Cross-generalization: learning novel classes from a single example by feature replacement

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

Abstract

We develop an object classification method that can learn a novel class from a single training example. In this method, experience with already learned classes is used to facilitate the learning of novel classes. Our classification scheme employs features that discriminate between class and non-class images. For a novel class, new features are derived by selecting features that proved useful for already learned classification tasks, and adapting these features to the new classification task. This adaptation is performed by replacing the features from already learned classes with similar features taken from the novel class. A single example of a novel class is sufficient to perform feature adaptation and achieve useful classification performance. Experiments demonstrate that the proposed algorithm can learn a novel class from a single training example, using 10 additional familiar classes. The performance is significantly improved compared to using no feature adaptation. The robustness of the proposed feature adaptation concept is demonstrated by similar performance gains across 107 widely varying object categories.

1. Introduction

The problem of classification is usually approached by using example images to train a classifier to deal with novel object classes [4]. Current classification methods achieve high levels of performance, but they typically require hundreds of training examples [1, 9, 13, 15, 17, 19].

The cost of collecting large amounts of training data may be prohibitive in some cases. 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).

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.

Training examples are necessary in the current methods to discover useful features in the data, to identify and reject poor and unreliable features, and to train a classifier for the learned class. In addition to feature appearance, classifiers can use spatial configuration of the features. In the scheme used below, the contribution of feature appearance is more significant than that of their spatial configuration. Therefore, the discussions below will focus on feature appearance.

The main hypothesis used in this paper is that a feature is likely to be useful for a novel class (e.g. dogs) if a similar feature proved effective for a similar class (e.g. horses) in the past. This hypothesis is validated empirically in section 4. The hypothesis can be used to facilitate learning, in the following manner. Assume that for several object classes (e.g. horses and cows), a sufficient number of examples has been available to perform feature extraction by one of the existing methods [1, 17, 19]. These classes will be referred to as 'known' or 'familiar'. The task is then to learn a new class (e.g. dogs) from a single example. The main problem is that a single training example by itself is insufficient to decide which of the features of the novel class are useful for classification. Using incorrect features will deteriorate performance. To overcome this problem, features that proved useful for already familiar classes are adapted to the novel class. This adaptation is performed by replacing the familiar features with novel features that have similar appearance, as illustrated in Figure 1. In this manner, the merit of each novel feature is predicted based on its similarity to features that proved useful in the past. With this approach, a system that can already classify horses and cows can be extended to classify dogs using only a single dog example. Since in this way generalization across different classes is achieved, we refer to this method as cross-generalization. We demonstrate (section 4) that cross-generalization allows to obtain useful classification performance from a single training example.

A remarkable characteristic of human cognition is the ability to adapt and reuse a wide variety of previously acquired skills when learning a new task [12]. This is achieved by recognizing and exploiting the similarities between previously acquired skills and the skills required to perform the new task. The proposed cross-generalization algorithm can be seen as an initial step towards imitating this important capability in visual classification.

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

2. Related previous work

Typically, hundreds of examples are required for training by the current classification algorithms [1, 9, 13, 15, 17, 19]. When the number of training examples is insufficient, ‘data manufacturing’ [4] can be used. In this method, each training example is used to generate several additional examples by performing simple image transformations such as adding random noise or introducing small distortions. If a generative model that describes the allowed variations of images within the class is known, then such a method can significantly improve classification. However, models to account for natural variability of objects within a class are usually not available.

