The Customized-Queries Approach to CBIR Using EM

J. G. Dy, C. E. Brodley, A. Kak, C. Shyu
School of Electrical and Computer Engineering
Purdue University
West Lafayette, IN 47907
{*dy, brodley, kak, chiren*}*@ecn.purdue.edu*

L. S. Broderick
Department of Radiology
University of Wisconsin Hospital
Madison, WI 53792
lsbroderick@facstaff.wisc.edu

Abstract

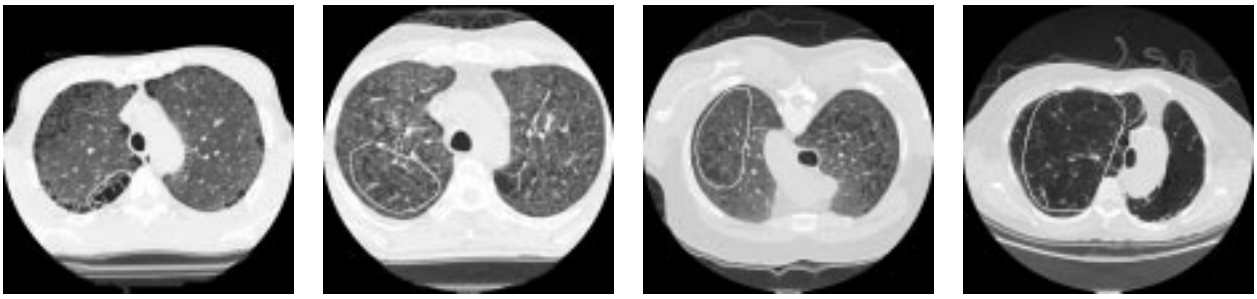
This paper makes two contributions. The first contribution is an approach called the "customized-queries" approach (CQA) to content-based image retrieval. The second is an algorithm called FSSEM that performs feature selection and clustering simultaneously. The customized-queries approach first classifies a query using the features that best differentiate the major classes and then customizes the query to that class by using the features that best distinguish the images within the chosen major class. This approach is motivated by the observation that the features that are most effective in discriminating among images from different classes may not be the most effective for retrieval of visually similar images within a class. This occurs for domains in which not all pairs of images within one class have equivalent visual similarity, i.e. subclasses exists. Because we are not given subclass labels, we must simultaneously find the features that best discriminate the subclasses and at the same time find these subclasses. We use FSSEM to find these features. We apply this approach to content-based retrieval of high-resolution tomographic images of patients with lung disease and show that this approach radically improves the retrieval precision over the traditional approach that performs retrieval using a single feature vector.

1. Introduction

An effective approach to content-based image retrieval (CBIR) represents each image in the database by a vector of feature values [5, 7, 11]. During retrieval, the query image's feature vector is compared to the database vectors via the chosen indexing scheme. For such approaches, the choice of features to include in the image characterization is a critical factor in their ability to achieve high retrieval precision. For many feature selection problems, including image characterization for CBIR, a human defines the features that are potentially useful and then a subset is chosen from this set using an automated feature selection algorithm. Typically feature selection is performed with respect to classification information – i.e., each data point (image in the database) is labeled with a class name. However, complete data classification can be labor intensive when we have a huge collection of data. In the absence of class label information, we must simultaneously find the features that best discriminate the classes and at the same time find these classes. We resort to unsupervised clustering, which allows us to categorize data based on its structure. The clustering problem is made more difficult when we need to select the best features simultaneously. To find the features that maximize our performance criterion (e.g. retrieval precision), we need the clusters to be defined. Moreover, to perform unsupervised clustering we need the features or the variables which span the space we are trying to cluster.

