Training Templates for Scene Classification using a Few Examples

A. Lakshmi Ratan and W.E.L. Grimson*
Artificial Intelligence Lab, M.I.T.
545 Tech. Square, Cambridge, MA 02139.
aparna@ai.mit.edu, welg@ai.mit.edu

Abstract

In this paper, we investigate a method for extracting simple, flexible, relational templates that capture the color, luminance and spatial properties of classes of natural scene images from a small set of examples. We have built an interactive system that allows the user to build his own templates by selecting a set of example images and letting the system extract a set of templates that capture the common relations within the set of images. Given a small set of example images, the classification system works by automatically building flexible templates that define color, luminance and spatial relations between image patches in these examples. These extracted templates can be matched against the entire database to obtain other images that belong to the query-class. The system also gives the user the option of using the results of the initial classification in order to refine the query and perform a more selective search if needed.

The system uses very low resolution images to extract the templates from a set of examples. It has been used to retrieve a few classes of natural scenes from the COREL photo library. Our experiments show that the algorithm is fast, requires little storage and works reliably in this domain.

1 Introduction

We investigate a method for learning relational templates that capture the color and spatial properties of classes of natural scene images given a set of examples that belong to a class. The templates that are extracted contain information about the relative luminance and color properties as well as the spatial configuration of image patches in the example set of images. These templates can be used to represent a user-defined class and can then be used to classify other target images. This system allows the user to define the class using example images rather than hand-drawn representations or other abstract queries and gives the user a chance to refine the query and perform a more selective search based on the initial set of matches that the system returns.

 This work is to appear in the Proceedings of CAIVL '97 in Puerto Rico. Research supported in part by ARPA under ONR contract N00014-95-1-0600

1.1 Image Classification Systems

In the past few years, the growing number of digital image and video libraries has led to the need for flexible, automated content-based image retrieval systems which can efficiently retrieve images from a database that are similar to the user's query. Since what a user wants can vary greatly, we also want to provide a way for the user to explore and refine the query by letting the system bring up examples. Typical techniques that have been proposed to search through these large image databases use simple image properties like color histograms and color layout([15] [3]), use image texture properties ([1] [9]), add spatial information to feature-based methods ([12],[13], [16], [14]) or use the information encoded by the dominant wavelet coefficients ([5]). These methods extract features from the image and represent them as a vector which is then compared with similar vectors from the target images. Most of these methods deal with absolute properties of the image like distribution of absolute color in various parts of the image or distribution of textural properties in the image. More recently, work by Lipson and Sinha ([6], [11]) in scene classification illustrates that pre-defined flexible templates that describe the relative color and spatial properties in the image can be used effectively for this task. They demonstrate that the following three ideas are crucial and effective in scene classification: (1) relative relationships matter more than absolute properties, (2) global configuration of the relationships is important and (3) low resolution information is sufficient. In this paper, we investigate the problem of trying to learn photometric and spatial relations between image patches. Our method builds these flexible, relational templates given a set of example images that represent a user-defined class. The system uses a multiresolution approach to extract and refine these relational templates.

1.2 Previous Work

One of the most popular global techniques for color indexing is color-histogramming which measures the overall distribution of colors in the image. While histograms are useful because they are relatively insensitive to position and orientation changes, they do not capture the spatial relationships of the color regions and thus have limited discriminating power. Many of the existing image-querying systems work on entire images or in user-specified regions by using distribution of color, texture and structural properties. The QBIC system [3] and the Virage system [1] are examples of such systems. Some recent systems try to incorporate some spatial information into their color feature sets ([12],[13], [16], [14], [4], [10]). Pass et al. [16] have a technique that is better than color histogramming by splitting the global histogram into coherent and scattered components which identifies connected color regions. However, it does not support querying with spatial locations of the color regions. Stricker et al. [14] divide the image into five overlapping regions and compute the first three moments of the color distributions in each region. They allow querying by using the five absolute region locations and properties and the method is insensitive to small rotations. VISUALSeek [12] is another system that supports query by color, texture and spatial layout. It also partitions the image into regions, but allows a region to contain multiple colors and lets pixel belong to several different categories. All of the methods described above use the absolute color and texture properties of the whole image or fixed regions in the image and some incorporate spatial relations between connected regions having similar color properties. If we consider the domain of natural scene classification, these absolute properties can vary between images of the same class. For example, Figure 8 shows variations in color, texture, illumination and shape for images that belong to the class of "Fields".

