermore we count the number of object views indicating the presence (D^+_{oc}) and absence (D^-_{oc}) of category c in the following way:

$$D^+_{oc} := D^+_{oc} + 1 \text{ if } u^{k_{min}}_c(\mathbf{x}^i) = +1 \tag{10}$$

and

$$D^-_{oc} := D^-_{oc} + 1 \text{ if } u^{k_{min}}_c(\mathbf{x}^i) = -1, \tag{11}$$

where the sum of $D^+_{oc} + D^-_{oc} = Q_o$.

Based on these detection rates and the predetermined thresholds ϵ^+ and ϵ^- the correct target values $t^i_c \in \{-1, 0, +1\}$ are estimated for all views of the same object.

The assignment of the target values is done in the following way:

$$t_c^i = \begin{cases} +1 & : \text{ if } d_{oc}^+ > \epsilon^+ \\ -1 & : \text{ if } d_{oc}^+ <= \epsilon^+ \ \& \ d_{oc}^- > \epsilon^- \\ 0 & : \text{ else.} \end{cases} \quad (12)$$

The selection of ϵ^+ and ϵ^- is crucial with respect to the potential performance gain of this bootstrapping phase. If these values are chosen too conservative many t_c^i become zero and the corresponding object views have no effect to the representation. On the contrary the possibility of mislabeling increases if these values are low. In general our cLVQ approach is robust with respect to a smaller amount of mislabeled training vectors, because additional network resources are only allocated if the performance gain is above the insertion thresholds ϵ^1 and ϵ^2. Nevertheless if the number of wrongly labeled training views becomes to large the categorization performance can possibly also decrease.

B. Modification of the cLVQ Learning Approach

For our first evaluation of the unsupervised bootstrapping of visual category representations we keep the incremental learning approach as in [8]. Thus also in this bootstrapping phase the learning process is subdivided into epochs and also the overall cLVQ learning dynamics is reused. This means the category representation is enhanced by making small changes to the category representation by selecting new category-specific features or by allocating additional representation nodes. Furthermore the same learning parameters like the learning rate Θ, the feature insertion threshold ϵ^1 and node insertion threshold ϵ^2 are used.

Although the same learning parameters are utilized we still want to express the reliability of the autonomously estimated category labels. This means if the reliability is low only small changes with respect to the modification of existing nodes, the allocation of new category-specific features and representation nodes should be applied. To achieve this effect all learning parameters are modulated based on the parameter $r_c^i \in \{0, \ldots, 1\}$ that is defined as follows:

$$r_{oc}^i = \begin{cases} d_{oc}^+ & : \text{ if } t_c^i = +1 \\ d_{oc}^- & : \text{ if } t_c^i = -1 \\ 0 & : \text{ if } t_c^i = 0. \end{cases} \quad (13)$$

The r_{oc}^i value is assigned to each unlabeled object views and is equal for all views of one physical object o.

For both insertion thresholds ϵ^1 and ϵ^2 this r_{oc}^i modulates the measurement of the performance gain after the insertion of a new feature v_c or representation node $\mathbf{w}^k$. In the basic cLVQ each erroneous training view that could be resolved by such slight modification of the representation is counted with 1.0. In contrast to this for the modified version of the cLVQ each resolved erroneous training view is counted as r_{oc}^i only. This means that the required amount of training vectors, necessary to reach the insertion threshold, is inversely proportional to the corresponding r_{oc}^i values (e.g. if for all current training views $r_{oc}^i = 0.8$ a factor of 1.25 views are required compared to the basic cLVQ). The fundamental effect of the modulation of ϵ^1 and ϵ^2 is that it becomes distinctly more difficult to allocate new resources the more unreliable the corresponding estimated category labels become. Therefore the allocation of category unspecific or even erroneous network resources should be strongly reduced.

Also for the adaptation of the representation nodes $\mathbf{w}^k$ the original cLVQ learning rule (see Eq. 2) is multiplied with r_{oc}^i. Besides the node dependent learning rate $\Theta^{k_{\min}(c)}$ this modification guarantees the stability of the learned visual category representation. The update step for the winning node $\mathbf{w}^{k_{\min}(c)}$ of category c is calculated as follows:

$$w_f^{k_{\min}(c)} := w_f^{k_{\min}(c)} + r_{oc}^i \mu \Theta^{k_{\min}(c)} (x_f^i - w_f^{k_{\min}(c)}) \quad \forall f \in S_c, \quad (14)$$

where r_{oc}^i is the reliability factor and the μ indicates the correctness of the categorization decision.

