Evolutionary Feature Synthesis for Image Databases

Anlei Dong, Bir Bhanu, Yingqiang Lin
Center for Research in Intelligent Systems
University of California, Riverside, California 92521, USA
{adong, bhanu, yqlin}@cris.ucr.edu

Abstract

The high dimensionality of visual features is one of the major challenges for content-based image retrieval (CBIR) systems, and a variety of dimensionality reduction approaches have been proposed to find the discriminant features. In this paper, we investigate the effectiveness of coevolutionary genetic programming (CGP) in synthesizing feature vectors for image databases from traditional features that are commonly used. The transformation for feature dimensionality reduction by CGP has two unique characteristics for image retrieval: 1) nonlinearlity: CGP does not assume any class distribution in the original visual feature space; 2) explicitness: unlike kernel trick, CGP yields explicit transformation for dimensionality reduction so that the images can be searched in the low-dimensional feature space. The experimental results on multiple databases show that (a) CGP approach has distinct advantage over the linear transformation approach of Multiple Discriminant Analysis (MDA) in the sense of the discrimination ability of the low-dimensional features, and (b) the classification performance using the features synthesized by our CGP approach is comparable to or even superior to that of support vector machine (SVM) approach using the original visual features.

1 Introduction

Content-based image retrieval (CBIR) systems are designed to automatically extract various visual features from images, such as color, texture, shape, structure, and use them to represent images in the computer. The collection of such visual features usually yields the high-dimensionality of the feature space, which deteriorates retrieval performances due to the well-known "curse of dimensionality". Thus, a key task in CBIR is to find the most discriminant features from the original feature collection, so that both retrieval precision and search speed is improved.

Fisher discriminant analysis (also called Multiple discriminant analysis (MDA) for multiple-class case) [1] is a widely-used approach to find discriminating features due to its straight-forward idea of making linear data projection by maximizing the ratio of the *between scatter matrix* and the *within scatter matrix*. The kernel trick [2] can be combined with Fisher discriminant analysis (e.g., [3]), so that the nonlinear generalization of MDA is achieved with classification improvement.

Swets and Weng [4] propose a *self-organizing hierarchical optimal subspace learning and inference framework* (SHOSLIF) for image retrieval. Such hierarchical linear analysis is a nonlinear approach in nature. Hastie and Tibshirani [5] present the approach of *discriminant adaptive nearest neighbor* (DANN) for classification. Based on the analysis of local dimension information, they also propose a global dimensionality method by pooling local discriminant information. Wu et al. [6] reduce the feature dimensionality for image databases using the approach of *weighted multi-dimensional scaling* (WMDS), whose main characteristic is preserving the local topology of the high dimensional space. Su et al. [7] exploit principle component analysis (PCA) for dimensionality reduction by using relevance feedback.

In this paper, we investigate the effectiveness of coevolutionary genetic programming (CGP) [8] in generating composite operator vectors for image databases, so that feature dimensionality is reduced to improve retrieval performances. Genetic programming (GP) is an evolutionary computational paradigm [1] that is an extension of genetic algorithm and works with a population of individuals. An individual in a population can be any complicated data structure such as linked lists, trees and graphs, etc. CGP is an extension of GP in which several populations are maintained and employed to evolve solutions cooperatively. A population maintained by CGP is called a sub-population and it is responsible for evolving a part of a solution. A complete solution is obtained by combining the partial solutions from all the sub-populations. For the task of object recognition in [9], individuals in sub-populations are composite operators, which are the elements of a composite operator vector. A composite operator is represented by a binary tree whose internal nodes are the pre-specified domain-independent primitive opera-

Proceedings of the Seventh IEEE Workshop on Applications of Computer Vision (WACV/MOTION'05)
0-7695-2271-8/05 $ 20.00 IEEE

tors and leaf nodes are original features. It is a way of combining original features. The CGP approach in this paper follows the algorithm proposed in [9].

The advantage of using a tree structure is that it is powerful in expressing the ways of combining original features. The original features are visual features (e.g., color, texture, structure) extracted from the images. With each element evolved by a sub-population of CGP, a composite operator vector is cooperatively evolved by all the sub-populations. By applying composite operators to the original features extracted from images, composite feature vectors are obtained. These composite feature vectors are fed into a classifier for recognition.