Next, we discuss several algorithms that can handle small training sets. In [5, 6], parametric class models are used for classification. The distribution of parameters in models for the familiar classes is estimated and used as a prior for parameters of the novel class. This prior helps avoid inaccurate parameter estimates and increases performance compared to using no prior [6]. The two main differences from the cross-generalization algorithm proposed here are the following. First, in [6], a single prior distribution is learned from all familiar classes. This single prior is then used for all novel classes. Such a prior will bias 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. Similar bias towards the common classes is present in [10]. 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 [6, 10] 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, only similar familiar classes contribute significantly to the novel class. Therefore, uncommon novel classes are not biased by a significant number of irrelevant classes. Continuing the example, the novel flower class will not be affected by the 95 familiar quadruped classes because features that are useful for quadrupeds will be dissimilar to features found in the flower class. Therefore, only the five relevant flower classes will contribute significantly to the novel class. Second, the algorithm proposed below uses class-specific features and relies mainly on feature appearance. The appearance of familiar features is adapted to be characteristic of the novel class (Figure 1). This adaptation helps prevent confusion between the novel class and similar familiar classes. In contrast, features used in [6] are more generic in appearance. (Features in [6] are constrained to lie in a low-dimensional subspace common to all classes and all features.) The algorithm in [6] therefore relies mostly on shape (spatial configuration of features). It is therefore complementary to the approach in this paper that focuses on the selection of new features from a single example. The two methods can be combined to achieve improved performance (see section 5).

In [16], features are shared between several classes. This strategy is useful for reducing the total number of features required for classifying a large number of classes. The application of feature sharing to learning from few training examples has also been described [16]. However, the motivation behind [16] is not single-example learning; therefore, this application has several disadvantages. First, the sharing algorithm used in [16] produces simple generic features, such as lines and edges. Such generic features are usually outperformed by more class-specific features [18]. In contrast, the proposed cross-generalization algorithm uses features that are highly specific to the novel class (Figure 1). Second, to obtain shared features in [16], 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. An additional disadvantage of simultaneous training is that the accumulated amount of training

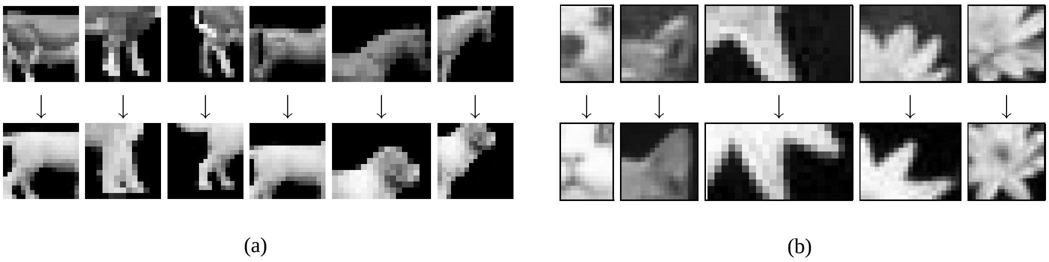


Figure 1. Feature adaptation. (a) Top row: features extracted from multiple images of cows (first three) and horses (last three), as described in section 3.1. Bottom row: features adapted to the dogs class by the proposed cross-generalization algorithm (section 3.2), using a single dog image. Each cow or horse feature is replaced by the most similar dog feature, as indicated by the arrows. Ordinarily, a single training example would be insufficient to determine the merit of the resulting features. With cross-generalization, their usefulness can be predicted from the usefulness of the original features. Each novel feature is similar in overall appearance to the corresponding familiar feature above it, but at the same time has some distinctly dog-like characteristics. (Images from the ETH database [7].) (b) Similar to (a), left to right: dog face feature adapted to cat face, cougar adapted to cat, starfish and two lotus features adapted to water lily.

data may present computational difficulties [16]. 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, since each class can be trained separately, the cross-generalization scheme has favorable time and memory requirements.

In [8], prior knowledge about useful feature types and sizes is utilized to facilitate feature selection for novel classes. However, this knowledge is insufficient for single-example learning (10 positive examples and 5000 negative examples were used in [8]). In addition, human involvement is required to extract the information about useful feature parameters from the familiar classes. In contrast, the proposed cross-generalization algorithm performs feature adaptation with a single positive example, using no negative examples. In addition, feature adaptation is completely automatic.

3. Cross-generalization

In this section, we first outline the general classification scheme used in our experiments. We then describe how feature adaptation within this scheme can be performed to produce cross-generalization.

3.1. Classification by image fragments

A number of recent methods use image patches, or fragments, as features for classification [1, 5, 14, 17, 19]. We used in our experiments a method similar to the fragment-based algorithm described in [17]. However, the idea of feature adaptation is applicable also in other similar schemes [1, 19].