More recently, work by Lipson [6] and Sinha [11] in scene classification illustrates that the relative color and spatial properties in the image are more invariant than absolute properties and that the global configuration of these relationships is effective in classifying scenes. Such flexible templates that define relative photometric properties and spatial relations for classes of natural images are used to query the system. The system uses the template to search the database and returns the set of images that satisfy the relations specified by the template. The system has been tested with a variety of predefined templates that represent natural scenes like fields, rivers, waterfalls, snowy mountains and retrieves images that belong to the class specified by the template efficiently and reliably. In this paper, we describe an interactive process where the system extracts these relations automatically from a set of example images selected by the user.

All of the systems described above require the users to precisely specify what they want. Minka and Picard [7] introduced a learning component in their system by using positive and negative examples which lets the system choose image groupings within and across images based on color and texture cues. Their system provides a way to learn labels that describe various parts of the scene based on the examples. In this paper, we focus on learning luminance, color and spatial relations between image patches and building these flexible, relational templates automatically from a set of example images that represent a user-defined class.

Given a set of examples, our system extracts the dominant relative luminance, color and spatial relations and builds flexible relational templates similar to Lipson's pre-defined templates [6]. Our system uses a multiresolution approach to extract and refine these relational templates. The extracted set of templates are matched across all the images and the user is presented with a set of possible images in the class specified by the examples. The system can be more selective if the user chooses more examples which lets the system refine the template by moving to a higher resolution.

2 Method

2.1 Motivation: Pre-defined flexible templates for scene classification

The flexible templates constructed by Lipson [6] encode the scene classes as a set of image regions and the qualitative relationships between those regions. This is similar to the ratio-template constructed by Sinha [11] to detect faces under varying illumination conditions. The ratio template captures the changes in luminance between fixed image patches. The flexible templates constructed by Lipson for the task of scene classification include spatial and photometric relationships between image patches. Each image patch has properties in the color and luminance channels. These templates describe the color relationship (relative changes in the R,G,B) channels, luminance relationship (relative changes in the luminance channel) and spatial relationship between two image patches. Lipson hand-crafted these flexible templates for a variety of scene classes and showed that they could be used to classify natural scenes efficiently and reliably. In this paper, we would like to create an interactive system that allows the user to build his own templates by selecting a set of example images and letting the system extract a set of templates that capture the common relations within the set of images.

2.2 Example of a predefined template

Figure 1 is an example of a predefined template used to classify scenes (Lipson [6]). The template describes the (color, luminance and spatial) relationship between image patches. The relations used for the color and luminance channels are $(>, <, =)$. The spatial relations are (above, below, left, right). These hand-built templates capture the relevant color and spatial relations of classes of natural images and have been used to classify scenes efficiently.

2.3 Our System

Our goal is to construct flexible templates that capture the common color and luminance patterns and their spatial relationships present in a set of example images. The method used to construct these flexible templates from a training set of images consists of the following steps (Figure 2):

1. building histograms,
2. extracting peaks to form template units,
3. building templates,
4. classifying new images

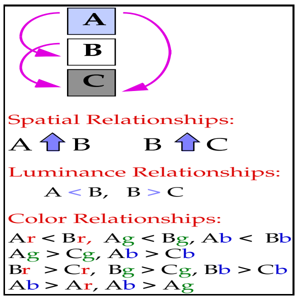

Figure 1: Example of a pre-defined template. This figure is taken from [6] with permission of the author.

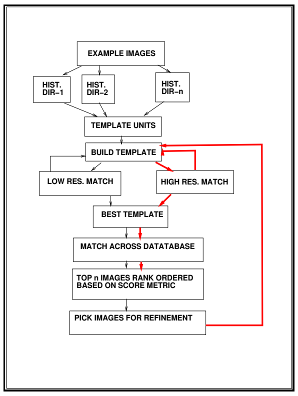

Figure 2: Flow of control in the system

5. incorporating new examples and refining the template at higher resolution.

2.4 Directional histograms of luminance and color relations

Since we want to capture commonality of relationships across a set of example images while allowing for spatial variation, we cannot just use correlation on blurred, low resolution images. We use very low resolution images (16x16) to build a histogram of relations at the first stage. The relationships are defined on k channels for every pixel (eg. red, green and blue color channels, luminance, texture, etc.) and there are 4 choices of symbols for a relationship between a pair of pixels ($>$, $<$, $=$, $*$) where $*$ is the "don't care symbol". This implies that there are 4^k bins into which a pair of pixels can cast a vote. The $*$ symbols cause multiple votes to be cast. The current system uses 4 relative relations for the red, green, blue and intensity channels.