The transformation for feature dimensionality reduction by CGP has two unique characteristics that are suitable for image retrieval: 1) *nonlinearlity*: CGP does not assume any class distribution in the original visual feature space; 2) *explicitness*: unlike kernel trick, CGP yields explicit transformation for dimensionality reduction so that the images can be searched the low-dimensional feature space.

The contributions of this paper are: 1) The coevolutionary genetic programming (CGP) approach in generating composite operator vectors for feature dimensionality reduction and retrieval performance improvement in image databases (Section 2); 2) the comparisons of the experimental results by CGP, MDA and SVM on multiple image databases, so that the advantage of the CGP approach is demonstrated (Section 3).

2 Technical Approach

In this paper, with the purpose of demonstrating the effectiveness of coevolutionary genetic programming approach for image databases, we simply adopt the scenario of training the system by directly providing labelled images in advance.

2.1 Image distribution

By intuition the images belonging to the same class should be close in some sense in the feature space of the image database, i.e., these images form a cluster with a specific shape, which reflects the relevances of the various visual features for the corresponding class. However, we cannot ignore the possibility that the images in the same class (as perceived by the user) may form multiple clusters, i.e., the images with different visual features may belong to the same class.

Based on the above observations, some researchers have proposed the Gaussian mixture model for the image distribution in the feature space of the image databases [10], with each class corresponding to a single mixture component or a variety of mixture components. Sometimes, mixture model assumption for the image distribution of a database may be too strict. In this paper, we do not have to make any model assumption for the image distribution.

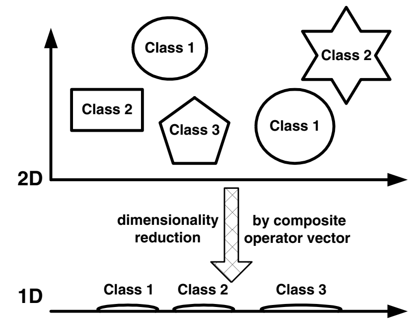


Figure 1: An illustrative example for feature reduction (from 2D to 1D) using composite operation. The original 2D feature space: three classes have different distributions, and Class 1 and Class 2 consists of multiple clusters. The transformed 1D feature space: each of the three classes corresponds to a single Gaussian distribution.

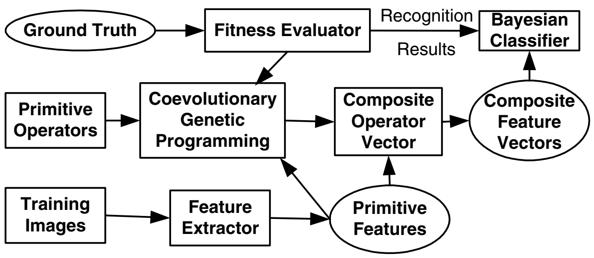
Figure 1 provides an example which illustrates the characteristics of our CGP approach for reducing the feature dimensionality of image databases. The original 2D data belong to three classes, two of which (Class 1 and Class 2) form multiple clusters. Different clusters have different distributions (represented by different shapes in Figure 1). The transformation by the CGP approach yield 1D feature space, in which each of the three classes corresponds to a single Gaussian distribution.

2.2 CGP

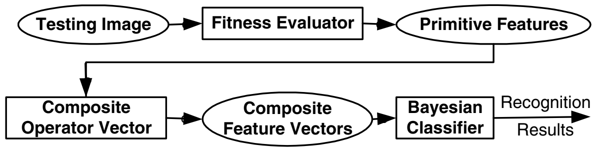
In CGP approach, each synthesized feature is derived by implementing a series of operators on the original visual features. Such operators are called composite operators, which are represented by binary trees with primitive operators as internal nodes and original features as leaf nodes. The goal of these composite operators is to map the original visual feature space to the low-dimensional synthesized feature space, in which the images belonging to the same class form a Gaussian component no matter how these images are distributed in the original visual space.