In [17], objects are represented using a set of selected sub-images, called fragments. Example fragments are shown in Figure 1. These fragments are combined in a linear classifier trained to discriminate between class and non-class images based on presence or absence of particular fragments. Spatial configuration of fragments is represented by specifying the rough location of each fragment relative to a reference frame common to all fragments (similar to the scheme described in [11]).

During the learning stage, fragments are created from example images by extracting sub-images of multiple sizes and at multiple locations. With each fragment, its location in the original image is stored and used to determine relative locations of different fragments. A large pool of candidate fragments is extracted first, and then a subset of useful fragments is selected from this initial pool. The selection is based on the measure of mutual information between the fragment and the class it represents [2, 17].

To classify an image, this set of fragments is searched for in the image, using the absolute value of normalized cross-correlation. For each fragment F, the relevant locations in the image are determined by the location of F relative to the common reference frame (see above). Image patches at the relevant locations are compared with F, and the location with the highest correlation is selected. When the correlation exceeds a pre-determined threshold θ_F, the fragment is considered present, or *active*, in the image. The threshold θ_F for each fragment F is determined automatically during learning to maximize the information delivered by the fragment about the class [2].

A linear classifier is then used for classification. Let $f_i = 1$ if F_i is present in the given image and $f_i = 0$ otherwise. The classifier labels the image as belonging to the class if

$$\sum_i w_i(f_i) > T. \tag{1}$$

The weights $w_i(f_i)$ of each fragment are estimated by the log-likelihood ratio

$$w_i(d) = \log \frac{P(f_i = d|C)}{P(f_i = d|\overline{C})}, \tag{2}$$

commonly used in signal detection theory. Here $d \in \{0,1\}$ represents the presence or the absence of the fragment, and each fragment has two weights, $w_i(0)$ and $w_i(1)$. $P(f_i = 1|C)$ is the probability that F_i is present in an image that belongs to class C and $P(f_i = 1|\overline{C})$ is the probability that F_i is present in an image that does not belong to class C. $P(f_i = 0|C)$ and $P(f_i = 0|\overline{C})$ are interpreted similarly. The probabilities are estimated from the training images. The global classifier threshold T can be adjusted to obtain the desired tradeoff between hit and false alarm rates.

3.2. Cross-generalization by feature adaptation

Image fragments provide a compact representation of object classes and can be used for efficient and accurate classification [1,14,19]. However, hundreds of training examples are required for fragment extraction [1,17]. When there is only a single training image, the method described in section 3.1 will often erroneously select poor features and produce unsatisfactory performance. The reason is that a given training image contains multiple uninformative fragments in addition to the more informative ones, and training examples are required to separate them and identify a subset of features that can be used reliably.

To identify useful classification features in the absence of additional training examples, we use the following hypothesis: a feature F is likely to be useful for class C if a similar feature F' proved effective for a similar class C' in the past. This hypothesis is validated empirically in section 4. Using this hypothesis, effective features for the novel class can be identified from a single training example by taking advantage of features from familiar classes. This is done by adapting each familiar feature to the novel class. This adaptation is performed by replacing a given familiar feature with the most similar feature from the novel class. Next, we describe in detail how this replacement is performed.

Let $C_1 \dots C_N$ denote the familiar classes. These are classes already learned by the system, and each class is represented by a set of fragments. Denote the i'th fragment of the k'th familiar class by F_i^k, its threshold by θ_i^k, and its weights by $w_i^k(\cdot)$. The classifier for the k'th class is then

$$\sum_i w_i^k(f_i^k) > T^k. \tag{3}$$

The task is now to learn a novel class C from a single example image E. Each familiar fragment F_i^k is searched for in E using normalized cross-correlation. The location in E with the highest correlation is selected; this highest correlation will be denoted S_i^k. From that location, a fragment F^{new} of the same size as F_i^k is extracted. (We say that F^{new} was *nominated* by F_i^k.) As in section 3.1, the location in E from which F^{new} was extracted is stored and used to determine the relative locations of different fragments. Examples of fragments obtained automatically in this manner are shown in Figure 1.