We allow for variation in position as follows. For each pixel in the low resolution image, take a neighbourhood of pixels in some direction relative to it (Figure 3). For example, in the NS direction, the neighbourhood for a pixel $P(i,j)$ is given by the set of pixels

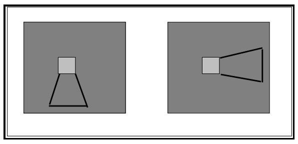

Figure 3: LEFT: Foreach pixel take a neighbourhood of pixels around it in different directions. Left Fig: top-bottom, Right Fig: left-right.

$P(i+k, j+l)$ where $k = -1, 0, 1$ and $l = 1 \ldots 6$. For each pixel, record the signature of the color and luminance relations between the pixel and the pixels in the neighbourhood corresponding to a particular direction. Histogram these signatures into bins. Each bin in the histogram represents a particular relationship defined on the red, blue, green and luminance channels using the relations ($>, =, <$). For each pixel, build these histograms for different directions separately. We ignore the bin which represents the equality relation i.e. uniform regions in the image do not contribute to the histogram peaks.

This is implemented as follows.

- Each pixel P_i in the low resolution image is now a unit represented with (1) a rough color $P_i(col)$ and (2) histograms of luminance and color changes in different directions $P_i(hist_d)$ where d = (NS, EW) (Figure 4).
- For each distinct pixel unit and a particular direction find the common peaks in each of the example images based on the rough color and dominant relations in the histogram. Each example image has $16 * 16$ pixel-units for each direction.

Once we find the common peaks, we have extracted template units which have rough color in a node and the dominant color and luminance relationships between that node and its neighbours in a particular direction (Figure 5).

2.5 Building templates from histogram peaks

Once we have built the directional histograms, we extract common peaks in the histograms of the sample images. Each peak gives us a rough node color and a relational signature that defines a part of the final template (Figure 5). We need to assemble the final template by trying possible combinations of the template units in the NS and EW directions (Figure 5). We implement this as follows. Once we pick the most frequent pixel-unit-peaks that overlap in the example images, we add relative spatial information to the set of template units by considering all the units pairwise and recording their spatial relationship which is represented by the pair (LEFT / RIGHT / SAME, UP/ DOWN / SAME). Figure 6 illustrates this process. This gives us a template with nodes that have rough color properties and are linked to other nodes by color, luminance and spatial relations.

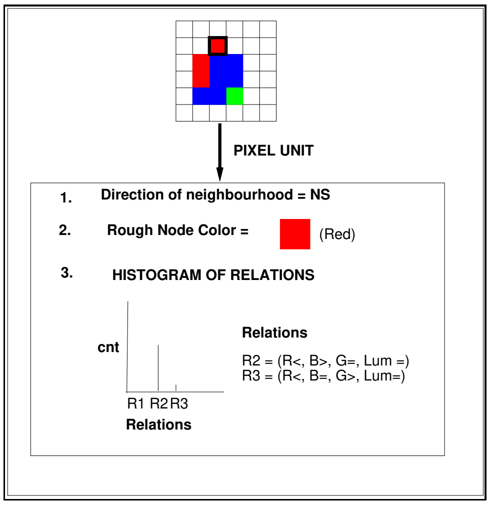

Figure 4: Example of a pixel unit extracted from an image. The pixel unit has a (1) rough node color and (2) a histogram defining the relations between the pixel and other pixels in the neighbourhood corresponding to the NS direction. The pixel also has a similar relational histogram defined in the neighbourhood corresponding to the EW direction which is not shown in the figure.

Finally we match the extracted template against all the images in the database and rank order the images based on how well they are explained by the extracted template.

