CONTENT-BASED IMAGE RETRIEVAL: A SHORT-TERM AND LONG-TERM LEARNING APPROACH

Samuel Barrett

Stevens Institute of Technology
Computer Science Department
Hoboken, NJ 07030

ABSTRACT

The ability to search through images based on their content rather than on their low-level features is becoming more important as the number of available images grows. To deal with this problem, we propose a content-based image retrieval system that uses a new short learning technique combined with a novel long term learning approach. The short term learning technique splits an image into several regions and then applies fuzzy support vector machine learning to these regions. The long term learning approach attempts to adaptively learn the semantic concepts represented in the images using relevance feedback and semantic clustering. One important advantage of the proposed system is its ability to scale to large image databases. Experiments comparing the proposed system with several other state-of-the-art techniques show the effectiveness of the proposed system.

***Index Terms*—** Content-based image retrieval, relevance feedback, region-based image retrieval, high-level semantics

1. INTRODUCTION

The ability to search through images continues to become more important as the number of available images rises drastically. However, it is impractical for humans to label every image available and dictate which images are similar to which other images. Therefore, there is a drive to teach computers to learn how images are related while minimizing human work. The normal approach is to use the low level features of the images like color, textures, and edges to try to predict the similarities between images. Unfortunately, there is a semantic gap between what the low level features show and the high level features that represent what a human understands from the image. For example, the low level features in an image may represent that there is a brown object in the upper-left corner of an image, while the desired high level features would represent that there is a dog shown in the image.

Content-based image retrieval (CBIR) attempts to bridge this semantic gap with techniques falling into two categories: short term learning tries to learn which images are relevant to the user's query over the course of a single query, and long term learning attempts to learn the connections between images over the course of many queries. Both techniques can be very effective, and it is often best to combine both techniques. Relevance feedback, user labeling of images that have been returned as either relevant or irrelevant to the query image, provides information to the computer that enables learning the semantic information contained in the images. However, users are unwilling to label too many images, so the amount of relevance feedback is limited. Therefore, CBIR systems use long term learning techniques to store the semantic information learned from previous queries and apply that knowledge to future searches.

The explanation of the paper's organization follows. Section 2 contains a discussion of previous research in content-based image retrieval. The proposed short term learning learning algorithm is introduced in Section 3 and its results are compared to other short term learning systems in Section 4. In Section 5, a new long term learning algorithm is proposed and this algorithm is tested and compared to other long term learning systems in Section 6. Finally, Section 7 draws conclusions from the experiments.

2. RELATED WORK

Content-based image retrieval has become a topic of interest in recent years, and there has been some substantial research in the area. Learning over a single query is called short term learning. For short term learning, research has found that support vector machines (SVMs) are effective at learning the boundary separating relevant and irrelevant images using relevance feedback, where images are returned several times for the user to label as relevant or irrelevant to the search. A popular alternative to SVMs is Bayesian networks, which use a probabilistic approach for the likelihood that an image is relevant to the query [1], [2]. As the CBIR system returns more images and the user labels them, the CBIR system learns what the user is searching for and it returns more relevant images at each iteration.

Thanks to the NSF REU program for funding.

Wu and Yap have improved the effectiveness of SVM learning by propagating the labels from images labeled by the user to similar images [3]. The images to be labeled automatically are chosen by their proximity to the images that have already been labeled. Intuitively, these labels are not as accurate as those provided by the user, so the labels are assigned a weight depending on the predicted accuracy of the label. These new labels are called pseudo-labels and are used with the user's original labels to train a fuzzy support vector machine to classify images as relevant and irrelevant to the query. The pseudo-labeled images increase the size of the training set, allowing the SVM to learn the boundary between the relevant and irrelevant images more accurately.

Wu and Yap use the whole image as one piece for their pseudo-labelling and image classification. However, an image may only display a semantic concept in a small part of the image. Therefore, it may be better to first segment and split the image into several smaller regions and then classify the image based on those images. Image segmentation regularly relies on splitting an image based on its colors and edge information. Unfortunately, image segmentation is a hard problem that currently has no good solution; segmentation algorithms tend to split the images poorly and do not fit with how humans would segment the image. Also, the low level features that short term learning algorithms use for classification do not deal well with inconsistencies in the size and shape of regions; in many segmentation techniques, regions may not even be connected. Therefore, work has been done on arbitrary segmentation, where images are split into regular regions, ignoring the color and edge information represented in the image. Block-based segmentation has been used effectively for content based image retrieval and annotation [4]. Qi and Han experimented with several methods of block segmentation for improving image annotation [5].

