Single-example learning of novel classes using representation by similarity

Evgeniy Bart* and Shimon Ullman
Dept. of Computer Science and Applied Mathematics
Weizmann Institute of Science
Rehovot, Israel 76100
evgeniy@csail.mit.edu, shimon.ullman@weizmann.ac.il

Abstract

We describe an object classification method that can learn from a single training example. In this method, a novel class is characterized by its similarity to a number of previously learned, familiar classes. We demonstrate that this similarity is well-preserved across different class instances. As a result, it generalizes well to new instances of the novel class. A simple comparison of the similarity patterns is therefore sufficient to obtain useful classification performance from a single training example. The similarity between the novel class and the familiar classes in the proposed method can be evaluated using a wide variety of existing classification schemes. It can therefore combine the merits of many different classification methods. Experiments on a database of 107 widely varying object classes demonstrate that the proposed method significantly improves the performance of the baseline algorithm.

1 Introduction

Recent methods of visual object classification achieve high performance levels. However, they require hundreds of examples for training [12, 13, 5, 2]. The cost of collecting such large amounts of training data may be prohibitive. For example, when learning to avoid dangerous objects (e.g. predators), situations that permit acquisition of training examples are hazardous. Since the system's behavior is incorrect until a sufficient number of examples has been gathered, minimizing this number is crucial to allow adaptation to new situations. Obtaining useful performance with very few training examples is also important when the learning is incremental (i.e. the examples are presented sequentially and the system is updated after each presentation). In addition, realistic classification schemes should be able to handle a large number of classes. For example, it is estimated that humans are familiar with tens of thousands of different classes [3]. As a result, the accumulated cost of learning all classes may become excessive. Reducing as much as possible the number of required training examples may help deal with this problem.

In this paper, we propose an object classification method that can learn from a single example. With this method, a system that can already classify several classes can be extended to classify an additional novel class using a single training example. We assume

*http://people.csail.mit.edu/evgeniy

BMVC 2005 doi:10.5244/C.19.82

that for several classes, a sufficient number of examples has been available to train classifiers by one of the existing methods. These classes will be referred to as 'familiar'. The novel class is then represented by its similarity to each of the familiar classes. This similarity is measured by the already trained classifiers for the familiar classes. Intuitively, this representation can be thought of as using a name such as 'catfish' or 'dragonfly' to describe a new class by its similarity to several more common classes. More formally, we show that similarities are preserved across different class instances. Therefore, the description in terms of similarities generalizes well to unfamiliar instances of the novel class. As a result, novel object classes can be learned from a single training example. Since in this scheme generalization to within-class variability is transferred across different classes, we refer to this method as cross-generalization.

The remainder of this paper is organized as follows. In the next section, relevant previous work is reviewed. In section 3, we describe and analyze the proposed cross-generalization method. Experimental evaluation of the method is presented in section 4. We conclude with additional remarks in section 5.

2 Related previous work

The classification approach proposed here is based on obtaining outputs from a set of classifiers and combining these outputs using a higher-level classifier. This approach is known as 'classifier stacking' [14]. Classifier stacking has been combined with representation by similarities and used successfully for object recognition in [5]. The present paper extends the work in [5] in several ways. The main contribution is that our algorithm can learn from a single training example. In contrast, over 100 examples have been used for training in [5]. In addition, we present a theoretical analysis of the proposed representation to explain its empirical success in single-example learning. The final important distinction between [5] and the proposed scheme is that only similarities to individual familiar objects (under varying viewing conditions) were used in [5]. As a result, the scheme in [5] did not learn novel object classes. The tasks in [5] were restricted to recognizing individual novel objects or assigning a novel object to one of the existing, familiar classes. In contrast, in the current scheme novel objects are characterized by similarities to familiar object *classes* (rather than individual objects). This simple modification allows the proposed scheme to learn novel object classes and generalize to new instances of these novel classes. This is an important extension, because for current computer vision algorithms, object classification is a more challenging task than individual recognition.

Next, we survey additional relevant approaches, emphasizing their training requirements. Algorithms such as nearest-neighbor classification [4] require the space of input patterns to be covered with sufficient density by training examples to achieve good performance [4]. Natural object images are usually scattered over a large region (i.e. the within-class scatter is high) due to the significant variability present in natural object classes. Consequently, thousands of training examples may be required to cover this region.