In this paper, we introduce a method for performing clustering and feature selection simultaneously using the expectation-maximization (EM) algorithm [3]. We apply this method to a CBIR domain in which we have partial class information – for each image we know the "major" class, but images within each class can vary widely with respect to visual similarity. Our "customized-queries" approach (CQA) to indexing and retrieval in such domains was introduced in an earlier paper [4]. The approach first classifies a query using the features that best differentiate the major classes and then customizes the query to that class by using the features that best distinguish the images within the chosen major class. This approach was motivated by the observation that the features that are most effective in discriminating among images from different classes may not be the most effective for retrieval of visually similar images within a class. This occurs for domains in which not all pairs of images within a given class have equivalent visual similarity. For example in the domain of transportation classification, the features that best distinguish airplanes from cars differ from the features that best distinguish commercial jets and stealth fighters. Such domains are appropriate candidates for our approach.

We have applied and evaluated this approach within ASSERT [13], a CBIR system for medical images. Our database of interest is high-resolution computed tomographic (HRCT) images of the lungs. Each image in our



(a) (b) (c) (d)

Figure 1. (a) P-Emphysema image. (b) C-Emphysema image in subclass 1. (c) C-Emphysema image in subclass 1. (d) C-Emphysema image in subclass 2.

database has a disease class label. The customized-queries approach is well suited to this domain because although a given set of features may be ideal for the disease categorization of a query image, those features may not always retrieve the images that are most similar to the query image. A query image may differ visually from other images within the same disease class on account of the severity of disease and other such factors. Figure 1 illustrates this point. Notice that within the class Centrilobular Emphysema (C-Emphysema), Figure 1d is visually dissimilar to Figures 1b and 1c. A feature that distinguishes Paraseptal Emphysema (P-Emphysema) from C-Emphysema is the distance of the "pathology bearing region (PBR)" [13] from the boundary of the lung, whereas the features that best discriminate the images within class C-Emphysema are those that measure the gray level intensity of the PBR. The PBR is the region marked by the physician as the region of interest or diseased region. The PBR's are encircled by a white boundary as shown in Figure 1.

2. Related Work

Forming a hierarchy of features for retrieval and storage has been explored by other researchers, but their end goals for doing so differ from ours. For example in the FourEyes system [10], highly structured objects in images, such as buildings and trees, are represented hierarchically to facilitate structural comparisons with a query image. In "Texture features and learning similarity" by Ma and Manjunath [9], they used a hybrid neural network algorithm to learn similarity by clustering in the texture feature space and then fine tuning the clusters using supervised learning. Their approach builds a hybrid neural network classifier that is applied during retrieval to classify the query as one of the given classes. Then they select the n most similar images within that class cluster using Euclidean distance. Note that the same feature set is used both for classification and for retrieval after classification. Our approach differs in that we do not require the feature sets for classification and retrieval to be the same. Chen and Bouman [2] developed an approach that organizes images in "similarity pyramids" by grouping images with the closest distances, as defined by an L_1 norm distance metric, together. They used an agglomerative (bottom-up) clustering algorithm to build the pyramid. The resulting organization is used for indexing and browsing purposes. In contrast, we group images according to disease classes and subclasses in order to emulate how expert radiologists would categorize them. Furthermore, we use different feature sets for comparing similarity at each level and for each class. Chen and Bouman's approach used the same feature set and similarity metric throughout the organization of their hierarchy or "pyramid".

3. The Customized-Queries Approach

The traditional approach to content-based image retrieval tries to find one set of features that distinguishes the different classes well and at the same time finds the n best images for retrieval. The customized-queries approach, on the other hand, breaks this task into two levels: 1) find the set of features that distinguishes the different major classes, and 2) "customize" the query by using the specialized set of features in the query's class to obtain the best n images to retrieve. To implement this approach we must first learn a classifier, thereby defining the "Level 1" features. For each class we must then learn the features (we call "Level 2" features) that best cluster the images into visually similar clusters.

The customized-queries approach is not to be confused with a decision tree [12]. A decision tree is a supervised learning (i.e. training examples with class labels are provided) algorithm for classifying instances into classes. CQA is an approach to automate the customization of the query (by modifying the feature representation based on an estimate of the query's major class) for retrieval.