Another approach for content-based image retrieval is long term learning, or learning over many queries. Long term learning techniques focus on learning the semantic relationships between images and tend to pay little attention to the low level features. They are usually combined with short term learning techniques, where short term learning techniques are used during the query and the long term learning techniques are applied to learn between the queries. This approach allows the user to find images relevant to the current search using the short term learning and for the system to improve over the long run using the long term learning.

Research has also been done exploring long term learning applied to image regions. Jiang et al. explored a technique for learning the semantic information contained in the regions of image using boosting and a fuzzy codebook [6]. Their system attempted to determine the probability that a semantic concept was contained in an image as well as proposing a technique for selecting the most representative features. Another approach to long term learning is the semantic kernel learning technique developed by Gosselin and Cord [7].

Carneiro et al. attempted to use supervised learning to create a system that automatically annotated images [4]. After annotating the images, keyword searching allows users to search for images. They used multiple instance learning and expectation-maximization with Gaussian mixture models to determine the probabilities of labels being present in an image. Other research has been done using a multiple Bernoulli relevance model rather than a Gaussian mixture model with good results [8].

An effective approach to long term learning proposed by Han et al. is tracking the number of times images are returned with each other and how many of those times the images were labeled as relevant to the query by the user [9]. This approach is effective for learning the semantic relationships in the images, but it has some problems. It stores the relationships between each two images, so it requires $O(n^2)$ space, where n is the number of images in the database. Furthermore, Han et al.'s memory learning technique is susceptible to images being mislabeled by the user, where the relationship between two images will therefore be learned incorrectly.

3. SHORT TERM LEARNING METHODS

3.1. Overview

To deal with the inefficiency issues related to global feature based SVM learning, we propose a region-based approach that uses fuzzy SVM learning. The proposed method uses relevance feedback combined the pseudo-labeling technique proposed by Wu and Yap [3] and the block segmentation technique described by Qi and Han [5] to classify images. The use of pseudo-labeling and fuzzy SVM learning increases the size of the training set, while the region-based approach allows the proposed algorithm to learn which regions represent a semantic concept. Therefore, Wu and Yap's pseudo-labeling technique is applied to the regions of an image rather than to the whole image and the fuzzy SVM classifies the regions of the image. Then, the fitness values of the regions are combined to obtain a more accurate fitness value for the image. Experiments showed that summing the directed distances of the regions from the SVM boundary is an effective measure of the fitness for the image. This technique allows the SVM to learn the boundaries between regions that are relevant and irrelevant to the query image.

However, state of the art image segmentation techniques are still inaccurate, so we use a image blocking technique proposed by Qi and Han. They found that the center of the image was more likely to capture the semantic concept and therefore took more regions near the center of the image. Their work found that a five-region segmentation had a good tradeoff between performance and running time. A sample of their segmentation is shown in Figure 1. The algorithmic view of a query is explained in Figure 2 and choosing the regions to pseudo-label and the calculation of the fuzzy weights for the

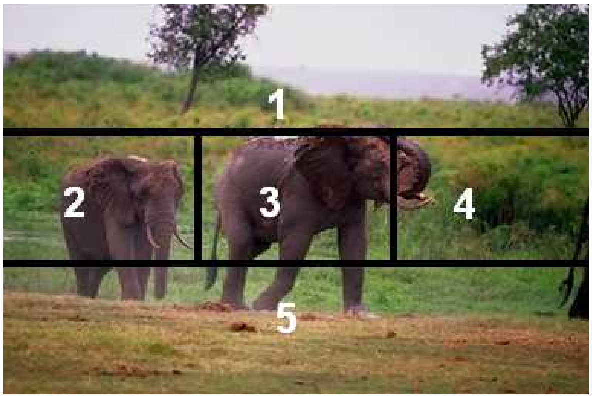


Fig. 1. An example of Qi and Han's five-region segmentation.

pseudo-labels is explained in depth in the following section.

1. Find the images most similar to the query image using Euclidean distances in the low level feature space.

2. The user labels the returned images as either relevant or irrelevant to the query image.

3. Split each image into five regions, with each region retaining the label of the original image.

4. Cluster the both the relevant and irrelevant regions separately using the K-means algorithm and pseudo-label the regions near the cluster centers with weights calculated in (1)-(4).

5. Train a fuzzy SVM on the labeled and pseudo-labeled regions and classify the regions using the fuzzy SVM. Sum the directed distances from the SVM boundary for all the regions in an image to get a total fitness score for that image.