Using features that are invariant to irrelevant variability decreases the within-class scatter and consequently reduces the required training set size [3, 9]. However, designing invariant features for every possible problem is not feasible [3, 9], and automatic learning of invariants [11] requires hundreds of training examples.

Linear discriminant analysis [4] (LDA, also called FLD) explicitly constructs features

to reduce the influence of within-class scatter. This is performed by maximizing the ratio of between-class to within-class scatters. However, hundreds of training examples are required for this method [2].

In [7, 6], parametric class models are used for classification. The distribution of parameters in models for the familiar classes is evaluated and used as a prior for parameters of the novel class. This prior helps avoid inaccurate parameter estimates and consequently increases the performance. However, only a single prior distribution is learned from all familiar classes in [7], and this single prior is then used for all novel classes. Such a prior biases the novel class parameters towards the values frequently appearing among the familiar classes. For novel classes with less frequent parameter values, the prior will assign low probability to the correct values of parameters, leading to degradation of performance. This undesirable behavior will not disappear when more familiar classes become available, because the parameter probabilities depend on relative fraction, rather than the absolute number, of uncommon classes. For example, suppose that 95 out of 100 familiar classes represent various quadrupeds and the remaining five classes represent different kinds of flowers. In this case, any novel class in [7] will be strongly biased towards quadrupeds. If the novel class in fact represents a new kind of flower, this bias will adversely affect the performance. In contrast, in the proposed cross-generalization scheme the influence of highly dissimilar classes is automatically reduced, and only relevant familiar classes contribute significantly. An additional advantage of the proposed cross-generalization method is that it is not restricted to a single classifier model. For example, the priors learned in [7, 6] are highly model-specific and are only suitable for the 'constellation' model. Such priors cannot be used with alternative models such as neural networks or SVMs. In contrast, cross-generalization can be used with a wide variety of classifiers, and can even combine several different classifier types. In spite of this generality, the performance of cross-generalization compares favorably to that of the model-specific schemes.

In [10], features are shared between several classes to reduce the total number of features required for classifying a large number of classes. This sharing also allows learning from few training examples. However, to obtain shared features in [10], all classes are trained simultaneously. This is an undesirable requirement, since in the current problem formulation, a novel class is assumed to appear after learning of the familiar classes is completed. In contrast, cross-generalization suits well the learning paradigm where classes become available incrementally, and it avoids the delays required to accumulate a large number of classes. In addition, the sharing algorithm used in [10] produces simple generic features, such as lines and edges. Such generic features are usually outperformed by more class-specific features. In contrast, in the proposed cross-generalization scheme the familiar classifiers are trained for optimal performance, using highly class-specific features. Although the resulting classifiers were not intended for reuse, cross-generalization allows to extract and use information from these classifiers to facilitate further learning.

3 Cross-generalization

In this section, the proposed cross-generalization scheme is presented and analyzed. In section 3.1, a brief description of the scheme is given. The analysis in section 3.2 shows that the scheme can generalize reliably to novel class images with limited training data.

3.1 Brief description of the scheme

Let $\mathscr{C}_1 \ldots \mathscr{C}_N$ denote the familiar, well-learned classes. Denote by C_i the classifier for the i'th class. Denote the class to which a pattern p belongs by $\mathscr{C}(p)$. Denote by $C_i(p)$ the output of C_i on pattern p.

In many classifiers, such as neural networks or Bayesian decision rules, meaningful interpretation can be given to the value $C_i(p)$ even when p does not belong to class $\mathscr{C}_i$. Such classifiers output a continuous value that indicates the degree of confidence that the input pattern belongs to the given class. In this case, $C_i(p)$ can be interpreted as the similarity of pattern p to the class $\mathscr{C}_i$. Note that the type of classifier C_i is irrelevant for this interpretation; arbitrary classifiers can be used (provided that they can output a similarity value), and different classifier types used for different classes can be combined.