Note that the familiar fragments are only used to facilitate selection. They do not appear in the final classifier, which consists entirely of novel fragments. This is important for two reasons. First, using novel fragments that are specific to the novel class increases the classification performance. Second, the use of class-specific novel fragments reduces the risk of confusion between the novel class and similar familiar classes. For example, if horse fragments were used directly to classify dogs, the classifier would run the risk of confusing the new dog class with the old horse class. (Experimental validation of these claims is not reported due to space limitations.)

The number of adapted fragments may be large. (In the experiments in section 4.1, more than 2000 fragments were obtained.) Such large number is typically unnecessary to achieve good performance, and therefore it is desirable to select a smaller subset of the adapted fragments. In addition, fragments nominated to the novel class by dissimilar familiar classes are expected to perform poorer than fragments nominated by similar familiar classes. For example, given a car fragment F_i^k, it is possible to find and extract its best-matching dog fragment. However, fragment similarity S_i^k is likely to be small, and the performance of the resulting adapted fragment may be inferior, compared to using better-matching fragments from similar classes (such as cows or horses). Therefore, it is desirable to remove the fragments with low S_i^k. In the experiments described below, a fixed small number (namely, 25) of the fragments with the highest S_i^k was selected. In this manner, by identifying the fragments with the highest S_i^k and the familiar classes to which these fragments belong, the familiar classes most similar to the novel class were determined.

Next, thresholds should be determined for the newly selected fragments. Each novel fragment F^{new} was nominated by some fragment F_i^k to which it was similar. Since F_i^k and F^{new} belong to different classes, it is desirable to prevent F^{new} from being detected in images that contain F_i^k. This can be achieved by choosing the threshold for F^{new} to exceed S_i^k, the correlation between F^{new} and F_i^k. In the experiments described below, good performance was obtained by setting the threshold to $1.1S_i^k$.

Finally, a classifier to recognize the new class should be constructed. A linear classifier that uses the fragments F_i^{new} has the form

$$\sum_i w_i^{new}(f_i^{new}) > T^{new}. \quad (4)$$

The classifier is determined by the values of the weights $w_i^{new}(\cdot)$; it is therefore sufficient to estimate these. This estimation was performed by reusing the weights of the fragment F_j^k that nominated F_i^{new}, i.e., by setting $w_i^{new}(0) = w_j^k(0)$, $w_i^{new}(1) = w_j^k(1)$. (An alternative scheme of setting $w_i^{new}(1) = 1$, $w_i^{new}(0) = 0$ was also evaluated, but performed significantly poorer.) Experiments evaluating the performance of the resulting classifier are described in the next section.

4. Results

In this section, we present the classification results obtained by the cross-generalization algorithm (section 3.2) and compare them to a standalone training algorithm. This standalone algorithm is identical to the original fragment-based algorithm described in section 3.1 and does not use the familiar classes when learning the novel class.

The tests were performed on a database of 107 widely varying classes. Of these, 101 classes were from the Caltech database [6], and six additional classes were incorporated. These additional classes included animals and animal faces. The images were obtained and preprocessed as described in [6]. The images were scaled to 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. Some examples are shown in Figure 2; more examples can be found in [6]. This large database was used to test the applicability of cross-generalization to different object classes (section 4.1). However, the method is not limited to large databases. To illustrate this, a subset of 11 classes was selected, and performance was also tested on this subset (section 4.2).

4.1. Large database tests

The cross-generalization algorithm was tested using the leave-one-out method. Each of the 107 classes was tested.