The search space of all possible composite operators is so large that it is difficult to find good composite operators unless one has a smart search strategy. How to design such a smart search strategy is the task for CGP algorithm. We follow the CGP algorithm proposed in [9], which attempts to improve the performance of object recognition. Figure 2 shows the training and test-

Proceedings of the Seventh IEEE Workshop on Applications of Computer Vision (WACV/MOTION'05)
0-7695-2271-8/05 $ 20.00 IEEE



(a) Training module: learning composite operator vectors and Bayesian classifier.



(b) Testing module: applying learned composite operator vector and Bayesian classifier to a test image.

Figure 2: System diagram for object recognition using co-evolutionary genetic programming.

ing modules of the system. During training, CGP runs on training images and evolves composite operators to obtain composite features. Since a Bayesian classifier is derived from the feature vectors obtained from training images, both the composite operator vector and the classifier are learned by CGP.

2.2.1 The set of primitive operators: A primitive operator takes one or two real numbers, performs a simple operation on them and outputs the result. Table 1 shows the 12 primitive operators being used, where a and b are two real numbers as the inputs to an operator and c is a constant real number stored in an operator.

2.2.2 Fitness measure: the fitness of a composite operator vector is computed in the following way: apply each composite operator of the composite operator vector on the original features of training images to obtain composite feature vectors of training images and feed them to a Bayesian classifier. Note that not all the original features are necessarily used in feature synthesis. Only the original features that appear in the leaf nodes of the composite operator are used to generate composite features. The recognition rate of the classifier is the fitness of the composite operator vector. To evaluate a composite operator evolved in a sub-population, the composite operator is combined with the current best composite operators in other sub-populations to form a complete composite operator vector where composite operator from the ith sub-population occupies the ith position in the vector and the fitness of the vector is defined as the fitness of the composite operator under evaluation. The fitness values of other composite operators in the vector are not affected. When sub-

Table 1: Twelve primitive operators.

Primitive Operator	Description
ADD (a, b)	Add a and b.
ADDC (a, c)	Add constant value c to a.
SUB (a, b)	Subtract b from a.
SUBC (a, c)	Subtract constant value c from a.
MUL (a, b)	Multiply a and b.
MULC (a, c)	Multiply a with constant value c.
DIV (a, b)	Divide a by b.
DIVC (a, c)	Divide a by constant value c.
MAX2 (a, b)	Get the larger of a and b.
MIN2 (a, b)	Get the smaller of a and b.
SQRT (a)	Return $\sqrt{a}$ if a≥0; otherwise, return $-\sqrt{-a}$.
LOG (a)	Return $log(a)$ if a≥0; Rotherwise, return $-log(-a)$.

populations are initially generated, the composite operators in each sub-population are evaluated individually without being combined with composite operators from other sub-populations. In each generation, the composite operators in the first sub-population are evaluated first, then the composite operators in the second sub-population and so on.

2.2.3 Parameters and termination: The key parameters are the number of sub-populations N, the population size M, the number of generations G, the crossover and mutation rates, and the fitness threshold. GP stops whenever it finishes the specified number of generations or the performance of the Bayesian classifier is above the fitness threshold. After termination, CGP selects the best composite operator of each sub-population to form the learned composite operator vector to be used in testing.

2.2.4 Selection, crossover and mutation: The CGP searches through the space of composite operator vectors to generate new composite operator vectors. The search is performed by selection, crossover and mutation operations. The initial sub-populations are randomly generated. Although sub-populations are cooperatively evolved (the fitness of a composite operator in a sub-population is not solely determined by itself, but affected by the composite operators from other sub-populations), selection is performed only on composite operators within a sub-population and crossover is not allowed between two composite operators from different sub-populations.

Selection: The selection operation involves selecting composite operators from the current sub-population. We use tournament selection with size = 5. The higher the fitness value, the more likely the composite operator is selected to survive.

Proceedings of the Seventh IEEE Workshop on Applications of Computer Vision (WACV/MOTION'05)
0-7695-2271-8/05 $ 20.00 IEEE

Crossover: Two composite operators, called parents, are selected on the basis of their fitness values. The higher the fitness value, the more likely the composite operator is selected for crossover. One internal node in each of these two parents is randomly selected, and the two subtrees rooted at these two nodes are exchanged between the parents to generate two new composite operators, called offspring. It is easy to see that the size of one offspring (i.e., the number of nodes in the binary tree representing the offspring) may be greater than both parents if crossover is implemented in such a simple way. To prevent code bloat, we specify a maximum size of a composite operator (called max-operator-size). If the size of one offspring exceeds the max-operator-size, the crossover is performed again. If the size of an offspring still exceeds the max-operator-size after the crossover is performed 10 times, GP selects two subtrees of same size (i.e., the same number nodes) from two parents and swaps the subtrees between the parents. These two subtrees can always be found, since a leaf node can be viewed as a subtree of size 1.