2.6 Classifying new images

We have class-templates that are built from example images belonging to class C, and a new image. For each pixel P_i in the new image, we compute pixel units that capture the rough color properties $P_i(col)$ and the dominant color and luminance relations $P_i(hist_d)$ between the pixel and its neighbours in different directions d. We want to determine if the new image is

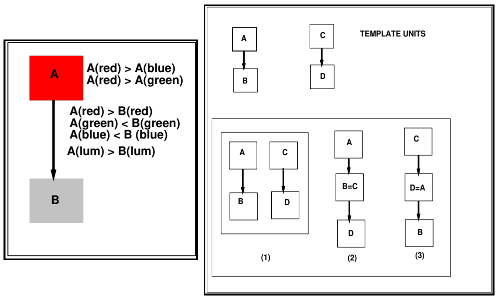

Figure 5: LEFT: Example of a template unit. RIGHT: Example of possible templates obtained by assembling two template units in the top-bottom direction.

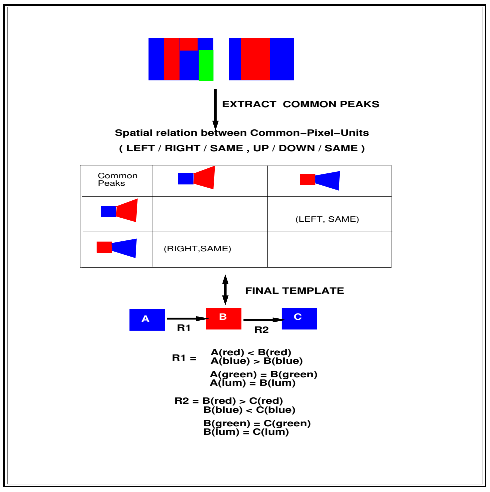

Figure 6: Example of a template extraction process. Note that the patches can vary in size and shape. Patches that are not common in both images do not appear in the template.

in class C based on how well it can be explained by the class-template. To do this, we find a path of regions that is consistent with the color, luminance and spatial relations encoded in the template. Our template T consists of n template units $(A_0 ... A_n)$, each describing a particular color and luminance relation in a particular direction. A sequence of A_i's also describes a particular spatial configuration.

We find parts of the new image that are consistent with the template by measuring the extent to which the template units match pixel units in the image and the extent to which the overall spatial configuration of the template holds in the image.

We use a metric which maximizes the parts of the image that are explained by the template and penalizes for parts of the image that have relations that are not explained by the template.

Let $f(T, C)$ be the part of the example images explained by the template and $f'(T, C)$ be the part of the example images not explained by the template. Similarly, let $f(T, I)$ be the part of the new image explained by the template and $f'(T, I)$ be the part of the new image not explained by the

$$Score = f(T, I) - \alpha * (f'(T, I))$$

where $\alpha = f(T, C)$

This scoring metric has a high value when $f(T, I)$ is high. When $f'(T, I)$ is not negligible, a higher α implies a lower score i.e. if the model explains the class-images very well, then an image with a lot of unexplained relations gets a lower score than in the case where the example images and the new image have many unexplained relations.

2.7 Template Refinement

In the first stage, the system builds the template given the example images and returns the top 30 matches from the database to the user. The user can now modify the example image set by adding or removing images and repeat the process at a higher resolution. The system can then refine the template by looking for finer distinctions in color and luminance properties of the new images and subdividing the original template patches to include these finer distinctions. We are currently testing the performance refinement step.

3 Experiments and Results

We have built an interactive classification system which allows the user to pick a few examples from a random set of images (top two rows of Figure 10) from the database. The system extracts the dominant luminance and color relations between image patches and their spatial relations from the query images and searches the database for other images that have the same relationships. The system returns an ordered set of images that belong the query-class. The dataset consists of 800 images of natural scenes like "sunsets", "lakes and rivers", "snowy mountains", "fields", "coasts" etc. from the COREL libraries. All the images are smoothed using a Gaussian filter and down-sampled to 32x32 and 16x16 images. All the templates were built using coarsely quantized 16x16 images.

3.1 Proof of Concept

In order to test if the training procedure works we performed the following experiments to compare it with hand-built templates. We ran the hand-crafted template [6] for the fields and snowy mountains classes against the database. We then picked 5 images from the resulting set and used these to build a template using the relational histograms. We then matched the new template against the images. The results shown in table 1 indicate that the trained templates perform as well as the hand-built ones for the set of 200 mixed images. The true positives and false positives do not change much when the classification is done using the trained templates. The false positives for the fields for both the hand-built and the trained templates were beaches and green mountains with sky. The false positives for the mountains in both cases included pictures of fields with clouds in the sky.