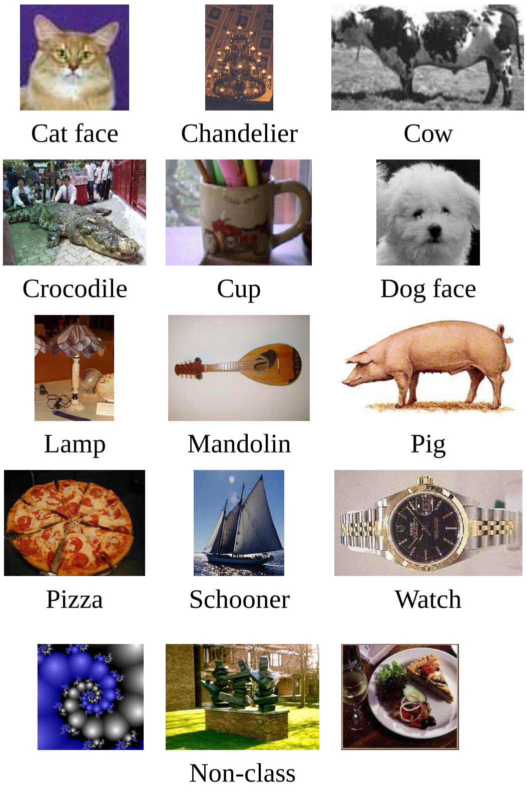


Figure 2. Examples of images used in the experiments.

With each class, the remaining 106 classes were used as the familiar classes for cross-generalization. This is similar to experiments in [5, 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 and a linear classifier was trained. 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 were used in training. The classifier was trained using the cross-generalization algorithm, as described in section 3.2. Subsequently, the 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 in the cross-generalization experiments was 0.5 ± 0.02 (average ± standard error of the mean). The difference from random is highly significant ($p \ll 0.001$). As shown in Figure 4(b), even 10 positive and 10 negative examples are not sufficient for the standalone scheme to achieve similar performance. The margin of 0.5 corresponds to average area under ROC of 0.75, which is slightly better than the average area of 0.71 reported in [6]. Note, however, that the experiments described here were more challenging than those in [6] due to the reduced resolution of the images. In addition, cross-generalization currently makes only rudimentary use of the spatial configuration of the features. It is likely that the incorporation of spatial structure similar to [6] in cross-generalization will lead to further improvement in performance.

Next, the performance of the cross-generalization algorithm was compared to the standalone algorithm. Since the standalone 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. Plots in Figure 3 illustrate the performance achieved on two of the 107 classes tested. As can be seen, cross-generalization performance is superior to the performance of the standalone algorithm.

To compare performance across all 107 classes, the difference between the cross-generalization and standalone ROC curves was calculated for each class. This difference is a curve which, for every false alarm rate, gives the difference in 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 4(a).

The average performance margin of cross-generalization algorithm in this experiment was 0.55 ± 0.02. On average, the margin of cross-generalization was $52\,\% \pm 15\,\%$ higher than that of the standalone algorithm. The difference is highly significant (paired t test, $p < 0.001$). The conclusion is that cross-generalization significantly improves the performance of the standalone algorithm.

The purpose of the next experiment was to determine how the performance of both algorithms depends on the training set size. For this, the number of positive training examples was varied between 2 and 10. The number of negative examples for each test was equal to the number of positive examples. (Only the standalone algorithm used these negative examples.) For each training set size, the average performance margin of the standalone algorithm and of cross-generalization was calculated. The resulting values are plotted in Figure 4(b). As can be seen, cross-generalization performance improves with the number of training examples and remains significantly above the standalone performance even with 10 positive and 10 negative examples. This suggests that the benefits of cross-generalization are not limited to single-example learning and can be substantial even for moderately-sized training sets.

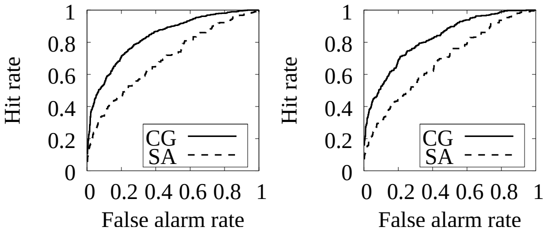
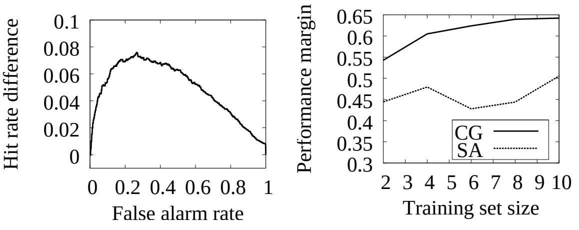


Figure 3. Performance of the cross-generalization and standalone algorithms trained with two positive examples. (The standalone algorithm also used two negative examples which were unavailable to the cross-generalization algorithm.) Full database (107 classes, section 4.1). Left: crayfish class. Right: mandolin class. Plotted: ROC curves. X axis: false alarm rate. Y axis: hit rate. Solid line: cross-generalization. Dashed line: standalone algorithm. As can be seen, cross-generalization improves the standalone performance significantly by using information from the familiar classes.