The main hypothesis that will be used below is that similar classes (such as horses and dogs) exhibit similar within-class variability (for example, in both classes, the limbs can move relative to the body). In section 4 we empirically validate some of the implications of this hypothesis. One such implication is that the similarity of p to class $\mathscr{C}_i$ depends mainly on the overall similarity between classes $\mathscr{C}(p)$ and $\mathscr{C}_i$. The similarity is thus roughly independent of the specific object p used to make the comparison, and is well preserved across different class instances. Therefore, the similarity of the object p to class $\mathscr{C}_i$ will generalize well to other objects of class $\mathscr{C}(p)$, and is therefore a good feature to use for classification.

To learn a novel class $\mathscr{C}$ from a single example $E \in \mathscr{C}$, the similarity of E to the familiar classes is recorded in a feature vector $C_1^N(E) = [C_1(E) \ldots C_N(E)]^T$ of N numbers. Other instances of class $\mathscr{C}$ are expected to have similarity vectors close to $C_1^N(E)$, because similarities are well preserved across different instances. Therefore, nearest-neighbor classification can be used. New patterns P with similarity vectors sufficiently close to $C_1^N(E)$ are classified as belonging to $\mathscr{C}$. Simple l_2 norm is used for this classification. A pattern P is classified as belonging to class $\mathscr{C}$ when $\|C_1^N(E) - C_1^N(P)\|_2$ is below a certain threshold θ. This threshold can be adjusted to obtain the desired tradeoff between hit and false alarm rates.

3.2 Analysis

We represent input patterns as d-dimensional vectors in the input space R^d. To facilitate the analysis, the distribution of patterns of the novel class is assumed to be Gaussian with mean μ and covariance matrix Σ. Typically, the dimensionality of the input space is high, but only relatively few features are characteristic of the class. The values of the remaining features may vary widely; therefore, the Gaussian is elongated in the direction of these irrelevant features, and the corresponding eigenvalues of Σ are large.

The classification schemes considered below map the input pattern $x \in R^d$ into some feature space R^N by a mapping f. For example, the mapping used by the cross-generalization classifier is $C_1^N(x)$. In this feature space, the classification is performed using the l_2 norm. The purpose of the analysis here is to evaluate the within-class variability in the feature spaces used by various classifiers. For this, the within-class scatter $S_w = E(\|f(x_1) - f(x_2)\|_2^2)$ (the expected squared distance between two instances of the class) is calculated. Intuitively, smaller S_w implies easier classification, because all class instances have similar feature values. (Note that S_w can be changed arbitrarily by rescaling

the features and therefore should in principle be normalized by some measure of the overall scale. This normalization was not performed since our goal here is to analyze how S_w depends on class parameters rather than to give numerical estimates.)

First, the standard nearest-neighbor classifier in the input space is evaluated. This classifier uses the identity mapping as f. It can be shown [1] that $S_w = E(\|x_1 - x_2\|_2^2) = 2\,\mathrm{tr}\,\Sigma$ in this case. As mentioned above, the eigenvalues of Σ in irrelevant directions are large. Therefore, S_w can become arbitrarily large, and nearest neighbor classification will perform poorly when many irrelevant features are present.

Next, the use of the Mahalanobis distance for classification is evaluated. This corresponds to using $f(x) = \Sigma^{-1/2}x$. It can be shown [1] that $S_w = E(x_1 - x_2)^T\Sigma^{-1}(x_1 - x_2) = 2\,\mathrm{tr}\,I = 2d$. Note that each dimension contributes 2 to S_w regardless of its original variance. The influence of irrelevant features with large variance is therefore reduced. However, the influence of irrelevant features with small variance (usually corresponding to noise) is increased. This is a well-known problem with Mahalanobis distance, and in principle it should be handled by removing features that correspond to noise. However, despite this undesirable effect, S_w is bounded (unlike with nearest-neighbor classification in input space). Therefore, using Mahalanobis distance for classification is in general preferable to using l_2 norm on input patterns. However, this requires estimating Σ from a sufficient number of training examples.

Next, we show that the proposed cross-generalization scheme has favorable properties similar to those of Mahalanobis distance. However, in contrast to Mahalanobis distance, cross-generalization does not require explicitly estimating Σ and can therefore be used when the number of available training examples is limited.