6. Return the images with the highest positive distance and repeat steps 2-6.

Fig. 2. Algorithmic view of a query

3.2. Pseudo-labeling

A query involves a number of iterations of returning images and user feedback of labeling the returned images. These images can then be deconstructed into their constituent regions, with the regions carrying their parent image's labels. Then, the relevant and irrelevant regions are clustered using the K-means algorithm. Wu and Yap used subtractive clustering to determine the number of clusters to use for the K-means algorithm, but subtractive clustering is slow and highly dependent on its parameters. Instead, this work experimentally determined that using eight clusters was close to optimal. After applying K-means clustering, the regions closest to the cluster centers are chosen to pseudo-label; this work experimentally determined that the three nearest regions should be chosen. These regions are given the same label as the nearby cluster center.

After the regions have been pseudo-labeled, an estimation of the accuracy of the label must be computed. First, calculate the ratio of the distance of the region to the nearest cluster center with the same label and the distance of the region to the nearest cluster center with the opposite label.

$$r(x_p) = \frac{\min_i d_e(x_p, v_{s_i})}{\min_j d_e(x_p, v_{o_j})} \tag{1}$$

where x_p denotes the pseudo-labeled region, d_e denotes the Euclidean distance function, v_{s_i} denotes the center of the i^{th} cluster with the same class label as x_p, and v_{o_i} denotes the center of the j^{th} cluster with the opposite class label. Therefore, $\min_i d_e(x_p, v_{o_j})$ denotes the distance of x_p from the nearest cluster center with the same label. The closer the region is to a cluster with the same label, the more likely it is related to that cluster. Similarly, the closer the region is to an oppositely labeled cluster, the less likely its label is to be correct. Therefore, Wu and Yap adopt an exponential fuzzy function based on the ratio of the distances calculated in (1).

$$w_1(x_p) = \begin{cases} \exp(-a_1 r(x_p)) & \text{if} \quad r(x_p) < 1 \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

where $a_1 > 0$ is a scaling factor.

The predicted weight of the pseudo-labeled regions should rely on how well the label agrees with the label determined by the trained SVM. A greater distance between the region and the SVM boundary implies that the label should carry more weight if the pseudo-label is the same as the SVM-predicted label. Therefore, the second factor of the weight function is given by:

$$w_2(x_p) = \begin{cases} \dfrac{1}{1+\exp(-a_2 y)} & \text{pseudo-label is positive} \\ \dfrac{1}{1+\exp(a_2 y)} & \text{otherwise} \end{cases} \tag{3}$$

where $a_2 > 0$ is a scaling factor and y is the directed distance from x_p to the SVM boundary.

Finally, the overall weight should be a combination of the two weighting functions. Therefore, the final predicted weight of the label is given by

$$g(x_p) = w_1(x_p)w_2(x_p) \tag{4}$$

This estimated weight is then used to train the fuzzy SVM and select regions that are most related to the query.

3.3. Feature Extraction

The features used for the short term learning algorithm are derived from work by Qi and Chang [10]. Two sets of features were extracted from the images; a 214-dimensional feature set was extracted from the whole image and sets of 36 features were extracted from each of the images regions. The 214-dimensional feature set contains a 64-dimensional color feature vector and a 150-dimensional edge feature. The color features are taken from the 64-bin (8x2x4) HSV-based scaled color descriptor and the edge features used are the expanded MPEG-7 edge histogram descriptor (EHD). The expanded EHD is constructed from the conventional 80-bin EHD, 5-bin global edges, and 65-bin semi-global edges, taken from five edges in 13 different block groupings. The 36-dimensional feature set consists of a 9-dimensional color feature vector, 18-dimensional edge feature vector, and a 9-dimensional texture feature vector. The specific color features used are the moments in the HSV color space, because these features are similar to those used in human perception. The edge features are constructed using the Sobel edge detector on a grayscale version of the image, where the edges are grouped into 18 bins (with each containing 20 degrees). Finally, the texture features are calculated by taking the entropy of each of 9 detail subband images obtained by a 3-level Daubechies-4 wavelet transform.