4.2. Small database tests

The tests described in section 4.1 were performed using 107 classes. This large database was used to test the applicability of cross-generalization to different object classes. However, the method is not limited to large databases. To illustrate this, a subset of 11 classes (beaver, brontosaurus, cougar, crocodile, dalmatian, elephant, emu, flamingo, gerenuk, hedgehog, leopard) was selected and cross-generalization performance was tested on this subset. The results of all experiments were similar to those reported in section 4.1. Due to space limitations, only a subset of the experiments is reported here.

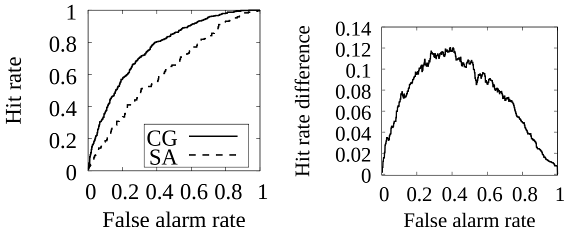
The cross-generalization algorithm was tested using the leave-one-out method, as in section 4.1. With each of the 11 classes, the remaining 10 classes were used as the familiar classes for cross-generalization. Only a single example of the novel class was selected at random for training the cross-generalization classifier. Experiments were repeated



(a)

(b)

Figure 4. Left: Difference of cross-generalization and standalone ROC curves averaged over the 107 classes (section 4.1). X axis: false alarm rate. Y axis: difference of hit rates. Positive values indicate advantage of cross-generalization. Both algorithms were trained with two positive examples, the standalone algorithm also used two negative examples. Right: Average performance margin as function of training set size. Full database (107 classes, section 4.1). X axis: number of positive examples in training (standalone algorithm also used same number of negative examples). Y axis: performance margin. Solid line: cross-generalization. Dashed line: standalone algorithm.



(a)

(b)

Figure 5. Performance of the cross-generalization and standalone algorithms trained with two positive examples. (The standalone algorithm also used two negative examples which were unavailable to the cross-generalization algorithm.) 11-class subset (section 4.2). Left: ROC curve for the crocodile class. X axis: false alarm rate. Y axis: hit rate. Solid line: cross-generalization. Dashed line: standalone algorithm. Right: difference of cross-generalization and standalone ROC curves averaged over the 11-class subset. X axis: false alarm rate. Y axis: difference of hit rates. Positive values indicate advantage of cross-generalization.

10 times with different random choices. No negative examples were used. The classifier was trained using the cross-generalization algorithm, as described in section 3.2. Subsequently, the classifier was tested using the remaining class images and the testing set of non-class images. The average performance margin obtained in this experiment was 0.33 ± 0.03. The difference from random is highly significant ($p \ll 0.001$).

Next, the performance of the cross-generalization algorithm was compared to the standalone algorithm. As in section 4.1, both algorithms were supplied with two positive examples (the standalone algorithm was also supplied with two negative examples). The plot in Figure 5(a) illustrates the performance achieved on one of the classes. As can be seen, cross-generalization performance is superior to the standalone algorithm performance.

The averaged difference of ROC curves is shown in Figure 5(b). The average performance margin of cross-generalization algorithm in this experiment was 0.4 ± 0.03. On average, the margin of cross-generalization was 80 % ± 20 % higher than that of the standalone algorithm. The difference is highly significant (paired t test, $p < 0.01$). The conclusion is that cross-generalization significantly improves the performance of the standalone algorithm even when only 10 familiar classes are available.