In Section 3.1, we describe the retrieval procedure. In

Sections 3.2 and 3.3 we present our approaches to finding the Level 1 and Level 2 features.

3.1. The retrieval procedure

Using the Level 1 features, we first classify the query image using a one nearest neighbor (1-NN) classifier. We used 1-NN because in a comparison to 2, 3, 4, 5-NN and decision trees, 1-NN yielded the lowest classification error. We measured the error using a ten-fold cross-validation, which randomly partitions the dataset into ten mutually exclusive subsets. Classification error is computed with each partition (or fold) as the test set and the rest as the training set. This is repeated ten times, one for each fold. The final estimate for the classification error is the average of these estimates.

The 1-NN predicts a class C_i for the query image. The system then uses the Level 2 features associated with class C_i to retrieve the n most similar images as defined by Euclidean distance to the query image (i.e. using n-NN). In the next two subsections, we will answer the following questions:

1. How does one determine the features for discriminating among the major classes?
2. Within each class, how does one determine the discrimination boundaries among visually similar subclasses? and
3. How does one customize the features within each class?

3.2. Customizing the Level 1 features

We treat each level as a separate classification and feature selection problem. On the first level we use the given image classes as our categories. In our domain of interest, this corresponds to the disease pathology assigned to each image. These pathology class labels are confirmed diagnoses obtained from medical records, hence we can consider these as ground truth labels.

To find the Level 1 features, we first extract all features from the query image. We call these the base features, $F_B = \{F_1, F_2, ..., F_N\}$. Then we use Feature Subset Selection (FSS) [8] wrapped around the Instance-Based (IB) classifier (a 1-nearest-neighbor classifier) using MLC^{++} [1] (we chose the forward search direction option) to find the subset of features from F_B that best discriminate the Level 1 classes. Other inducers such as decision trees or neural networks could be used in place of IB. FSS is wrapped around the IB classifier because this is the inducer (classifier) used to classify the query at Level 1. This means that we are using Euclidean distance as our dissimilarity metric and that we are using nearest neighbor to identify the class to which our query belongs. FSS is a greedy search algorithm that tries to find the best set of features for delineating the different classes [8]. FSS adds the feature that when combined with the current chosen set yields the largest improvement in classification accuracy of the classifier. To estimate the classification error, FSS uses ten-fold cross-validation. For our dataset using the FSS features gave a classification accuracy of $93.33\% \pm 0.70\%$ which was slightly better than the $92.67\% \pm 0.79\%$ classification accuracy obtained using all of the features. FSS chose eleven Level 1 features, which is a substantial reduction from using all of 125 possible features (a complete listing of these features is given in [4]).

[1] Available at http://www.sgi.com/Technology/mlc.

3.3. Customizing the Level 2 features and clustering the Level 2 classes

On the first level we used the different disease classes assigned to the images in our database to perform feature selection. At the second level we are not given class labels, requiring us to use unsupervised clustering to group the images within each disease class. In addition to learning the clusters, we also need to find the features that yield the best clustering and the optimal number of clusters, k. Hence, we need to simultaneously find k, the clusters and the feature set.

To select the features, we experimented with both the $trace(S_w)/trace(S_b)$ and the $trace(S_w^{-1}S_b)$ criteria [6]. S_w is the within-class scatter matrix and S_b is the between class scatter matrix, and they are defined as follows:

$$\begin{aligned} S_w &= \sum_{i=1}^{k} \pi_i E\{(X-\mu_i)(X-\mu_i)^T|\omega_i\} = \sum_{i=1}^{k} \pi_i \Sigma_i \\ S_b &= \sum_{i=1}^{k} \pi_i (\mu_i - M_o)(\mu_i - M_o)^T \\ M_o &= E\{X\} = \sum_{i=1}^{k} \pi_i \mu_i \end{aligned}$$

where π_i is the probability of class ω_i, X is a random feature vector representing the image, μ_i is the mean vector of class ω_i, M_o is the total mean across all data points or images in the database, Σ_i is the covariance matrix of class ω_i, and $E\{\cdot\}$ is the expected value operator. S_w measures how scattered the samples are from their cluster means, and S_b measures how scattered the cluster means are from the total mean [6]. We would like to select features that discriminate our clusters best, i.e. we would like the distance between each pair of samples in a particular cluster to be as close together as possible and the cluster means to be as far apart as possible with respect to the chosen similarity metric.