Mutation: To avoid premature convergence, mutation is introduced to randomly change the structure of some composite operators to maintain the diversity of sub-populations. Candidates for mutation are randomly selected and the mutated composite operators replace the old ones in the sub-populations. There are three mutations invoked with equal probability:

1) Randomly select a node of the composite operator and replace the subtree rooted at this node by another randomly generated binary tree.
2) Randomly select a node of the composite operator and replace the primitive operator stored in the node with another primitive operator randomly selected from the primitive operators of the same number of input as the replaced one.
3) Randomly select two subtrees of the composite operator and swap them.

The CGP algorithm is shown in Algorithm 1, and more details can be found in [9].

2.3 Indexing structure

For each class C_i, a Bayesian classifier is generated based on GP-learned composite features. In the low-dimensional feature space, each class is corresponding to a single Gaussian component, which is represented by the mean feature vector and the covariance matrix of feature vectors of this class.

We construct the indexing structure based on the Gaussian components obtained by CGP approach. When a query image comes, the system computes the probabilities that it belongs to those components using the components' parameters (means and covariances),

Algorithm 1 Coevolutionary genetic programming.

1. Randomly generate N sub-populations of size M and evaluate each composite operator in each sub-population individually.
for $gen = 1$ to $generation\_num$ **do**
  for i =1 to N **do**
    1) Implement standard EM algorithm on $\mathcal{X}$.
    2) Keep the best composite operator in sub-population P_i.
    3) Perform crossover on the composite operators in P_i until the crossover rate is satisfied and keep all the offspring from crossover.
    4) Perform mutation on the composite operators in P_i and the offspring from crossover with the probability of mutation rate.
    5) Perform selection on P_i to select some composite operators and combine them with the composite operators from crossover to get a new sub-population P_i' of the same size as P_i.
    6) Evaluate each composite operator C_j in P_i'.
    7) Perform elitism replacement.
    8) Form the current best composite operator vector consisting of the best composite operators from corresponding sub-populations and evaluate it. If its fitness is above the fitness threshold, goto 2.
  end for
end for
2. Select the best composite operator from each sub-population to form the learned composite operator vector and output it.

and executes the search within the component with the highest probability. The indexing structure implies two advantages for image retrieval: 1) the search is carried on only in the subset of the whole image database; 2) the search is carried on in the low-dimensional feature space. Both of these make the search faster; furthermore, our CGP approach guarantees that the retrieval precision is high since the Bayesian classification is good in the low-dimensional feature space.

3 Experiments

To evaluate the effectiveness of the CGP approach for reducing the feature dimensionality of image databases, we implement the CGP algorithm on three different image databases with the purpose to evaluate its effectiveness for different kinds of class distribution. All the images in these three databases are selected from Corel Stock Photo Library.We also compare the results of the CGP approach with other approaches including MDA and SVM.

3.1 Image Databases

1) DB1200: We select 1200 images belonging to 12 classes, and the images in each class have similar visual features, i.e., each class in the feature space forms a cluster. These 12 classes are corresponding to the CDs (series number) in the library including *Mayan & Aztec Ruins* (33000), *horses* (113000), *owls* (75000),

Proceedings of the Seventh IEEE Workshop on Applications of Computer Vision (WACV/MOTION'05)
0-7695-2271-8/05 $ 20.00 IEEE

(1) CD *owls*
Class Bird

(2) CD *hawks and falcons*
Class Bird

(1) CD *tulips*
Class Flower

(2) CD *wildflowers*
Class Flower

(1) CD *tigers*
Class Beast

(2) CD *lions*
Class Beast

Figure 3: DB1500: sample images from the three classes containing multiple CDs.

sunrises & *sunsets* (1000), *North American wildflowers* (127000), *ski scenes* (61000, 62000), *coasts* (5000), *auto racing* (21000), *firework photography* (73000), *divers* & *diving* (156000), *land of the Pyramids* (161000) and *lions* (105000).