4. SHORT TERM LEARNING RESULTS

4.1. No Mislabeling

The method proposed in this paper is compared to three other methods: normal SVM learning on the whole image, fuzzy SVM (FSVM) learning on the whole image, and normal SVM learning on the regions of the image. The fuzzy SVM learning used is based on Wu and Yap's work [3] and the region based approaches use the segmentation proposed by Qi and Han [5]. The proposed method is a combination of the fuzzy SVM learning on the regions of the image. These four methods were tested on a 6,000 image subset of the COREL database with 100 images in each of the 60 categories. All of the methods were tested using four iterations of image return and feedback, with 25 images returned every iteration. The algorithms are compared on their precision, where

$$\text{precision} = \frac{\text{number of relevant images returned}}{\text{total number of images returned}} \tag{5}$$

The non-fuzzy SVM methods perform poorly in the second iteration due to the small size of the training set. However, as the training set increases due to the user's labeling, the performance of the SVM increases greatly. On the other hand, the pseudo-labeling improves the results especially well during the early iterations, when expanding the size of the training set can greatly improve the effectiveness of the fuzzy SVM. However, in the later iterations, the training set has increased and increasing the size of the training set is not as necessary. Furthermore, it is possible for the pseudo-labeling technique to mislabel some of the images, which may result in less accurate classifications. Therefore, the precision of the global FSVM increases rapidly in the early iterations, but levels off in the later iterations.

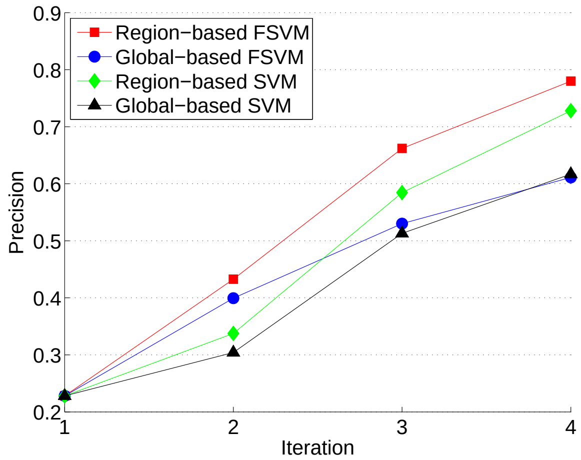


Fig. 3. Precision over a query with no simulated user errors

The region-based SVM performs similarly to the global-based SVM in the early iterations, when the small training set restricts the discovery of regions that are more relevant to the semantic concept of the query However, as the training set grows, the region-based approach improves more rapidly than the global-based approach. The proposed approach of using pseudo labels and working with the regions performs the best. The pseudo-labeling increases the size of the training set, improving the performance in the early iterations, and the use of regions improved the performance in the later iterations by allowing the algorithm to identify the regions that are most relevant to the query.

4.2. Simulated User Mislabeling

Due to user mistakes and carelessness, errors in a user's labels of returned images may occur, so it is important for a short term learning algorithm be resilient to these user errors. If an image retrieval system is incapacitated by a single image being mislabeled by a user, it will be impractical in the real, noisy world. To test the algorithms' resilience to simulated user errors, they were also tested with 5% noise, where 5% of the images were mislabeled by the user. The results with 5% simulated user errors indicate that the proposed technique is relatively resillient, while the other techniques are more affected by the user errors. The mislabeling causes the non-fuzzy SVM methods to not improve significantly during the second iteration. However, the non-fuzzy SVM methods do

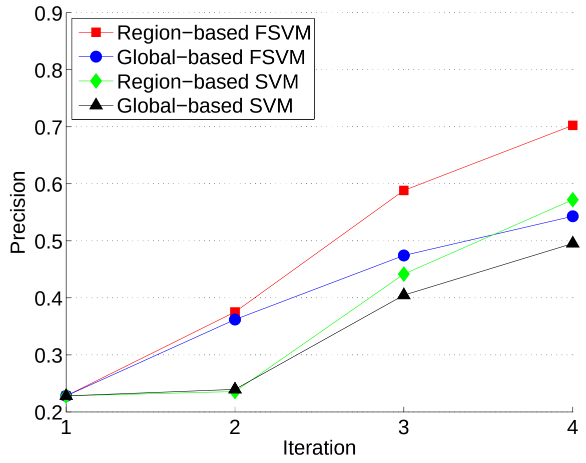


Fig. 4. Precision over a query with 5% simulated user errors

improve in the later iterations, after the size of the training set has increased. The fuzzy SVM methods do perform worse with the simulated errors, but they are more resillient than the non-fuzzy techniques.

5. LONG TERM LEARNING METHODS

5.1. Overview

Long term learning attempts to learn the semantic relationships between images. Several previous techniques for long term learning use a large matrix to store each image to image semantic relationship [9]. However, this requires O(n^2) space and is not feasible for large databases. The long term learning technique proposed in this paper offers a new method for storing the semantic information in a more reasonable amount of space.