In Section 3.3.1 we describe our first ad hoc approach to performing feature selection and clustering simultaneously. In Section 3.3.2 we present an approach based on the EM algorithm. And finally in Section 4 we present experiments comparing these two approaches to the traditional method, which performs retrieval using a single query vector.

3.3.1. Our previous approach. Our previous approach first considers each feature one at a time to assess its ability to cluster the data into k classes using the k-means clustering algorithm [6]. We then sort the features

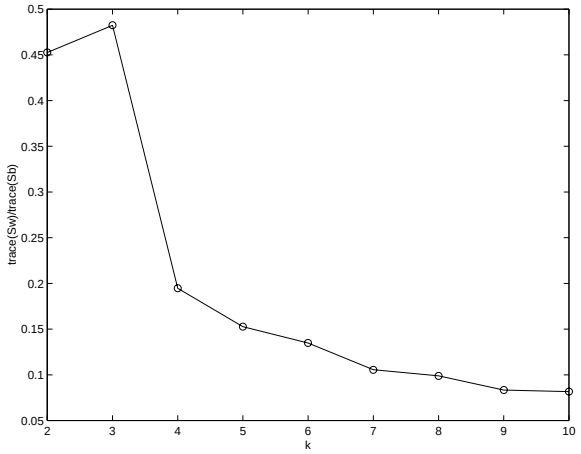


Figure 2. The $trace(S_w)/trace(S_b)$ criterion value with respect to k for C-Emphysema.

in increasing order based on the value of our criterion for the clusters generated by the feature. Next, we pick the best n features that are not pairwise correlated by more than 50%. We need to check for correlation because many of our base features are highly correlated. We chose n to be the number of Level 1 features chosen by FSS as described in Section 3.2. We then cluster the data using k-means applied to the n selected features and then compute the $trace(S_w)/trace(S_b)$ criterion. Note that other criteria for evaluating clustering performance could also be used [6]. We repeat this procedure for k ranging from two to ten. Since the criterion function is a decreasing function with respect to k [6], we choose k to be the smallest k for which the decrease in criterion value is no longer as significant as the previous decrease. For example Figure 2 shows the $trace(S_w)/trace(S_b)$ criterion value with respect to k for C-Emphysema, resulting in a choice of $k = 5$. Once we establish the clusters obtained by the selected n features and the chosen k, we define each of the k clusters to be a distinct subclass. Using the generated subclass labels, we apply FSS to customize the features to be our Level 2 features for that class. We chose to customize the features rather than retain the features used for clustering in an attempt to improve our ability to discriminate among the subclasses and to reduce the number of features required [4].

3.3.2. Feature subset selection and clustering using EM. A weak point of our previous approach is that it considers the clustering ability of the features individually. However to select the best feature subset for clustering, it is important to assess the clustering ability of the features jointly. For example, weight and height are good features for discriminating people who are overweight from those who are not. But individually, height or weight does not provide as high discrimination power.