2) DB1500: We add other 300 images (from other three CDs in the Library) into DB1200 to obtain DB1500. The three new CDs (series number) are *hawks and falcons* (70000), *tigers* (108000) and *tulips* (258000). Each of these three CDs is merged to one existing CDs in DB1200 to form a class, so that DB1500 still has 12 classes. In these 12 classes, there are three classes each of which consists of two clusters in visual feature space: the CD of *Hawks and Falcons* and the CD of *owls* form the class of *bird*, the class of *tulips* and the class of *North American wildflowers* form the class of *flowers*, and the class of *tigers* and the class of *lions* form the concept of *wild beasts*. Figure 3 shows the sample images of these three classes containing multiple CDs. Obviously, DB1500 is challenging for the linear transformation approach such as MDA, as we will show later in the experiments.

3) DB6600: The 6,600 images are obtained from 66 CDs, which are assigned to 50 classes , i.e., each class may consist of a single CD or multiple CDs. For instance, the CDs *Arabian horses* and *horses* are assigned to the same class since they are both for the same concept of *horse*. On the other hand, even the images within the same CD may form multiple clusters in sense of visual features. For example, the CD of *city of Italy* consists of images for different objects such as building, road, people. For most of the classes, there are many outlier images, each of whose visual features is far away from the cluster(s) its corresponding class contains.

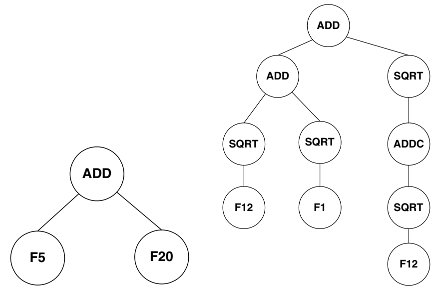


Figure 4: Sample composite operator vectors in DB6600 (the leaf nodes are original visual features): (a) a simple composite operator vector, (b) a complex composite operator vector.

Original visual features: Images are represented by texture features, color features and structure features. The texture features are derived from 16 Gabor filters [11]. We also extract means and standard deviations from the three channels in HSV color space. For structure features, we use the water-filling approach [12] to extract 18 features from each image. Thus, each image is represented by 40 visual features.

3.2 Experiments results

Experimental parameters: For each of the three image databases, we randomly select half of the images as training data and another half as testing data. We implement CGP algorithm on the training images, and the parameters values are: (a) sub-population size: 50; (b) crossover rate: 0.6; (c) number of generation: 50; (d) mutation rate: 0.05; (e) fitness threshold: 1.0; (f) tournament size: 5. For both CGP approach and MDA approach, we reduce the feature dimensionality from 40 to 10, so that it is fair to compare these two approaches.

Figure 4 gives two sample composite operator vectors in DB6600. The complex case in (b) illustrates that the unconventional combinations of the operators exists in CGP, which may help to achieve good classification performance.

Classification performance: Table 2 shows the classification performances of the approaches of CGP, MDA, and SVM. The first two attempt to reduce the feature dimensionality in the *explicit* ways, so that the data in the same class form a single Gaussian component in the lower-dimensional feature space. Thus, we present their Bayesian classification errors in the lower-dimensional (10D) feature space. Since SVM exploits kernel trick instead of providing explicit transformation, we present its classification error in the original visual feature space (40D).

From Table 2, we observe that the classification performance of CGP is better than that of MDA on all

Proceedings of the Seventh IEEE Workshop on Applications of Computer Vision (WACV/MOTION'05)
0-7695-2271-8/05 $ 20.00 IEEE

Table 2: Classification errors: the comparison of the three different approaches on different databases.