3.2 Results using trained templates

Figure 8 shows the results of training the template on the example field images selected in red in the top two rows of the figure. The query images are selected from a random set of 20 images that the user can bring up from the database. The database has 800 images. Figure 7 shows examples of the quantized patches in the low resolution images for fields. The templates were built from these low-resolution images by extracting the dominant luminance and color relations (units of the template) and assembling the template units using spatial relations. The top 30 matches that matched the field-template are also shown. The false positives (Figure 9) indicate that the simple color relationships defined by the template might not be sufficient to make finer distinctions when the color and spatial relations are satisfied by 2 classes (for example, the difference between a beach and a brown field with sky).

Temp. type	F TP	M TP	M FP	F FP
Built	37	39	2	3
Train	37	37	4	4

Table 1: Comparison with hand-crafted templates for a set of 200 mixed images. F= fields, M = Mountain, FP= false positive, TP=true positive

Figure 10 shows the results of running our system on the example snowy-mountain images which are the ones selected in red in the top two rows of the figure. The database has 800 images. Figure 7 shows examples of the quantized patches in the low resolution images for snowy mountains. The snowy-mountain template extracted from the examples is a 3 patch template in the N-S direction (see figure 7). The top 30 matches that matched the snowy-mountain-template in the order they were found are also shown in the last three rows of the figure. The false positives for this query were mostly fields with clouds.

Figure 11 shows the results of the system running with the "snowy mountain with lake" query.

We compared our results with the results obtained for similar queries using QBIC's color layout mode and color histogramming (table 2). We ran the QBIC system using a field image as the query in the color-layout mode. We took the top 50 matches and used three of the field scenes returned to build our template. The results of matching our template with the 50 images indicate that out of the 43 false positives returned by QBIC, only 21 were classified as fields by the built template. Table 2 shows that the number of false positives goes down with our method. The top 30 matches contain 10 out of the 12 fields in the test set in our method which is an improvement over the top 20 matches returned by QBIC.

Table 3 summarizes the average performance of the system over 50 trials for two classes (fields and snowy-mountains). There were 110 field images and 100 snowy mountain images in the dataset. These were classified manually for the purpose of evaluating the performance of the system.

4 Summary and Future Directions

In this paper, we have demonstrated a system that classifies images by automatically building flexible templates that define color, luminance and spatial relations between image patches, given a set of example images. The system also gives the user the option of using the results of the initial classification in order to refine the query and perform a more selective search. Lipson's experiments with pre-defined templates [6] showed that low resolution information can be used effectively to classify images and avoid false

Method	TP in top 30	FP in top 50
QBIC	7	43
Histogram	5	45
Trained templates	10	21

Table 2: Comparison with QBIC (color layout) and histogramming. There were a total of 12 fields in the test set. The table shows how many of these 12 came in the first 30 matches. The trained templates returned 10 fields and 21 false positives out of the top 50 images from QBIC. The remaining false positives were not classified as fields.

Class	FP in top 30	FP in top 100	FN
Fields	5	28	7
Snowy Mountains	3	16	13

Table 3: Summary of Results using trained templates. The table has the average number of false positives in the top 30 and the top 100 matches and the av. number of false negatives (images in the class that did not get classified correctly by the system) over 50 runs.

targets caused by high frequency noise. Our experiments illustrate that low resolution images are sufficient to automatically extract and encode properties of natural scene images that can be used for classification.

The system has been tested for a few classes on a database of 800 images of natural scenes from the COREL photo library. The scenes in the database include "sunsets", "coasts", "fields", "mountains and glaciers" among others. This method of building relational histograms can be used to prune the database efficiently before performing a more selective search if needed. The algorithm can also be used to organize the database so that matching at query time can be more effective.

We have tested the system with a 800 image database in order to verify that learning these templates from an example set provide an effective means of classifying scenes. We are in the process of testing the system with a much larger database to see how the performance scales up in terms of computational complexity and reliability. We would like to include other relational cues (eg. texture, structure) in addition to the color and luminance properties described above in order to refine templates and discriminate between images at a finer level. We would also like to extend the system to incorporate false positives from the initial stages as negative examples in order to reduce the false positive rate.

References

[1] J.R. Bach, C.Fuller, A. Gupta, A. Hampapur, B. Horowitz, R. Humphrey, R.C. Jain, and C. Shu, "Virage image search engine: an open framework for image management", *Symposium on Electronic Imaging: Science and Technology - Storage and Retrieval of Image and Video databases*, Vol. 4, pp. 76–87, IS&T/SPIE, 1996.