Our new method is inspired by the wrapper approach for feature subset selection [8]. Instead of using feature subset selection wrapped around a classifier, we wrap it around a clustering algorithm. The basic idea of our approach is to search through feature subset space, evaluating each subset, F_t, by first clustering in space F_t using the EM [3] algorithm and then evaluating the resulting cluster using our chosen clustering criterion. The result of this search is the feature subset that optimizes our criterion function. Because there are 2^n feature subsets, where n is the number of available features, exhaustive search is impossible. To search the features, sequential forward, backward elimination or forward-backward search can be used [6]. In this paper we evaluate sequential forward selection. This is a greedy search algorithm that adds one feature at a time. This method adds the feature that when combined with the current chosen set yields the largest improvement to our separability criterion. We use the $trace(S_w^{-1}S_b)$ as our criterion because it is invariant under any nonsingular linear transformation unlike the $trace(S_b)/trace(S_w)$ criterion [6]. Transformation invariance means that once m features are chosen, any nonsingular linear transformation on these features does not change the criterion value. To cluster our data, we currently choose k, the number of clusters, using the procedure described in Section 3.3.1. In the future we will automate our EM clustering algorithm to optimize k simultaneously [1].

In the remainder of this section, we describe our application of the EM algorithm and a modification to the basic approach outlined above that speeds up the runtime.

EM Clustering. We treat our data (the image vectors in our database) as a d-dimensional random vector and then model its density as a Gaussian mixture of the following form:

$$f(X_i|\Phi) = \sum_{j=1}^{k} \pi_j f_j(X_i|\theta_j)$$

where $f_j(X_i|\theta_j) = \frac{1}{(2\pi)^{\frac{d}{2}}|\Sigma_j|^{\frac{1}{2}}} e^{-\frac{1}{2}(X_i-\mu_j)^T\Sigma_j^{-1}(X_i-\mu_j)}$, is the probability density function for class j, $\theta_j = (\mu_j, \Sigma_j)$ is the set of parameters for the density function $f_j(X_i|\theta_j)$, μ_j is the mean of class j, Σ_j is the covariance matrix of class j, π_j is the mixing proportion of class j (subject to the following constraints: $\pi_j \geq 0$ and $\sum_{j=1}^{k} \pi_j = 1$), k is the number of clusters, X_i is a d-dimensional random data vector, $\Phi = (\pi_1, \pi_2, \cdots \pi_k, \theta_1, \theta_2, \cdots \theta_k)$ is the set of all parameters, and $f(X_i|\Phi)$ is the probability density function of our observed data point X_i given the parameters Φ. Here, we made a standard assumption that our data is Gaussian. However, future work will include testing the performance of other distributions as well.

The X_i's are the data vectors we are trying to cluster. To cluster X_i, we need to estimate the parameters, Φ. One method for estimating Φ is to find Φ which maximizes the log-likelihood, $log f(X|\Phi)$. A popular iterative optimization algorithm to solving the maximum likelihood estimate for missing data problems is the expectation-maximization

(EM) algorithm [3]. In the clustering task, the missing data is knowing to which cluster each X_i belongs. EM provides us with "soft-clustering" information, i.e. a data point X_i can belong to more than one cluster weighted by the probability of that data to belong to a cluster. Going through the derivation, we obtain the following EM update equations [15]:

$$E[z_{ij}] = p(z_{ij} = 1|X, \Phi^{(t)}) = \frac{p_j(X_i|\Phi_j^{(t)})\,\pi_j^{(t)}}{\sum_{s=1}^{k} p_s(X_i|\Phi_s^{(t)})\,\pi_s^{(t)}}$$

$$\pi_j^{(t+1)} = \tfrac{1}{N} \sum_{i=1}^{N} E[z_{ij}]$$

$$\mu_j^{(t+1)} = \tfrac{1}{N\pi_j^{(t+1)}} \sum_{i=1}^{N} E[z_{ij}]\, X_i$$

$$\Sigma_j^{(t+1)} = \tfrac{1}{N\pi_j^{(t+1)}} \sum_{i=1}^{N} E[z_{ij}]\,(X_i - \mu_j^{(t+1)})(X_i - \mu_j^{(t+1)})^T$$

where $E[z_{ij}]$ is the expected value of the probability that the data X_i belongs to cluster j and $\sum_{i=1}^{N} E[z_{ij}]$ is the estimated number of data points in class j. In the EM algorithm, we start with an initial estimate of our parameters, Φ, and then iterate using the update equations until convergence.