The mapping used by the cross-generalization scheme is $f(x) = C_1^N(x)$; the resulting feature space will be called 'similarity space'. We assume that the familiar classes are also Gaussian with means μ_{fi} and covariances Σ_{fi}. We also assume that the classifier for the i'th familiar class uses the Mahalanobis distance for classification. Larger Mahalanobis distance indicates less similar patterns. To limit the influence of these dissimilar patterns, their similarity should be close to 0. A simple way to achieve this is to use the inverted Mahalanobis distance as similarity: $C_i(x) = \left[(x-\mu_{fi})^T\Sigma_{fi}^{-1}(x-\mu_{fi})\right]^{-1}$. In this case, $S_w = E(\|f(x_1) - f(x_2)\|_2^2) = \sum_i T_i$, where T_i is the contribution of the i'th familiar class. This contribution is given by $T_i = E(R_i)$, where

$$R_i = \frac{[d_{Mi}(x_1,\mu_{fi}) - d_{Mi}(x_2,\mu_{fi})]^2}{d_{Mi}^2(x_1,\mu_{fi})d_{Mi}^2(x_2,\mu_{fi})}.$$

Here $d_{Mi}(x,y) = (x-y)^T\Sigma_{fi}^{-1}(x-y)$ is the Mahalanobis distance between x and y, measured using the covariance matrix of familiar class i.

The numerator and denominator of R_i are always non-negative. To estimate the denominator, observe that it is proportional to the product of distances. The distance $d_{Mi}(x,\mu_{fi})$ is of the order of magnitude of $(\mu-\mu_{fi})^T\Sigma_{fi}^{-1}(\mu-\mu_{fi})$. Assuming that the novel class is well separated from the familiar classes, this distance is bounded away from zero. Then the denominator is also bounded away from zero. In this case, its exact value is not important since the inequality

$$R_i \le \frac{[d_{Mi}(x_1,\mu_{fi}) - d_{Mi}(x_2,\mu_{fi})]^2}{\min(d_{Mi}^2(x_1,\mu_{fi})d_{Mi}^2(x_2,\mu_{fi}))}$$

can be used to obtain a bound. The 'zeroth-order' approximation is to assume that the denominator is roughly constant and equal to its average value. The denominator can then be moved outside the expectation sign. In this case [1],

$$T_i = 4 \cdot \frac{\mathrm{tr}(\Sigma\Sigma_{fi}^{-1})^2 + 2\Delta_i^T \Sigma_{fi}^{-1}\Sigma\Sigma_{fi}^{-1}\Delta_i}{[2\,\mathrm{tr}(\Sigma\Sigma_{fi}^{-1})^2 + 4\Delta_i^T \Sigma_{fi}^{-1}\Sigma\Sigma_{fi}^{-1}\Delta_i + (A+B)^2]^2},$$

where $A = \Delta_i^T \Sigma_{fi}^{-1}\Delta_i$, $B = \mathrm{tr}\,\Sigma\Sigma_{fi}^{-1}$, $\Delta_i = \mu - \mu_{fi}$. More accurate approximations should take into account the higher-order moments of the denominator.

Assume first that the familiar class $\mathscr{C}_i$ is similar to the novel class $\mathscr{C}$. The similarity is measured by Δ_i, the distance between their means. For similar classes, Δ_i is small, and the terms involving it can be neglected. We assume that similar classes have similar structure, as determined by their covariance matrices. To measure the similarity of Σ and Σ_{fi}, note that for equal matrices, $\Sigma = \Sigma_{fi}$ implies $\Sigma_{fi}^{-1}\Sigma = I$. Therefore, $\mathrm{tr}\,\Sigma_{fi}^{-1}\Sigma = \mathrm{tr}\,\Sigma\Sigma_{fi}^{-1} = \mathrm{tr}\,I = d$. Approximating $\mathrm{tr}\,\Sigma_{fi}^{-1}\Sigma \simeq d$ for similar classes, T_i can be simplified to $T_i \simeq 4/d^3$.