[2] I. Cox, M. Miller, S. Omohundro, P. Yianilos, "PicHunter: Bayesian Relevance Feedback for Image Retrieval", *Int. Conf. on Pattern Recognition*, Vienna, Austria, 1996.

[3] M. Flickner et al., "Query by image and video content: The QBIC System", *IEEE Computer*, Vol. 28:9, pp. 23 – 32, 1995.

[4] W. Hsu, T.S. Chua and H.K. Pung, "An integrated color-spatial approach to content-based image retrieval", *Proc. ACM Intern. Conf. Multimedia*, pp. 303–313, 1995.

[5] C.E. Jacobs, A. Finkelstein, and D.H. Salesin, "Fast Multiresolution Image Querying," *ACM, SIGGRAPH, Computer Graphics Proceedings*, pp. 277-286, Los Angeles, CA, 1995.

[6] P. Lipson "Context and Configuration Based Scene Classification", *PhD. Thesis*, MIT, 1996.

[7] T.Minka and R. Picard, "Interactive Learning using a "society of models", *CVPR*, 1996.

[8] S. Baker and S. Nayar, "Pattern Rejection", *CVPR*, pp. 544–549, 1996.

[9] A. Pentland, R. Picard and S. Sclaro, "Photobook: Content-based manipulation of image databases", '*IJCV*, Vol. 18:3, pp. 233–254, 1996.

[10] R.Rickman and J. Stonham, "Content based image retrieval using color-tuple histograms", *SPIE Proceedings*, Vol. 2670, pp. 2–7, 1996.

[11] P. Sinha "Image Invariants for Object Recognition", *Investigative Opthamology and Visual Science*, 1994.

[12] J. Smith and S. Chang, "VisualSEEK: a fully automated content-based image query system", *Proc. ACM Intern. Conf. Multimedia*, Boston, MA, 1996.

[13] J. Smith and S. Chang, "Tools and techniques for color image retrieval", *SPIE Proceedings*, Vol. 2670, pp. 1630–1639, 1996.

[14] M. Stricker and A. Dimai, "Color indexing with weak spatial contstraints", *SPIE Proceedings*, Vol. 2670, pp. 29–40, 1996.

[15] M.J. Swain and D. H.. Ballard, "Color Indexing", *IJCV*, Vol. 7:1, 1991.

[16] G. Pass, R.Zabih, and J. Miller, "Comparing images using color coherence vectors," *Proc. ACM Intern. Conf. Multimedia*, Boston, MA, 1996.

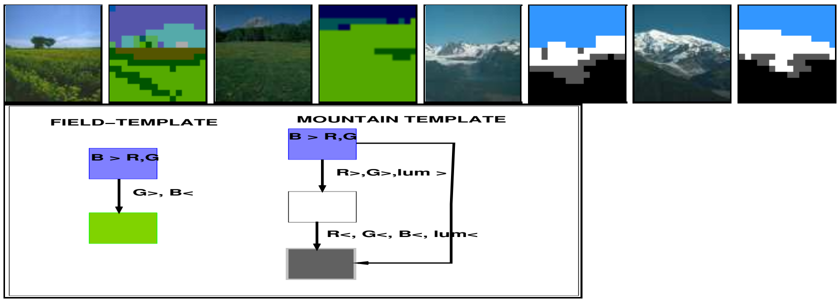

Figure 7: Top Row: Examples of images in fields and snowy-mtns classes and the same low resolution quantized images. Bottom Row: Templates for fields and snowy mountains built from these examples.