Besides this modulation of the learning parameters, weighted with reliability, the continuous update of the scoring values h_{cf} was deactivated for this bootstrapping phase, because these values are most fragile with respect to errors in the estimation process of category labels. A larger amount of such errors could strongly interfere globally with the previous trained category representations. This can cause a global performance decrease of all categories, while all other modifications due to the allocation of new features and representation nodes have only a local effect.

IV. Experimental Results

A. Image Ensemble

As experimental setup we use an image database composed of 44 training and 33 test objects as shown in Fig. 3. This image ensemble contains objects assigned to five different color and ten shape categories. Each object was rotated around the vertical axis in front of a black background. For each of the training and test objects 300 views are collected. The views of all training objects are furthermore subdivided into labeled and unlabeled views as illustrated at the bottom of Fig. 3. In general out of all 300 views are 200 used to train the seed of the category representation in a supervised manner, while the remaining 100 object views (view range 50–100 and 150–200) are used for the unsupervised bootstrapping of this representation. This separation into labeled and unlabeled object views means that for the autonomous bootstrapping the cLVQ has to generalize to a quite large unseen angular range of object views. Compared to a random sampling of the unlabeled object views this is more challenging, because for random selected views the appearance difference to already seen labeled views would be considerably smaller.

B. Feature Representation

For the representation of visual categories we combine simple color histograms with a parts-based feature representation, but we do not utilize this a priori separation for our category learning approach. Therefore for each object view all extracted features are concatenated into a single