Instead of storing the image to image semantic relationships, it is possible to create semantic clusters and only store each image's relationship to each cluster. There will be far fewer semantic clusters than images, so the storage space required will be significantly less than storing the image to image semantic relationships. Also, this technique more closely models the human way of storing information. Humans tend to classify objects into categories and remember how well each object belongs to each category [11]. Note that it is possible for an image to belong to several semantic clusters, which allows for images containing several objects and therefore several semantic ideas.

A query in this framework involves a user inputting a search image and then several images are returned based on low level features and any semantic information stored about the search image. Next, the user labels these images as relevant or irrelevant to the search image. Then, using short term learning and the semantic information about these labeled images, more images are returned and the process is repeated for a number of iterations. In this work, there were 4 iterations and 25 images were returned in each iteration. Each query results in a set of images relevant to the search image and a set of images irrelevant to the search image. The algorithm treats these results as a new semantic cluster and then tries to determine whether this cluster represents the same semantic concept as any existing clusters. If the new cluster does represent an existing semantic concept, the semantic information is added to the existing cluster and the new cluster is removed.

Determining whether two clusters represent the same semantic concept is not simple; any merging algorithm must be resilient to noise and recognize that an image may represent multiple semantic concepts. To determine the similarity of two semantic clusters, calculate the overlap between them, finding the images that are relevant to both clusters. Then, calculate the difference between the clusters, finding the images that are relevant to one cluster, but irrelevant to the other cluster. If the similarity of the clusters outweighs the difference, the clusters should be merged. This method is explained in more depth in Section 5.2.

The other important part of the long term learning algorithm is to predict which images are most semantically related to the search image. At each iteration of the query, there is a set of images that have been labeled as relevant to the search image, including the search image itself. There is also a set of images that have been labeled as irrelevant to the search image. Then, the system attempts to predict which semantic cluster best represents the information in the relevant and irrelevant sets, calculating a fuzzy membership for the query into each semantic cluster. Next, the system calculates the probability that each image in the database is a member of each of the semantic clusters. Finally, the similarity of each image to the search image is given by the product of the probability that the query represents a semantic cluster and and the probability that an image is a member of that semantic cluster. This method is explained in more depth in Section 5.3.

A cluster contains two pieces of information about every image: the number of times it has occurred with other images from the cluster and the number of times it was relevant to other images from the cluster. The fuzzy membership of the image i to semantic cluster j is estimated by

$$\text{membership}(i,j) = \frac{\text{number of times relevant}(i,j)}{\text{number of times occurred}(i,j)} \quad (6)$$

For example, if image I has been returned in 4 queries where the user was searching for the concept represented in semantic cluster C, then

$$\text{number of times occurred}(I,C) = 4$$

If it was labeled as relevant to the query 3 of those times, then

$$\text{number of times relevant}(I,J) = 3$$

Therefore, image I's membership in the semantic cluster C is approximated by

$$\text{membership}(I, J) = \frac{3}{4} = 0.75$$

5.2. Merging Clusters

Merge clusters c and d if they represent the same semantic concept.

1. The fuzzy membership of image i to cluster j is given by:

$$\text{M}(i, j) = \frac{\text{relevant}(i, j)}{\text{occurred}(i, j)}$$

where occurred(i, j) is the number of times image i was returned with other images from cluster j and related(i, j) is the number of times that the image was labeled as relevant to cluster j.

2. Then, define the similarity of clusters c and d as:

$$\text{S}(c, d) = \frac{\sum_i^N \text{M}(i, c) * \text{M}(i, d)}{N}$$

where N is the number of images in the database. The similarity value represents how many images are in both clusters.

3. The irrelevance of image i to the cluster j is given by:

$$\text{I}(i, j) = \begin{cases} 1 - \text{M}(i, j) & \text{if relevant}(i, j) > 0 \\ 0 & \text{otherwise} \end{cases}$$

The irrelevance value measures how many images are relevant to one semantic cluster, but not the other cluster.

4. The difference between clusters c and d is defined as:

$$\text{D}(c, d) = \max\Bigg(\frac{\sum_i^N \text{I}(i, c) * \text{M}(i, d)}{N}, \frac{\sum_i^N \text{I}(i, d) * \text{M}(i, c)}{N}\Bigg)$$

where N is the number of images in the database.

5. If $\text{S}(c, d) - \text{D}(c, d) \geq k$ where k is a constant with $0 \leq k < 1$, then merge the clusters to form a new cluster. This merger is performed by summing the number of times relevant and the number of times occurred of clusters c and d.

5.3. Calculating Semantic Distance

Calculate the semantic distance from an image to the current query.