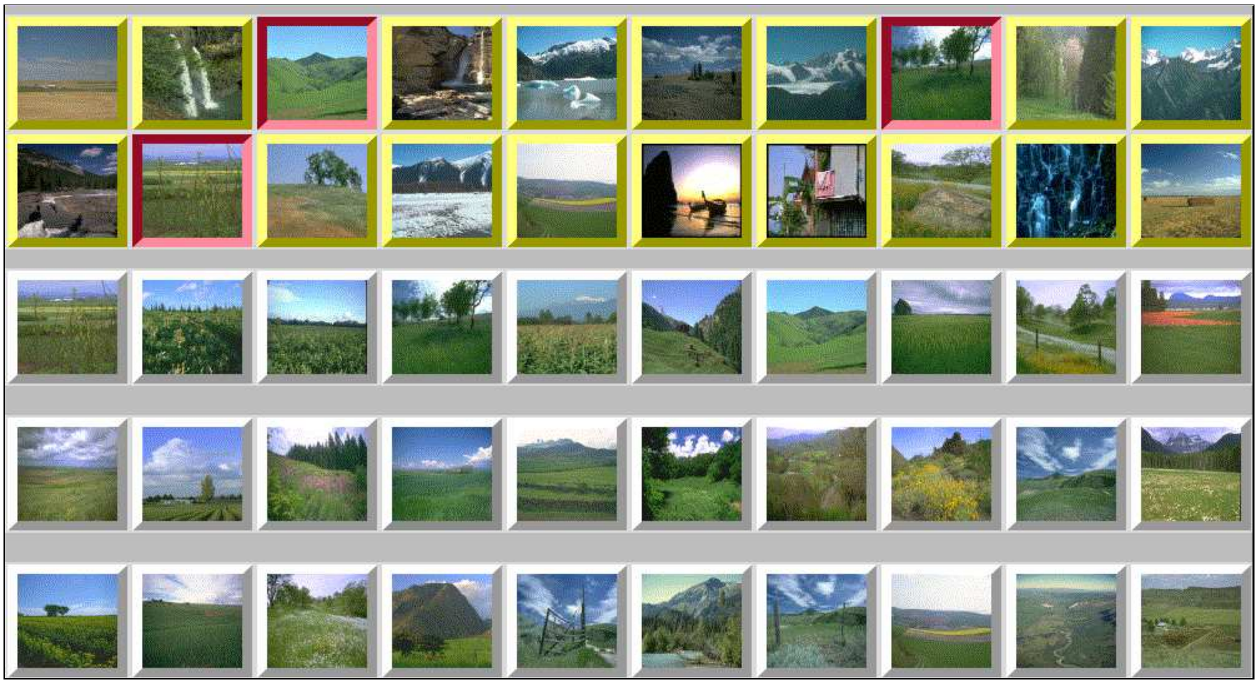

Figure 8: The query images are the ones selected in red in the top two rows. The next 3 rows are the top 30 matches returned by the system.

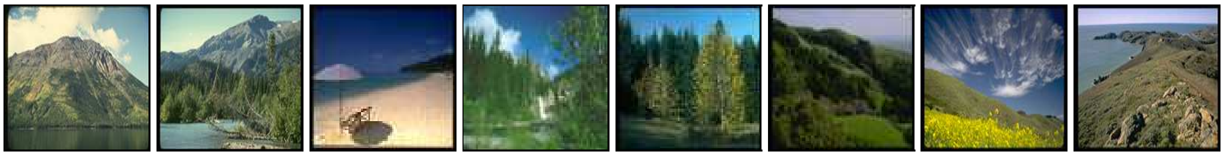

Figure 9: Some examples of the false positives in the top 100 matches.