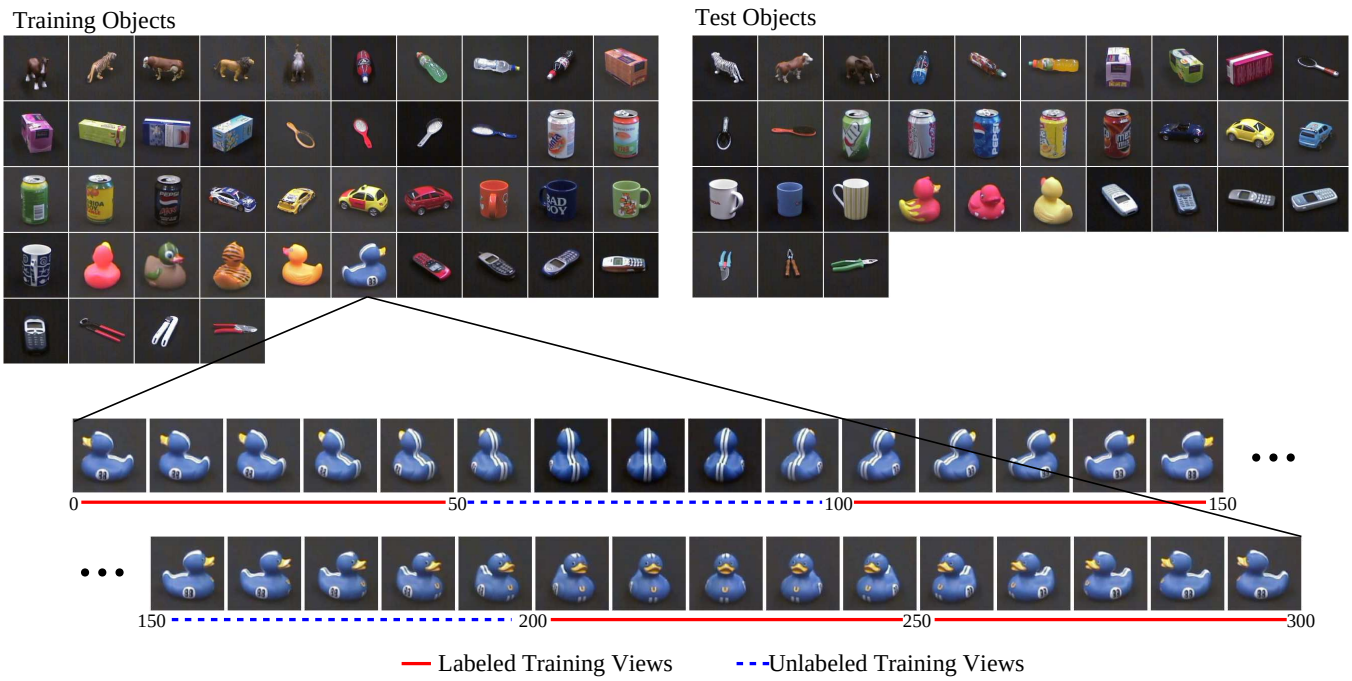

Fig. 3. **Image Ensemble.** At the top of this figure all 44 training and 33 test objects are shown. Each object was rotated around the vertical axis, resulting in overall 300 views per object. The images of the training objects are splitted into a set of labeled and unlabeled views as illustrated at the bottom.

structureless feature vector. We use color histograms because they combine robustness against view and scale changes with computational efficiency [16]. The parts-based shape feature extraction [5] is based on a learned set of category-specific feature detectors that are based on SIFT descriptors [11]. Commonly these descriptors are only determined around some highly structured interest points, while the used feature extraction method applies them at all image position. This especially allows the representation of less structured categories. For the final shape feature response only the maximum detector value is selected, so that all spatial information is neglected.

C. Categorization Performance

As already mentioned for the experimental evaluation of our semi-supervised category learning framework the training is splitted into two training phases. The first training phase is based on the basic supervised cLVQ training. Afterwards the categorization performance on the distinct test set is calculated as baseline performance. In the second training phase the categories are bootstrapped based on the incremental presentation of the unlabeled training set. Again we calculate the categorization performance to measure the effect of this second learning phase.

In general we consider two different experiments. The first experiment investigates the influence of the detection thresholds ϵ^+ and ϵ^- that are used for the autonomous category estimation of the unlabeled training views. A proper selection of these thresholds is important for a potential performance increase during the bootstrapping phase. In contrast to the first experiment, where all additional unlabeled training views are used, we are additionally interested in how the overall performance changes if more an more object views of the unlabeled training set are presented.

For the parameter search of the detection thresholds, depicted in Fig. 4, we first trained five different cLVQ networks in a supervised manner. Based on this representation the categorization performance for the 33 distinct test objects is calculated. The measured performance is used as the baseline performance. Afterwards the complete set of unlabeled training views are randomly and incrementally presented to the modified cLVQ approach and the detection thresholds ϵ^+ and ϵ^- are varied. In this evaluation we changed the threshold $\epsilon^+ \in \{0.2, 0.21, \ldots, 1.0\}$ and $\epsilon^- \in \{0.9, 0.91, \ldots, 1.0\}$. We selected a distinctly smaller range for the threshold ϵ^- because due to the selection of low-dimensional feature sets the rejection of categories is typically nearly perfect.

For this investigation we expected an approximately constant performance for the color categories, because the used color histograms should be similar to the previously seen labeled object views. In contrast to this for the shape categories we expect an increased categorization performance, because the angular range of the unlabeled object views covers approximately one third of the overall object rotation. Additionally are the fluctuations in the feature responses of the extracted parts-based features larger during the object rotation compared to the color features, so that the unlabeled object views contain further information with respect to the representation of shape categories.

The results of the threshold search experiment are shown in Fig.4. As expected in an intermediate range of the threshold ϵ^+ ($0.4 < \epsilon^+ < 0.6$) a performance increase