The EM algorithm can get stuck at a local maximum, hence the initialization values are important. We used $r = 10$ random restarts on k-means and pick the one with the highest maximum likelihood to initialize the parameters [14]. We then iterate until convergence (likelihood does not change by 0.0001) or up to n iterations whichever comes first for each search step. We experimented with $n = 20$ and $n = 50$ and and found that they yielded the same features for our database. We limit the number of iterations because EM converges asymptotically, i.e. convergence is very slow when you are near the maximum. Moreover we often do not require many iterations, because initializing with k-means starts us at a high point on the hill of the space we are trying to optimize.

4. Experimental Results

Our current database consists of 312 HRCT lung images from 62 patients. These images yield 615 pathology bearing regions (PBR) [13]. PBR's are local image regions marked by the physician as pathological. A single image may have several PBR's and these PBR's may have different diagnoses. Throughout the experiment we considered each PBR as a data point, i.e. a single image with three PBR's gives us three data points. We used 125 implemented features from ASSERT as our base features [13]. These features include measures of geometric properties (centroid, area, distance from boundary), gray level mean, standard deviation, gray level histogram, area histogram, texture (using co-occurrence matrix), and edginess measures of the local pathology bearing regions and of the global image [13]. In selecting the Level 2 features, we excluded the PBR location and size features that may capture systematic effects of the PBR markings made by our physician. Although these features cluster the images well, the resulting clustering does not group the PBR's according to visual similarity. This cuts down our initial feature space to 110 features.

Table 1. The disease class distribution, query distribution and the number of clusters, k.

Disease Class	Database Frequency	Query Frequency	k
C-Emphysema (CE)	314	18	5
P-Emphysema (PE)	54	3	4
IPF	51	2	3
EG	57	1	4
Sarcoid (Sar.)	16	1	5
Aspergillus (Asper.)	12	1	5
Bronchiectasis (Bron.)	14	1	2

Our experiments compare using just the Level 1 features for retrieval (the traditional approach) to our customized-queries approach, which first classifies the query image using the Level 1 features and then uses the Level 2 features from the resulting class for retrieval. We also compare the performance among the different methods for customizing the Level 2 features. Specifically, we compare the results of the following three methods:

Method 1: The traditional approach: Level 1 features are used for retrieval across the entire database using Euclidean distance.

Method 2: The customized-queries approach: Level 1 features classify the query image as one of the Level 1 classes. Then retrieve the nearest neighbors within that class as measured by Euclidean distance over the corresponding Level 2 features. The Level 2 features are obtained using our previous approach described in Section 3.3.1.

Method 3: The customized-queries approach: Identical to Method 2 except that the Level 2 features are obtained using feature subset selection wrapped around EM.

In assessing the performance of Methods 2 and 3, we assumed an ideal classifier to classify the query as a Level 1 class. We did this to isolate the effect of using the appropriate Level 2 features in retrieving the images, i.e. the utility of customizing a query. This assumption is not too limiting since the classification accuracy we obtained from a ten-fold cross-validation applied to a 1-NN classifier using the Level 1 features is $93.33\% \pm 0.70\%$.

To determine which method is best, the radiologist in our team was asked to evaluate the retrieval results for the three methods. Throughout the test, the radiologist was not

Table 2. Results.

Disease Class	Method 1					Method 2					Method 3				
	SA	A	NS	D	SD	SA	A	NS	D	SD	SA	A	NS	D	SD
CE	28	9	5	2	28	58	5	8	0	1	69	2	1	0	0
PE	0	0	4	0	8	11	0	1	0	0	10	0	1	0	1
IPF	5	0	0	0	3	4	0	0	1	3	3	2	2	0	1
EG	0	0	0	0	4	0	4	0	0	0	4	0	0	0	0
Sar.	0	0	0	0	4	0	0	0	0	4	0	0	0	0	4
Asper.	0	0	0	0	4	4	0	0	0	0	3	1	0	0	0
Bron.	0	0	0	0	4	3	0	0	0	1	3	1	0	0	0
total	33	9	9	2	55	80	9	9	1	9	92	6	4	0	6
score	-37					150					178				
ret. precision	38.89%					82.41%					90.74%				