Assume now that $\mathscr{C}_i$ and $\mathscr{C}$ are highly dissimilar, and the distance Δ_i is large. Since the denominator contains terms of higher order in Δ_i than the numerator, $T_i \simeq 0$. Notice that no explicit knowledge of dissimilarity between $\mathscr{C}_i$ and $\mathscr{C}$ is required. The contribution T_i from dissimilar classes is automatically reduced in the similarity space.

The intuition behind these results is as follows. The familiar classifiers are trained to ignore irrelevant within-class variability. Since similar classes have similar structure, this capacity to ignore irrelevant variability can be generalized to the novel class. Therefore, the contribution T_i from similar classes is bounded. For dissimilar classes, the similarity values are (by definition) small; therefore, their contribution is negligible.

Therefore, $S_w \leq 4N/d^3$ (where N is the total number of familiar classes), i.e. S_w is bounded by a constant independent of the class parameters. This implies that instances of the novel class become clustered tightly in the similarity space no matter how broadly they are distributed in the input space. Therefore, the similarity between an instance of the novel class and a familiar class is indeed preserved across different instances of the novel class, and novel class instances can be classified reliably in the similarity space. Since the transformation to similarity space does not require explicit estimation of the novel class parameters (such as Σ), the scheme can be trained using a limited number of examples.

Note that the scatter S_w was used only to estimate the difficulty of the classification task. In the similarity space the scatter is automatically reduced, and therefore algorithms that use representation by similarity need not minimize scatter explicitly. (Such explicit minimization of scatter would require hundreds of training examples.)

4 Results

In this section, we present the empirical evaluation of the proposed cross-generalization algorithm.

To train classifiers for the familiar classes, we used a scheme similar to [12]. In this scheme, objects are represented using a set of selected sub-images, called fragments. These fragments are extracted from training images and combined in a linear classifier trained to discriminate between class and non-class images based on presence or absence of particular fragments.

Note that the baseline classifiers used for the familiar classes represent the spatial configuration explicitly. Since the cross-generalization classifier relies on the baseline classifiers to calculate similarity, it incorporates the spatial configuration implicitly. Therefore, explicit representation of spatial configurations is not necessary in the cross-generalization classifier.

4.1 Empirical scatter analysis

In this section, the within-class scatter in the similarity space was compared to scatter in the input space to empirically verify the conclusions of section 3.2. The experiment was performed using images from the ETH database [8]. Five familiar classes (cows, cups, horses, pears, and tomatoes) were used. The familiar classifiers were trained as described above and were used subsequently to evaluate similarities in the cross-generalization algorithm. The class of dogs was used as the novel class. The within-class scatter, S_w, was calculated for this class in the similarity space. As mentioned above, the scatter should be normalized by some measure of overall scale. For this, the between-class scatter S_b, defined as the average squared distance of a positive example from a negative example, was calculated. The ratio of the within-class to between-class scatter was $S_w/S_b = 0.18$. The scatters were also measured in the input space, by using the l_2 norm to evaluate similarity of input images directly. In this case, the ratio of scatters was 0.83. The conclusion is that the input patterns are better clustered in the similarity space than in the input space, as predicted in section 3.2.

4.2 Classification experiments

In this section, the classification performance of cross-generalization is evaluated and compared to the baseline method described in [12]. The experiments were performed using a database of 107 widely varying classes. Of these, 101 classes were from the Caltech database [7], and six additional classes were incorporated. These additional classes included animals and animal faces. The images were obtained and preprocessed as described in [7]. All images were scaled to a height of 45 pixels. Most classes contained between 40 and 100 examples. In addition, non-class images, which did not contain any of the familiar classes, were used as negative examples. A set of 400 non-class images was used for training. A separate set of 324 non-class images was used for testing. Example images are shown in Figure 1; more examples can be found in [7].

Cross-generalization performs classification in the similarity space, and it is natural to compare it to input-space classification. However, input-space classification in our case corresponds to direct matching of images. It is well-known that this method gives poor classification performance. Our experiments confirmed that input-space classification produced essentially random results (data not shown due to space limitations). The more interesting comparison with the baseline algorithm is reported below.