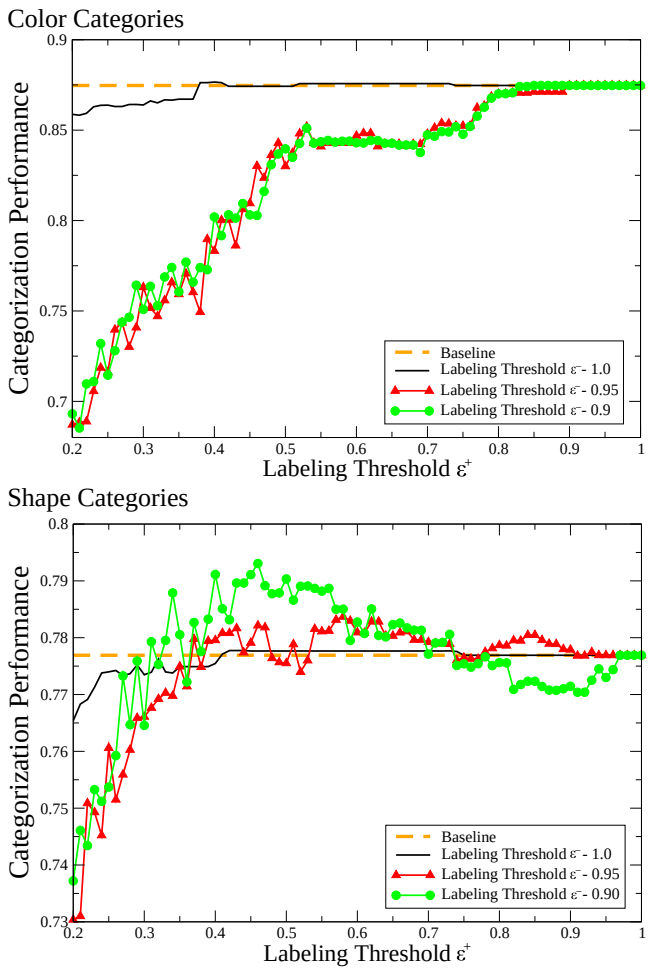

Fig. 4. **Influence of the Detection Thresholds to the Categorization Performance.** For this evaluation five different cLVQ networks are first trained in a supervised manner. The resulting category representation and its corresponding performance on the distinct test set is used a the baseline performance. For the second learning phase we incrementally added the complete set of unlabeled training views and vary the detection thresholds ϵ^+ and ϵ^-. It can be seen that for the color categories soon the performance drops if $\epsilon^+ < 0.8$, while for the shape categories in the intermediate range of $0.4 < \epsilon^+ < 0.6$ a performance increase can be measured.

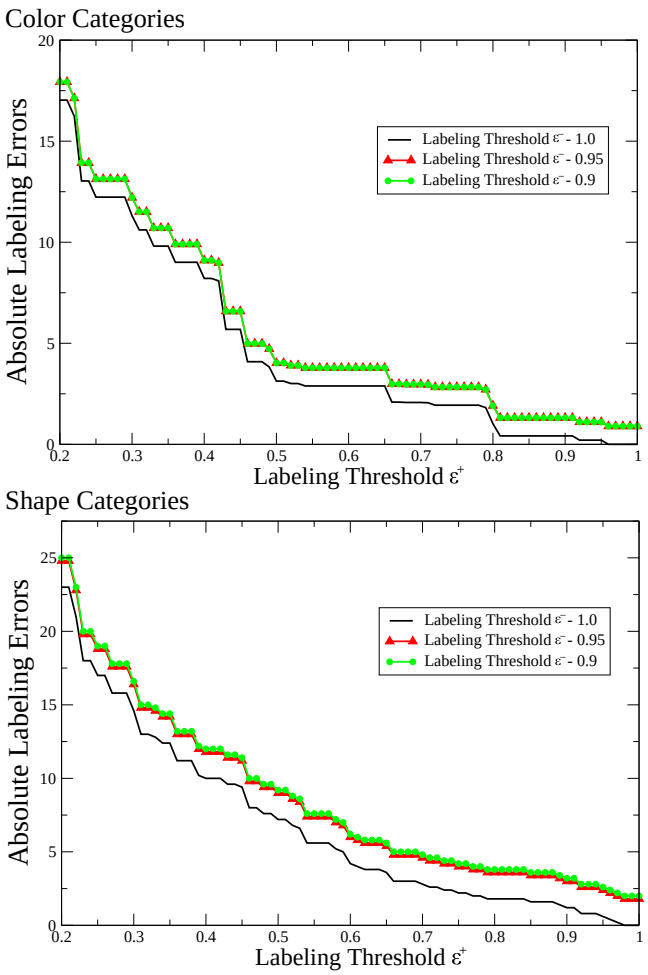

Fig. 5. **Influence of the Detection Thresholds to the Absolute Labeling Errors.** We repeated the experiment shown Fig. 4 with five different cLVQ networks, where in this evaluation we measured the average amount of wrongly labeling for different detection thresholds. As expected the average number of wrongly classified unlabeled object views increases with decreasing detection thresholds ϵ^+ and ϵ^-. Additionally it can be seen that there is a slightly higher tendency of wrongly labeled shape categories.