informed as to which method produced the retrieved images. We used eighteen images selected randomly from the C-Emphysema class, three from P-Emphysema, two from IPF and one from each of EG, Bronchiectasis, Sarcoid and Aspergillus as test query images. We chose eighteen from C-Emphysema because it is the largest class in our collection (51% of our database is of class C-Emphysema). The disease class distribution and the k chosen for each class is shown in Table 1. The four images ranked most similar to the query image were retrieved for each method. Note that all images of the query patient are excluded from the search. The user can choose from five responses: strongly-agree (SA), agree (A), not sure (NS), disagree (D) and strongly-disagree (SD) for each retrieved image. To measure the performance of each method, the following scoring system was used: 2 for SA, 1 for A, 0 for NS, -1 for D and -2 for SD. The results are summarized in Table 2.

Method 1 received a total of -37 points, Method 2 garnered 150 points and Method 3 received 178 points. If SA and A are considered as positive retrievals and the rest as negative retrievals, Method 1 resulted in 38.89% retrieval precision. Method 2 resulted in 82.41% precision and Method 3 resulted in 90.74% precision. Notice that the Method 1 precision is not the same as the accuracy obtained for the Level 1 classifier because there were many cases where the radiologist did not mark SA or A even though the retrieved images have the same diagnosis as the query image. Note also that we cannot measure recall because we do not have the subclass labels.

From these results, we can see that customized queries (Methods 2 and 3) dramatically improve retrieval precision compared to the traditional approach (Method 1) for this domain. Moreover, our new method (Method 3) for feature selection and clustering performed better than our previous method (Method 2).

It is interesting to note that the features selected by Method 3 for the class C-Emphysema (the largest class in our database) are two features for uniformity of energy, one for histogram, one for texture correlation and one for ratio of PBR area to lung area. These are the same features that our domain experts chose for describing the structure of C-Emphysema. Method 2 on the other hand picked only gray level mean features. Method 2 resulted in only 87.5% retrieval precision for the C-Emphysema class, whereas Method 3 resulted in 98.61%. The features selected by Method 2 were not able to capture the structure of the C-Emphysema class. Furthermore, note that the Level 2 features chosen by Methods 2 and 3 are different for each disease class and different from the Level 1 features as shown in Table 3.

Our results show that it is not enough to retrieve images based on just the disease class. In addition, we need to find the best image within that class on the basis of visual similarity. Obtaining better precision is very important in medical image retrieval, because once the best match is found doctors can compare the medical history of the two patients and hopefully be able to use the same diagnosis and treatment. Moreover, the feature set that best discriminates the disease classes is different from the feature set that best discriminates between visually similar subclasses within each disease class. Hence, there exists a need for customized queries.

5. Conclusion

We introduced a customized-queries approach to CBIR that first classifies a query using the features that best differentiate the Level 1 classes and then customizes the query to that class by using the features that best distinguish the Level 2 classes within the chosen Level 1 class. Our approach was motivated by the observation that the image features that work best to discriminate among different classes are different from the features needed to retrieve the most visually similar images within each class. For the domain of HRCT images of the lung, our results show that given a correct Level 1 classification the customized-queries approach yields 90.74% retrieval precision, whereas the traditional single feature vector approach yields only 38.89% retrieval

Table 3. The Level 1 and Level 2 features.