1. Find which clusters the relevant images belong to. For cluster c,

$$\text{related}(c) = \sum_{i \in R} \text{relevant}(i, c)$$

where R is the set of images that have been labeled as relevant to the search image.

2. Find which clusters the irrelevant images belong to. For cluster c,

$$\text{unrelated}(c) = \sum_{i \in IR} \text{relevant}(i, c)$$

where IR is the set of images that have been labeled as irrelevant to the search image.

3. Determine the probability that a cluster represents the semantic concept.

$$\begin{aligned} \text{difference}(c) &= \text{related}(c) - \text{unrelated}(c) \\ \text{P}(c) &= \frac{\text{difference}(c)}{\max_d(\text{difference}(d))} \end{aligned}$$

4. The fuzzy membership of image i to cluster c is given by:

$$\text{M}(i, c) = \frac{\text{relevant}(i, c)}{\text{occurred}(i, c)}$$

5. The semantic distance of image i to the current query is:

$$\text{distance}(i) = 1 - \sum_c (\text{M}(i, c) * \text{P}(c))$$

6. LONG TERM LEARNING RESULTS

The proposed algorithm was tested against Han et al.'s memory learning technique [9]. The tests were performed on a 6,000 image subset of the COREL database in a total of 60 categories. Four feedback iterations were used and 25 images were returned at each iteration. Each test started with training on a number of queries; the algorithms were trained on 120, 300, and 600 queries; 2%, 5%, and 10% of the database respectively. During this learning period, if an image was returned in a query, it could not be returned later in that query. This technique maximized the amount of semantic information that could be learned on the training set. After the initial training, the systems were tested on the remaining 5,400 images. In the testing phase, no learning was performed and images were allowed to be returned multiple times during a query. It is possible for users to accidentally mislabel some returned images, and therefore it is necessary for content-based image retrieval systems to be resillient to this noise. To test their resillience to noise, the systems were tested with and without a 5% chance of the user mislabeling each returned image. The systems were compared based on the precision of the returned images, where precision is defined in (5). The results of this testing are displayed in figures 5 - 7.

The results in the tests with different training sizes are similar. The larger training sets gave better precision with the queries, but the relationship of the systems remained the same. The results indicate that the proposed method of semantic clustering performs similarly to Han et al.'s technique when there is no user mislabeling [9]. However, the proposed technique's advantage is that is far more efficient with regards to space. Han et al.'s technique requires O(n^2) space where n is the number of images in the database because it stores the image to image relationships. On the other hand, semantic clustering only requires O($n * c$) space where c is the number of semantic clusters. It is worth noting that c will be significantly smaller than n. During the testing, n was 6,000, while c was adaptively found by the algorithm and was approximately 81 for the regular SVM testing and 67.5 for the fuzzy SVM testing. These are close to 60, the optimal number of semantic clusters as determined by humans for the database.

The results also indicate that the semantic clustering is significantly more resilient to user mislabeling. This resilience is due to the calculations about merging the semantic clusters contained in Section 5.2. These calculations mean that semantic clusters are merged only if they overlap enough. The overlap refers to how many images are labeled as relevant to both clusters and how many images have been labeled as relevant to one cluster and irrelevant to the other cluster. This technique uses the number of times an image was labeled as relevant rather than the percentage of the time it was labeled as relevant like Han et al.'s memory learning technique does. If an image is mislabeled once as being relative to another picture, then Han et al.'s technique treats it the same as if it was returned many times and labeled as relevant each time. The semantic clustering technique does take the number of times it was relevant into account and therefore is less susceptible to noise. Resilience to noise is important in the real world, because it is normal for users to make mistakes, so a CBIR system must be able to handle a small amount of images being mislabeled. The proposed semantic clustering algorithm is far more effective at handling the simulated user mislabeling than Han's memory learning algorithm.