can be measure for the shape categories. Although we do not expected a performance gain for the color categories the performance drop for $\epsilon^+ < 0.8$ is somehow astonishing. The amount of labeling errors could be one potential reason for this performance drop. Therefore we performed a similar threshold search experiment and focused on the labeling errors with respect to different detection thresholds. The results depicted in Fig. 5 corresponds to the absolute amount of wrongly labeled bootstrapping views for the five color and ten shape categories averaged over five different cLVQ networks. As expected the average number of wrongly classified unlabeled object views increases with decreasing detection thresholds ϵ^+ and ϵ^-. The number of errors for color and shape categories are in the same range with a slightly higher tendency for wrongly labeled shape categories, reflecting the lower categorization performance of shape categories. The results in Fig. 5 show that, at least in direct comparison with the shape categories, the labeling errors are not a plausible explanation of this performance drop. During the bootstrapping process we noticed that, compared to the supervised learning part, a considerably higher amount of shape features are selected for the color categories. These additionally allocated shape features are most probable the cause for the slight performance decrease of the color categories. It indicates that for the remaining categorization errors of color categories some shape features respond as stable as the color features, so that the greedy feature selection method has difficulties to distinguish them from the correct color features.

For our second evaluation shown in Fig. 6 we selected the optimal detection thresholds for the shape categories ($\epsilon^+ = 0.5$ and $\epsilon^- = 0.9$) and investigated the effect of the performance change during the bootstrapping process by adding more and more unlabeled object views. Although in the previous investigation we could measure a performance

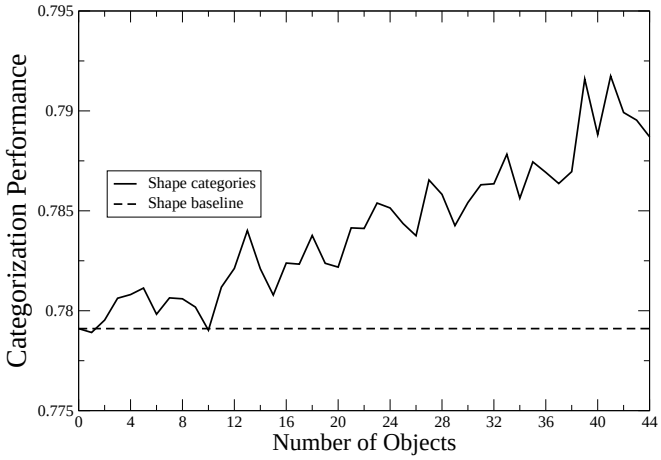

Fig. 6. **Performance Change of Incremental Adding Unlabeled Object Views.** In this experiment we selected the optimal detection thresholds $\epsilon^+ = 0.5$ and $\epsilon^- = 0.9$ for the shape categories and investigate the effect of an continuously increasing set of additional object views with respect to the change in categorization performance. In general it can be seen that the performance increases with the number of presented object views, so that the proposed modified cLVQ can be utilized for the bootstrapping of shape categories.

increase for this group of categories, it is still possible that at a certain amount of views the categorization performance starts to decrease. For an autonomous bootstrapping this would be an undesired effect, because it strongly restricts its usability for assistive systems. The optimal case would be that the performance is increasing even for large numbers of unlabeled object views.

The depicted results in Fig. 6 are averaged over 20 different networks to reduce the performance fluctuations due to the random selection of the unlabeled object set. Additionally also the presentation order of the selected objects is random. It is obvious that especially the random selection of the object set has a strong effect to the possible performance gain. Despite this smaller fluctuations in the shape category performance in general a continuously increases can be achieved (see Fig. 6). As a consequence our modified cLVQ can at least for the shape categories be used for longer bootstrapping phases without tutor interactions.

V. Discussion

In this paper we presented a first analysis for an autonomous bootstrapping of visual representations for a challenging categorization tasks. Furthermore we focused on semi-supervised learning in the context of life-long learning that is compared to the popular expectation-maximization (EM) methods [15] considerably more difficult. This higher difficulty is basically reflected in the incremental allocation of network resources even during the bootstrapping phase, while for EM-based approaches commonly no incremental learning is performed. Due to the higher difficulty we consider in this paper only the bootstrapping with familiar objects but different views.