Database	**CGP** (10D)	**MDA** (10D)	**SVM** (40D)
DB1200	14.8%	25.5%	12.6%
DB1500	16.5%	31.7%	29.6%
DB6600	51.8%	73.3%	52.6%

of the three databases. It is easy to understand the significant advantage of CGP over MDA on DB1500 and DB6600, both of which contain some classes consists of multiple clusters in the original visual feature space. Therefore, the linear approach of MDA cannot deal with them well. Although each of the class in DB1200 only corresponds to a single cluster in the original visual feature space, CGP still outperforms MDA (14.8% versus 12.6%) due to the reason that the distribution of these clusters may not be Gaussian while MDA assumes the cluster distribution as Gaussian and CGP does not make such assumption.

We now compare CGP and SVM, which have similar classification errors on DB1200 and DB6600. However, the error (16.5%) by CGP is significantly superior to that (29.6%) by SVM on DB1500, which implies that CGP is more suitable to deal with the class consisted of multiple clusters in the original feature space. With many outliers in DB6600, such advantage of CGP is not so obvious as in the case of DB1500 (CGP: 51.8% versus MDA: 52.6%), and neither CGP nor SVM can yield classification performance which is as good as those on DB1500.

Retrieval performance: After reducing the feature space, we construct the indexing structure based on the Gaussian components obtained by CGP approach or MDA approach. We use each of the image the database as query, and obtain average retrieval precisions. Figure 5 shows the retrieval-precision curves on DB6600 by CGP and MDA. We observe that CGP yields better retrieval results than MDA.

From above experiments, we conclude that (a) the feature synthesizing by CGP has obvious advantage over that by MDA in the sense of both classification and retrieval. (b) The classification performance in the synthesized feature space by CGP is comparable to or even superior to that in the original feature space by SVM.

4 Conclusions

This paper presents a coevolutionary genetic programming (CGP) for feature dimensionality reduction. The two characteristics of the transformation by CGP, i.e., nonlinearity and explicitness, make it suitable for image retrieval: (a) the nonlinearity yields good classification performance without any class distribution model assumption, and (b) the explicitness achieves the image search in the low-dimensional feature space. Experimental results on multiple image databases with different types of distributions have demonstrated the effectiveness of improving image retrieval performance by the CGP approach.

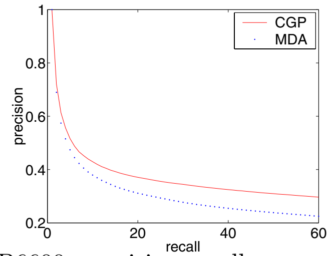


Figure 5: DB6600: precision-recall curves: CGP versus MDA.

References

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

[2] B. Schölkopf and A.J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*, the MIT Press, 2001.

[3] G. Baudat and F. Anouar, "Generlized discriminant analysis using a kernel approach," *Neural Computation*, vol. 12, pp. 2385–2404, 2000.

[4] D. Swets and J. Weng, "Hierarchical discriminant analysis for image retrieval," *IEEE Trans. PAMI*, vol. 21, no. 5, pp. 386–401, May 1999.

[5] T. Hastie and R. Tibshirani, "Discriminant adaptive nearest neighbor classification," *IEEE Trans. PAMI*, vol. 18, no. 6, pp. 607–616, 1996.

[6] P. Wu, B. S. Manjunath, and H. D. Shin, "Dimensionality reduction for image search and retrieval," *ICIP*, 2000.

[7] Z. Su, S. Li, and H. J. Zhang, "Extraction of feature subspaces for content-based retrieval using relevance feedback," *Proc. ACM MM*, pp. 98–106, 2001.

[8] J. Koza, *Genetic Programming II: Automatic Discovery of Reusable Programs*, MIT Press, 1994.

[9] Y. Lin and B. Bhanu, "Evolutionary feature synthesis for object recognition," *IEEE Trans. SMC, Part B*, to appear.

[10] N. Vasconcelos, *Bayesian Models for Visual Information Retrieval*, Ph.D thesis, MIT, 2000.

[11] B. S. Manjunath and W. Y. Ma, "Texture feature for browsing and retrieval of image data," *IEEE Trans. PAMI*, vol. 18, no. 8, pp. 837–842, August 1996.

[12] X.S. Zhou, Y. Rui, and T.S. Huang, *Exploration of Visual Data*, Kluwer Academic Publishers, 2003.

Proceedings of the Seventh IEEE Workshop on Applications of Computer Vision (WACV/MOTION'05)
0-7695-2271-8/05 $ 20.00 IEEE