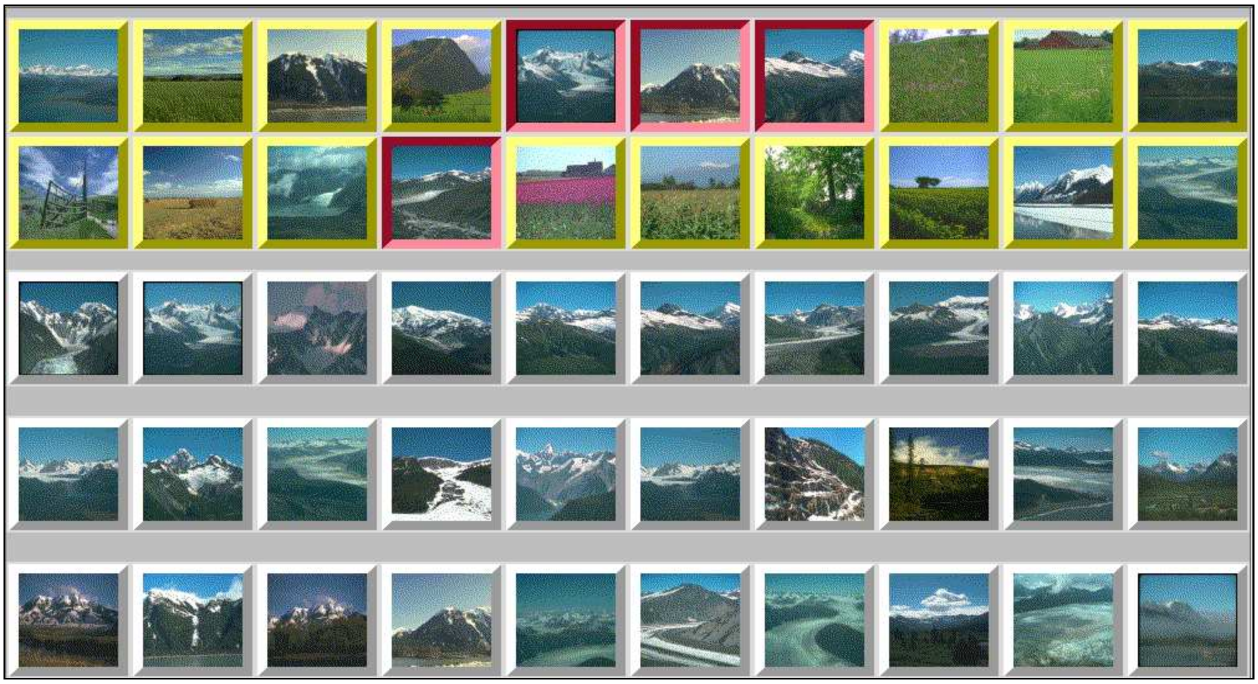

Figure 10: Training images for mountains are the ones selected in red in the top two rows. Middle Rows: Top 30 matches for mountain template

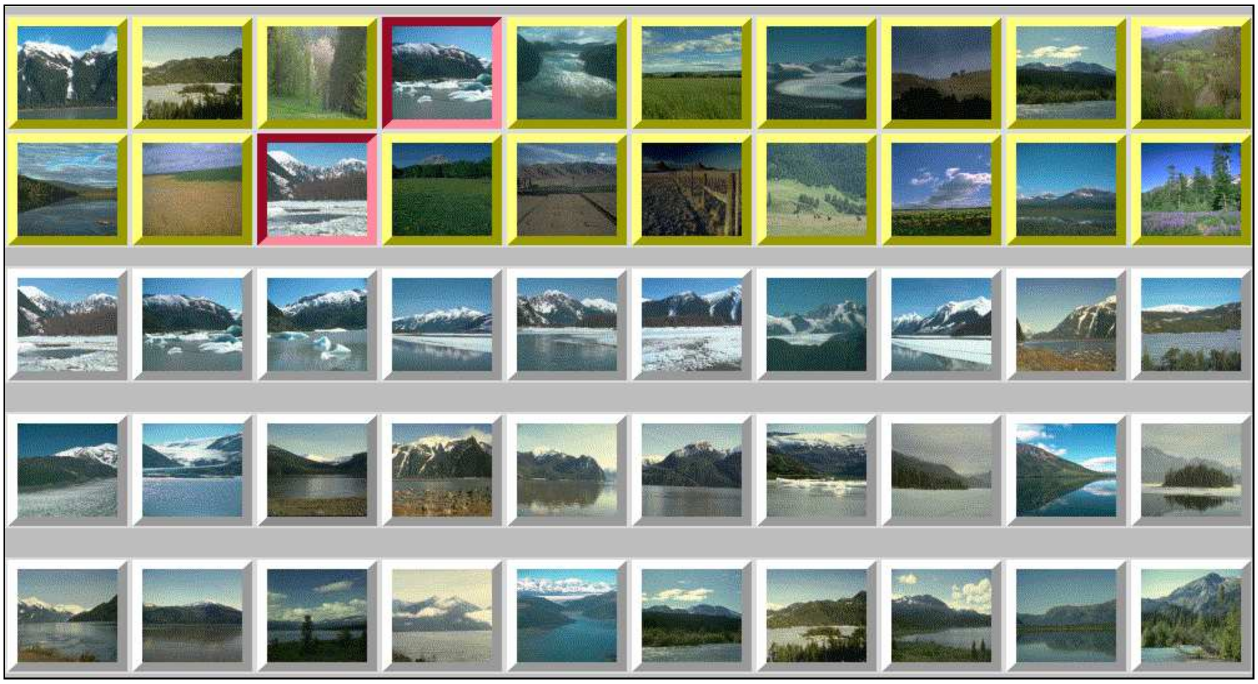

Figure 11: Top: Query images for mountains with icy lakes are the ones selected in red. Top few matches.