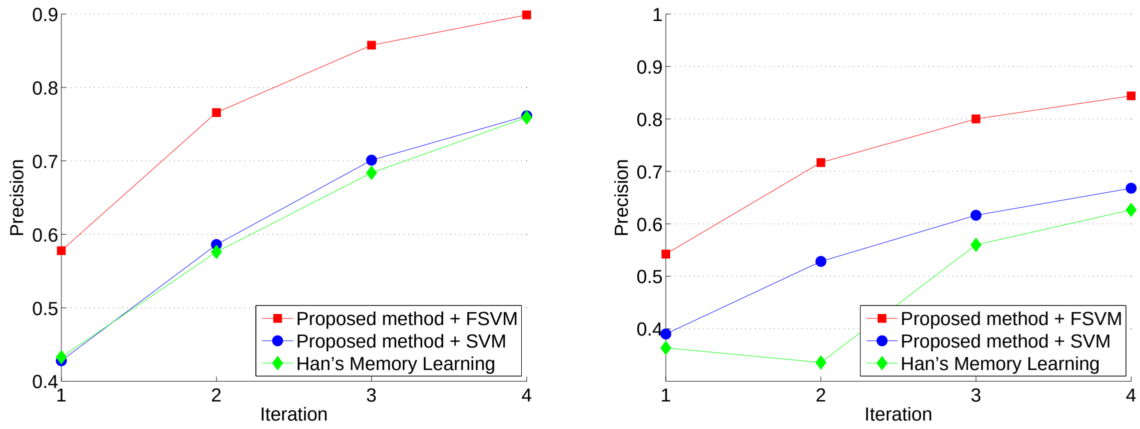
The size of the training set did not affect the relationship of the systems. However, it did affect the precision of all the systems, but not as significantly as expected. The proposed system using normal SVM learning without simulated user errors achieved a precision of 0.87 by the fourth iteration with 600 training queries and still achieved a precision of 0.76 with only 120 training queries. This shows that even a small training set can be sufficient for getting high precision results over the whole query. On the other hand, the size of the training set had a much stronger influence on the first iteration of the image retrieval. The proposed system using normal SVM learning without simulated user errors had a precision of 0.61 in the first iteration with 600 training queries, but only had an average precision of 0.43 in the first iteration when using only 120 training queries. Overall, the results indicate that additional training can be helpful, but is not necessary for a CBIR system to be effective.

The results also show that the short term learning can greatly affect the results of a long term learning algorithm. A better short term learning algorithm, like the region based fuzzy SVM learning technique proposed earlier in the paper, causes more relevant images to be returned during the training stage as well as improving the short term learning results during the query. Therefore, it is possible for the long term learning algorithm to learn more semantic information about the images. Combining good short term learning techniques with good long term learning techniques is very effective and is necessary for any real world content-based image retrieval application.

7. CONCLUSIONS

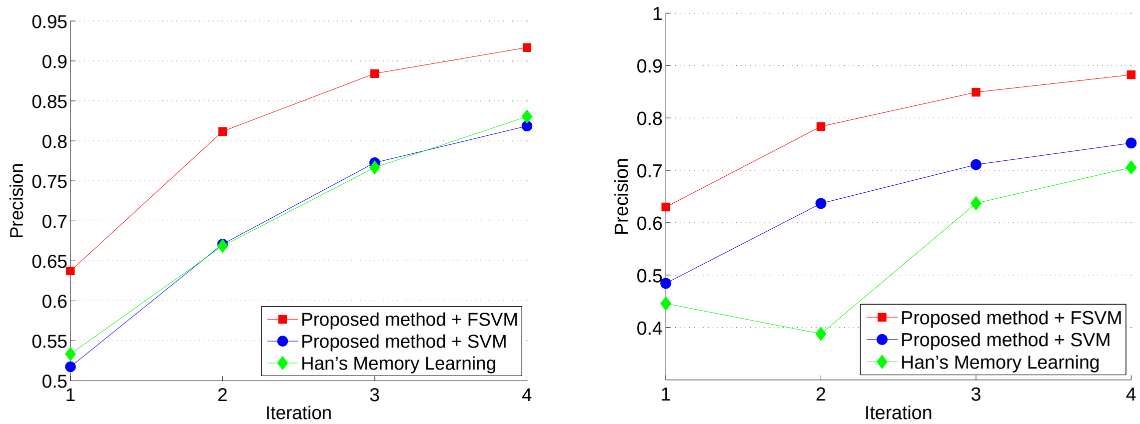
Any real world application of content-based image retrieval must rely on both short term and long term learning. For short term learning, using pseudo-labeling and fuzzy SVM learning can be effective for increasing the size of the training set and the use of a region-based approach can improve the precision of the system. Also, semantic clustering is effective and efficient for long term learning. Semantic clustering is more resilient to users' mislabeling than previous techniques, and performs as well as previous techniques when users do not mislabel. Furthermore, semantic clustering requires significantly less space than previous techniques and is more capable of scaling to large databases.

This research has clear applications to work in image annotation and should be explored. Semantic clustering provides an effective method for determining the relations of images and then propagating users' annotations to related images. Also, this work may be effective for images with multiple categories. Semantic clustering allows an image to belong to several clusters and should be capable of determining to which semantic cluster the query belongs, but this must still be tested. However, it is difficult to find standardized databases that categorize an image into several clear categories. Finally, this work should be tested on a large database to test how well it scales. However, it is difficult to compare this work to previous work because the previous work is not capable of handling large databases.