The cross-generalization algorithm was tested using the leave-one-out method. Each of the 107 classes was tested. With each class, the remaining 106 classes were used as the familiar classes for cross-generalization. This is similar to experiments in [7, 6]. Each familiar classifier was trained using $2/3$ of the available class images as positive examples and the training non-class images as negative examples. (The same negative examples were used for all classes.) For each familiar class, 25 fragments were selected

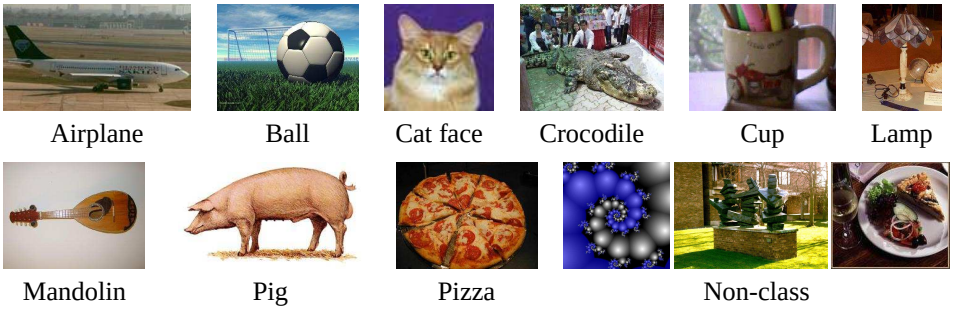

Figure 1: Examples of images used in the experiments.

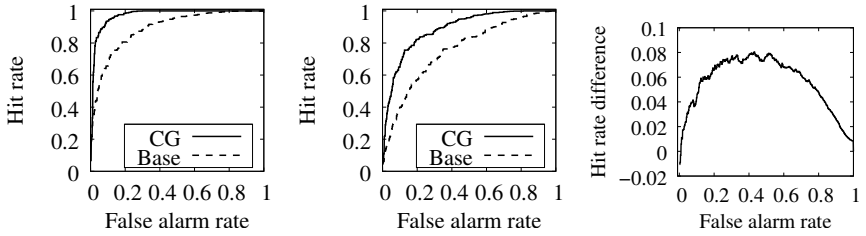

Figure 2: Comparison of cross-generalization to the baseline. Trained with two positive examples. (a), (b): Sample ROC curves. (a): cars class. (b): helicopter class. *X* axis: false alarm rate. *Y* axis: hit rate. Solid line: cross-generalization. Dashed line: baseline algorithm. (c): Difference of ROC curves averaged over all 107 classes. *X* axis: false alarm rate. *Y* axis: difference of hit rates. Positive values indicate advantage of cross-generalization.

and a linear classifier was trained [12]. Next, a cross-generalization classifier was created for the novel class. Only a single example of the novel class was selected at random for training. Experiments were repeated 10 times with different random choices. No negative examples have been used in training. Following training, the cross-generalization classifier was tested using the remaining class images and the testing set of non-class images.

Classifier performance for each class can be characterized by the area under the ROC curve. However, since random guessing would give an area of 0.5, we use instead *performance margin*, defined as $2 \cdot (Area - 0.5)$, where *Area* is the area under the ROC curve. Perfect classification would give a margin of 1, while random guessing would give a margin of 0. The average margin obtained by the cross-generalization algorithm was 0.46 ± 0.03. This margin corresponds to average area under ROC of 0.73, which is slightly better than the average area of 0.71 reported in [7]. (Note that the experiments described here were more challenging than those in [7] due to the reduced resolution of the images.) As mentioned in section 2, [7] uses a highly model-specific scheme. In

contrast, cross-generalization is a general scheme, not tuned to some particular model. Therefore, cross-generalization could in principle fail to exploit some information available to the model-specific scheme. The comparable performance of cross-generalization and the model-specific scheme indicates that this did not happen. The conclusion is that cross-generalization managed to extract a significant amount of information from the familiar classifiers without detailed information about these classifiers.