5. Conclusions

We described a classification algorithm that is able to learn a novel class from a single example. This algorithm performs feature selection for the novel class by using prior experience with features from familiar classes. The merit of a novel feature is determined by its similarity to other features, already known to be useful, rather than by using additional training examples. As a result, novel classes can be learned from a single example, without even using negative examples. This is a significant advantage over many previous methods that require multiple positive and negative examples [1, 9, 13, 15, 17, 19]. The ability to learn completely without negative examples is particularly interesting, since the underlying classification scheme used for familiar classes depends substantially on negative examples [17]. This ability resembles human learning capability: when taught a new class, humans rarely receive explicitly any negative examples.

One limitation of the classification method described in section 3 is its limited use of spatial relations between features. We therefore plan in the future to include the use of spatial information in the cross-generalization scheme. Currently, cross-generalization relies mainly on appearance to perform classification. Our experiments demonstrate that appearance of a novel class can be learned from a single example. In contrast, [6] uses mainly shape (spatial relations), while the appearances of the features are more generic (constrained to lie in a low-dimensional subspace common to all classes and all features). It has been shown [5, 6, 10] that shape can also be learned when the number of training examples is limited. Therefore, a promising direction for future research is to perform cross-generalization of appearance and shape simultaneously. This can be achieved by combining the scheme proposed here with [6]. Since cross-generalization of appearance and shape separately achieves significant performance improvements over standalone algorithms, combining the two information sources is likely to further increase the performance.

An additional topic for further investigation is improved use of similarity between the novel class and the familiar classes. Currently, the adapted features are selected independently, based on the similarity of each adapted feature to its nominating feature. This strategy may be prone to errors caused by spurious similarities between individual features of otherwise dissimilar classes. To avoid such errors, feature selection can be restricted to familiar classes that nominate a large number of features with high similarity. In this manner, selection will be based on the overall similarity of a familiar class to the novel class, as measured by the similarity of multiple features (rather than just a single feature). The overall similarity between classes determined in this manner can also be used to improve the shape estimation. To estimate the shape of the novel class, only the prior information from similar familiar classes should be used (rather than information from all familiar classes, as in [6]).

The experiments in section 4 demonstrate that cross-generalization outperforms the standalone algorithm even for training sets of moderate size. This suggests that prior knowledge remains useful even when multiple training examples are available. Therefore, it would be interesting in the future to combine cross-generalization with the regular standalone algorithm to perform incremental learning. In the initial stages of learning, cross-generalization should be used to achieve useful initial performance. In the later stages, as more training examples become available, features adapted by cross-generalization and features learned by the standalone algorithm can be combined to achieve optimal performance.

Acknowledgement

This work was supported in part by a grant from the Israeli Ministry of Science and Technology.

References

[1] S. Agarwal and D. Roth. Learning a sparse representation for object detection. In *ECCV*, pages 113–127, 2002.

[2] E. Bart, E. Byvatov, and S. Ullman. View-invariant recognition using corresponding object fragments. In *ECCV, Part II*, pages 152–165, 2004.

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

[6] 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.

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

[8] K. Levi, M. Fink, and Y. Weiss. Learning from a small number of training examples by exploiting object categories. In *Proceedings of CVPR Workshop on Learning in Computer Vision and Pattern Recognition*, 2004.

[9] K. Levi and Y. Weiss. Learning object detection from a small number of examples: the importance of good features. In *CVPR*, 2004.

[10] E. Miller, N. Matsakis, and P. Viola. Learning from one example through shared densities on transforms. In *CVPR*, pages 464–471, 2000.

[11] P. Moreels, M. Maire, and P. Perona. Recognition by probabilistic hypothesis construction. In *ECCV*, pages 55–68, 2004.

[12] S. Pinker. *How the Minds Works*. W. W. Norton, 1999.

[13] H. A. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *PAMI*, 20(1):23–38, 1998.

[14] E. Sali and S. Ullman. Combining class-specific fragments for object recognition. In *BMVC*, pages 203–213, 1999.

[15] H. Schneiderman and T. Kanade. A statistical model for 3D object detection applied to faces and cars. In *CVPR*, 2000.

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

[17] 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.

[18] M. Vidal-Naquet and S. Ullman. Object recognition with informative features and linear classification. In *ICCV*, pages 281–288, 2003.

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