Level 1 Features	F1, F3, F8-9, F29, F32, F134, F136, F155, F205, F229	
Level 2 Features	Method 2	Method 3
Disease Class	Feature Set	Feature Set
Alveolar Proteinosis	F22, F25, F177	F13, F247
Aspergillus	F6	F175, F179
Bronchiectasis	F20	F5, F165, F168, F175, F179-180, F204, F206-207, F217-218, F243
Bronchiolitis Obliterans	F20	F6, F11, F17, F19, F199, F205, F256
Centrilobular Emphysema	F1, F35, F194	F175, F203, F217-218, F239
EG	F1, F167	F7, F201
Hemorrhage	F1	F202
IPF	F72, F167, F220	F4, F173, F227, F229, F231
Panacinar	F44, F106, F19	F167
Paraseptal Emphysema	F1, F14, F193, F244	F8, F232
PCP	F20	F4, F15, F17, F170, F203, F239
Sarcoid	F94	F194

precision. Moreover, the traditional approach requires that one finds a set of features that both correctly classify the query and retrieve visually similar images. Whereas the customized-queries approach divides the problem into two parts: First optimize the features for Level 1 classification, then optimize the features for retrieving visually similar images within that class.

To select our Level 2 features, we introduced a general method for performing feature selection and clustering simultaneously. Our experiments showed that it outperformed our previous ad-hoc approach. Future work will investigate how this new method performs for other domains, compare the performance of using different criterion functions, test other distributions (aside from Gaussian) to model the data and incorporate the determination of the number of clusters, k, inside EM clustering.

Acknowledgments

We thank Mark Flick for his helpful comments and Dr. Alex Aisen for helping build our database. This research is supported by NSF Grant No. IRI9711535, and NIH Grant No. 1 R01 LM06543-01A1.

References

[1] C. A. B. Bouman, M. Shapiro, G. W. Cook, C. B. Atkins, and H. Cheng. Cluster: An unsupervised algorithm for modeling gaussian mixtures. *http://dynamo.ecn.purdue.edu/∼bouman/software/cluster*, October 1998.

[2] J. Chen, C. A. Bouman, and J. C. Dalton. Similarity pyramids for browsing and organization of large image databases. *SPIE Human Vision and Electronic Imaging III*, 3299, January 1998.

[3] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *J. Royal Statistical Society, Ser. B*, 39(1):1–38, 1977.

[4] J. G. Dy, C. E. Brodley, A. Kak, C. R. Shyu, and L. Broderick. The customized-queries approach to CBIR. *SPIE Storage and Retrieval for Image and Video Databases VII*, 3656:22–32, January 1999.

[5] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, B. Dom, Q. Huang, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker. Query by image and video content: The qbic system. *IEEE Computer*, 28(9):23–32, 1995.

[6] K. Fukunaga. *Introduction to Statistical Pattern Recognition, Second Edition*. Academic Press, Inc., 1990.

[7] P. M. Kelly, T. M. Cannon, and D. R. Hush. Query by image example: The CANDID approach. *SPIE Storage and Retrieval for Image and Video Databases III*, 2420:238–248, 1995.

[8] R. Kohavi and G. H. John. Wrappers for feature subset selection. *Artificial Intelligence Journal*, 97(1-2):273–324, 1997.

[9] W. Y. Ma and B. S. Manjunath. Texture features and learning similarity. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 425–430, 1996.

[10] T. P. Minka and R. W. Picard. Interactive learning using a 'society of models'. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 447–452, 1996.

[11] A. Pentland, R. W. Picard, and S. Sclaroff. Photobook: Tools for content-based manipulation of image databases. *SPIE Storage and Retrieval for Image and Video Databases II*, 2185, February 1994.

[12] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1:81–106, 1986.

[13] C. R. Shyu, C. E. Brodley, A. Kak, A. Kosaka, A. Aisen, and L. Broderick. ASSERT, a physician in the loop content-based image retrieval system for hrct image databases. *to appear in Computer Vision and Image Understanding*.

[14] P. Smyth. Clustering using Monte Carlo cross-validation. *The Second International Conference on Knowledge Discovery and Data Mining*, pages 126–133, 1996.

[15] J. H. Wolfe. Pattern clustering by multivariate mixture analysis. *Multivariate Behavioral Research*, 5(3):101–116, 1970.