Next, the performance of cross-generalization was compared to the baseline algorithm. Since the baseline algorithm requires multiple positive and negative examples to operate, it was supplied with a minimal set of two positive examples per class. In addition, two of the training non-class images were used as negative examples. The cross-generalization algorithm was also supplied with two positive examples (no negative examples have been used). The performance of the learned classifiers was subsequently tested on a data set containing images of the novel class (except for the training images) and the testing non-class images. Example ROC curves for two classes are shown in Figure 2. To compare performance across all 107 classes, the difference between the cross-generalization and baseline ROC curves was calculated for each class. This difference is a curve which, for every false alarm rate, gives the difference of hit rates. Positive difference indicates advantage of cross-generalization. The difference curves of the 107 classes were averaged. The average difference curve is shown in Figure 2(c).

The average performance margin of cross-generalization algorithm in this experiment was 0.52 ± 0.02. On average, the margin of cross-generalization was $38\,\% \pm 10\,\%$ higher than that of the baseline algorithm. The difference is highly significant (paired t test, $p < 0.001$). The conclusion is that cross-generalization significantly outperforms the baseline algorithm.

5 Discussion

We have described an object classification scheme in which the past effort expended for learning is reused to facilitate learning of novel classes. This approach is called cross-generalization, and it allows to obtain useful classification performance from a single training example. The implementation of the proposed scheme is particularly simple because each familiar classifier is treated as a black box and only the output values of these classifiers are used.

The similarity between the novel class and the familiar classes in the proposed method can be evaluated using a wide variety of existing classification schemes. Cross-generalization can therefore combine the merits of many different classification methods. Despite this generality, the scheme is competitive with highly classifier-specific methods.

A limitation of the proposed scheme is its inherent slowdown caused by using multiple classifiers. When 106 familiar classifiers are used (section 4.2), it takes ~ 1.8 seconds to classify an image on a 1.5 GHz Centrino laptop. If only a single classifier was used, the classification time would reduce to 0.025 seconds. However, it should be noted that the individual classifiers used in the scheme do not interact, and therefore the scheme can easily be implemented on massively parallel architectures. In particular, biological implementation seems plausible. In the future, we plan to investigate how the computational complexity of the scheme could be reduced. One possible direction is to select only the most relevant familiar classifiers and discard the rest. Preliminary results indicate that 10

familiar classifiers are sufficient to obtain performance comparable to using all 106 familiar classifiers. Alternatively, the computational complexity of the ensemble of familiar classifiers could be reduced by sharing features among different familiar classifiers [10].

Acknowledgment

This work was conducted at the Moross Laboratory for Vision and Motor Control supported by IMOS Grant 3-992.

References

[1] E. Bart and S. Ullman. Derivation of bounds for external cross-generalization, 2005. http://people.csail.mit.edu/evgeniy/Publications/2005/BartUllmanExtCGDerivation.pdf.

[2] P. N. Belhumeur, J. Hespanha, and D. J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. In *ECCV*, pages 45–58, 1996.

[3] I. Biederman. Visual object recognition. In S. F. Kosslyn and D. N. Osherson, editors, *An Invitation to Cognitive Science*, volume 2, pages 121–165. MIT Press, 2nd edition, 1995.

[4] R. Duda, P. Hart, and D. Stork. *Pattern Classification*. John Wiley & Sons, 2001.

[5] S. Edelman. *Representation and recognition in vision*. MIT Press, 1999.

[6] L. Fei-Fei, R. Fergus, and P. Perona. A Bayesian approach to unsupervised one-shot learning of object categories. In *ICCV*, 2003.

[7] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories. In *CVPR 2004 Workshop on Generative-Model Based Vision*, 2004.

[8] B. Leibe and B. Schiele. Analyzing appearance and contour based methods for object categorization. In *CVPR*, Madison, Wisconsin, June 2003.

[9] K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. In *ECCV*, pages 128–142, 2002.

[10] A. Torralba, K. P. Murphy, and W. T. Freeman. Sharing features: efficient boosting procedures for multiclass object detection. In *CVPR*, 2004.

[11] S. Ullman and E. Bart. Recognition invariance obtained by extended and invariant features. *Neural Networks*, 17:833–848, 2004.

[12] S. Ullman, M. Vidal-Naquet, and E. Sali. Visual features of intermediate complexity and their use in classification. *Nature Neuroscience*, 5(7):682–687, 2002.

[13] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR*, pages 511–518, 2001.

[14] D. H. Wolpert. Stacked generalization. *Neural Networks*, 5(2):241–259, 1992.