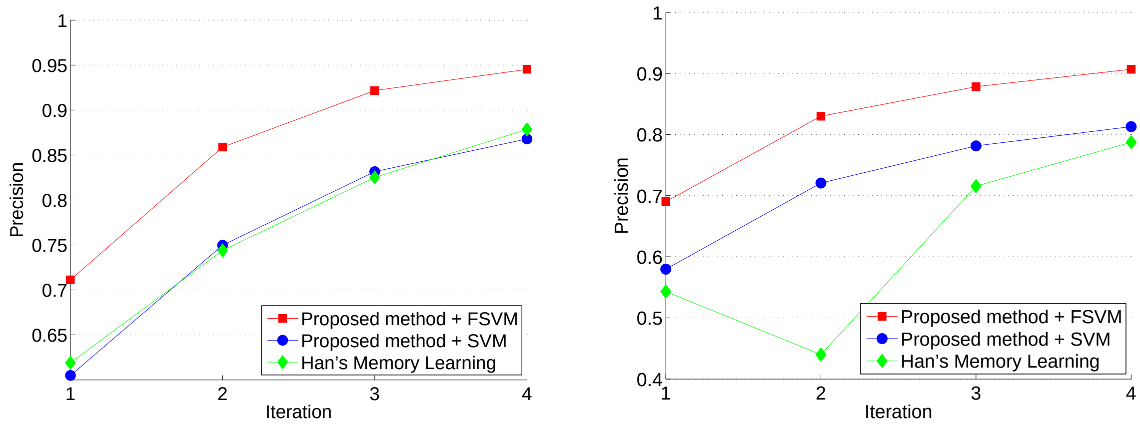
(a) No noise (b) 5% Noise

Fig. 5. 2% Training Set - 120 queries



(a) No noise (b) 5% Noise

Fig. 6. 5% Training Set - 300 queries



(a) No noise (b) 5% Noise

Fig. 7. 10% Training Set - 600 queries

8. REFERENCES

[1] Edward Chang, Kingshy Goh, Gerard Sychay, and Gang Wu, "Cbsa: content-based soft annotation for multimodal image retrieval using bayes point machines," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 1, pp. 26–38, January 2003.

[2] Simon Wilson and Georgis Stefanou, "Bayesian approaches to content based image retrieval," in *Proceedings of the International Workshop/Conference on Bayesian Statististics and its Applications*, Varanasi, India, January 2005.

[3] Kui Wu and Kim-Hui Yap, "Fuzzy SVM for content-based image retrieval," *IEEE Computational Intelligence Magazine*, vol. 1, no. 2, pp. 10–16, May 2006.

[4] Gustavo Carneiro, Antoni B. Chan, Pedro J. Moreno, and Nuno Vasconcelos, "Supervised learning of semantic classes for image annotation and retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 3, pp. 394–410, March 2007.

[5] Xiaojun Qi and Yutao Han, "Incorporating multiple SVMs for automatic image annotation," *Pattern Recognition*, vol. 40, no. 2, pp. 728–741, 2007.

[6] Wei Jiang, Guihua Er, Qionghai Dai, and Jinwei Gu, "Similarity-based online feature selection in content-based image retrieval," *IEEE Transactions on Image Processing*, vol. 15, no. 3, pp. 702–712, March 2006.

[7] Philippe H. Gosselin and Matthieu Cord, "Semantic kernel learning for interactive image retrieval," in *IEEE International Conference on Image Processing*, Genoa, Italy, September 2005, vol. 1, pp. 1177–1180.

[8] S. L. Feng, R. Manmatha, and V. Lavrenko, "Multiple bernoulli relevance models for image and video annotation," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, June 2004, vol. 2, pp. 1002–1009.

[9] Junwei Han, King N. Ngan, Mingjing Li, and Hong-Jiang Zhang, "A memory learning framework for effective image retrieval," *IEEE Transactions on Image Processing*, vol. 14, no. 4, pp. 511–524, April 2005.

[10] Xiaojun Qi and Ran Chang, "Image retrieval using transaction-based and SVM-based learning in relevance feedback sessions," in *Proc. of International Conference on Image Analysis and Recognition*, Montreal, Canada, August 2007, LCNS, vol. 4633, pp. 637–649, Springer.

[11] Steven Pinker, *How the Mind Works*, W. W. Norton & Company, New York, New York, 1997.