Nevertheless based on the utilization of the temporal context of the presented unlabeled object views, we could for the shape categories achieve an enhancement of the categorization performance. It should be mentioned that this performance increase can already be measured for a quite low number of additional unlabeled object views by allocating further representation nodes and category-specific features. On the contrary for the color categories no positive effect with regard to the performance could be measured. Unfortunately partially also a performance decrease can occur for such kind of categories, which is undesired for the autonomous bootstrapping. Therefore we propose to estimate the thresholds ϵ^+ and ϵ^- for each category independently to discriminate between both visual modalities. How these thresholds can be automatically estimated should be investigated in future work.

References

[1] G. A. Carpenter and S. Grossberg. “ART 2: Stable self-organization of pattern recognition codes for analog input patterns”, *Applied Optics* 26, 4919–4930, 1987.

[2] A. P. Dempster, N.M. Laird and D. B. Rubin, “Maximum likelihood from incomplete data using the EM algorithm”, *Journal of the Royal Statistical Society*, 39, pp. 1–38, 1977.

[3] R. M. French, “Catastrophic forgetting in connectionist networks”, *Trends in Cognitive Sciences*, 3, pp. 128–135, 1999.

[4] B. Hammer and T. Villmann, “Generalized relevance learning vector quantization”, *Neural Networks*, 15, 1059–1068, 2002.

[5] S. Hasler, H. Wersing and E. Körner, “A comparison of features in parts-based object recognition hierarchies”, *In Proc. International Conference on Artificial Neural Networks*, pp. 210–219, 2007.

[6] B. Heisele, T. Serre, M. Pontil, T. Vetter, and T. Poggio, “Categorization by learning and combining object parts”, *In Proc. Advances in Neural Information Processing Systems*, pp. 1239–1245, 2001.

[7] S. Kirstein, H. Wersing and E. Körner, “A biologically motivated visual memory architecture for online learning of objects”, *Neural Networks*, 21, pp. 65–77, 2008.

[8] S. Kirstein, A. Denecke, S. Hasler, H. Wersing, H.-M. Gross and E. Körner, “A vision architecture for unconstrained and incremental learning of multiple categories”, *Memetic Computing*, 1, pp. 291–304, 2009.

[9] T. Kohonen, “Self-Organization and Associative Memory”, *Springer Series in Information Sciences, Springer-Verlag, third edition*, 1989.

[10] B. Leibe, A. Leonardis, and B. Schiele, “Combined object categorization and segmentation with an implicit shape model”, *In ECCV workshop on statistical learning in computer vision*, pp. 17–32, 2004.

[11] D. G. Lowe, “Distinctive image features from scale-invariant keypoints”, *International Journal of Computer Vision*, 60, 91–110, 2004.

[12] J. MacQueen, “Some methods for classification and analysis of multivariate observations”, *In Proc. of 5th Berkely Symposium on Mathematical Statistics and Probability*, pp. 281–297, 1967.

[13] A. McCallum and K. Nigam, “Employing EM and pool-based active learning for text classification”, *In Proc. of the Fifteenth International Conference on Machine Learning*, pp. 350–358, 1998.

[14] K. Mikolajczyk, B. Leibe, and B. Schiele, “Multiple object class detection with a generative model”, *In Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2006.

[15] I. Muslea, S. Minton and C. A. Knoblock, “Active + semi-supervised learning = robust multi-view learning”, *Proceedings of the Nineteenth International Conference on Machine Learning*, pp. 435–442, 2002.

[16] M. J. Swain and D.H. Ballard, “Color indexing”, *International Journal of Computer Vision*, 7, 11–32, 1991.

[17] G. Tur, D. Hakkani-Tür, R. E. Schapire, “Combining active and semi-supervised learning for spoken language understanding”, *Speech Communication*, 45, 171–186, 2005.

[18] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features”, *In Proc. IEEE Conference on Computer Vision and Pattern Recogntion*, pp. 511–518, 2001.

[19] X. Zhu, “Semi-supervised Learning with graphs”, *Ph.D. Dissertation, Carnegie Mellon University*, 2005