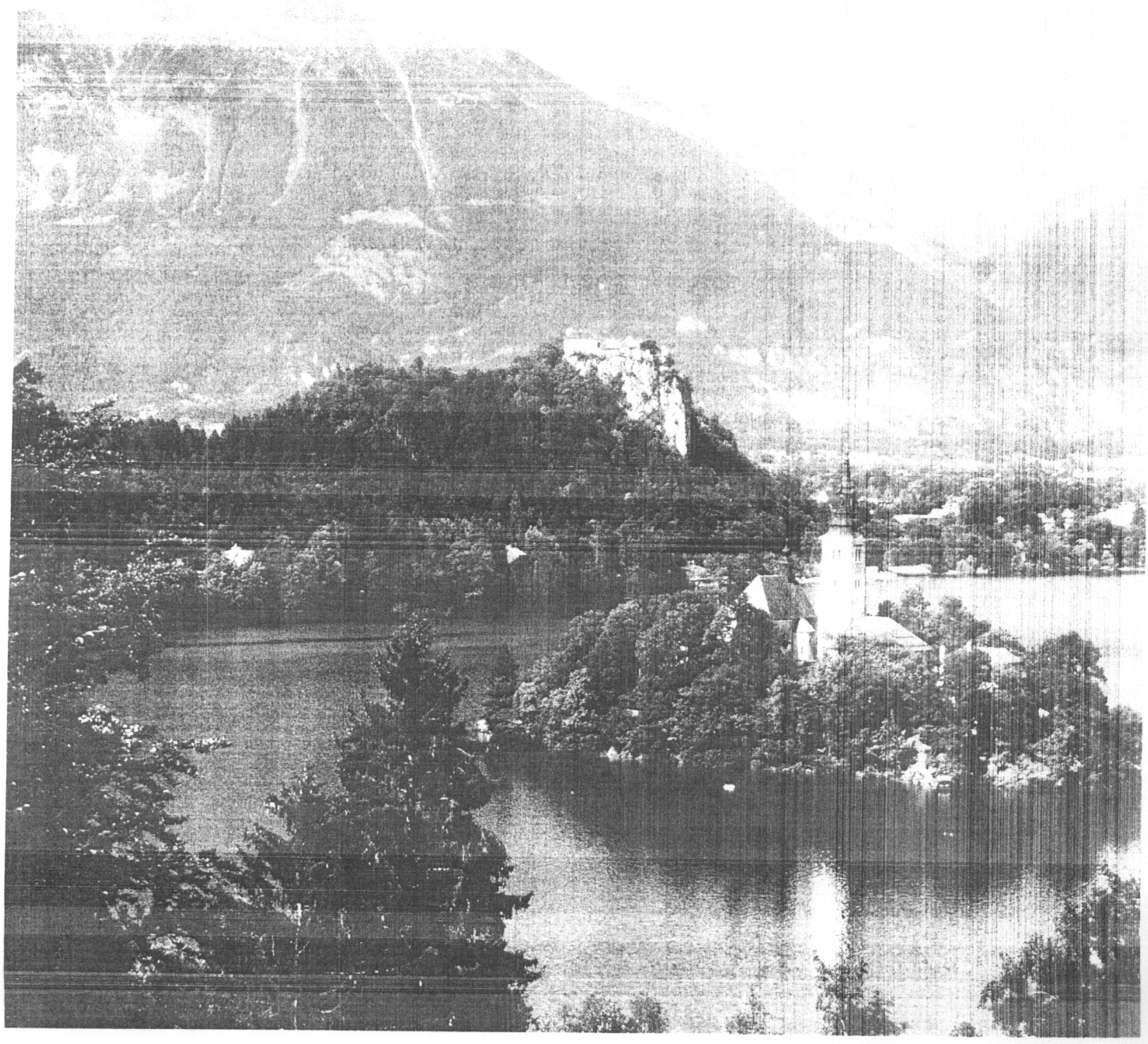
Proceedings of the ICML-99 Workshop on

Machine Learning in Computer Vision

Bled, 30 June 1999

Edited by: Tatjana Zrimec



16th International Conference on Machine Learning

Contents

Machine Learning for Content Based Image Retrieving

Janez Demšar, Franc Solina
Faculty of Computer and Information Sciences, University of Ljubljana
{janez.demsar,franc.solina}@fri.uni-lj.si

Abstract

There exists many different techniques for organization of multimedia collections and retrieval of the collected objects. After discussing the drawbacks of hierarchical and keyword based organization, we study the possibilities for image retrival by using standard machine learning algorithms ID3, Naive Bayesian classifier and k-nearest neighbours. We propose a procedure for usage of such algorithms in this domain. The user is given a random sample of images and grades them by *"NO"*, *"no"*, *"undecided"*, *"yes"* or *"YES"*. The grades are converted into classes and example weights. The learning algorithm predicts the probability for positive class and the system shows the best rated images, which can be graded by the user again and thus added to the initial query. Experiments show the applicability of the method for simple requests, but the inaccuracy of the attribute descriptions disables it to fulfill the more complicated ones.

1 Introduction

The increasing popularity of internet has made the exchange of information of different kinds a trivial task. Among with the ever growing capacities of fast storage media and the appearance of cheap devices and media for permanent data storage, this led to foundations of large personal collections of texts, images, audio and video files. Contrary to the textual databases, multimedia collections are much harder to organize into searchable libraries, which decreases their usefulness.

Personal collections of images are often organized in the way which is directly supported by the operating system, that is by defining a hierarchy of directories reflecting a hierarchy of "themes" and dividing the objects (images, sound files, video files) onto the defined groups. To find a particular object, the user steps down the hierarchy to the desired subgroup and then browses the acquired objects. The simple idea has certain drawbacks. It requires a well defined and extendible structure and a certain amount of work for manual classification of each object, including the objects that arrive later. In addition, such a structure is normally not disjunctive, i.e. one and the same object (or a subtheme) falls into different groups (or themes), which can be confusing. Our personal experience with collections organized like this is that searching through the structure can be quite slow and inefficient.

The other approach is to describe each image by a set of keywords which are later used by the search engine. Both ways are not mutually exclusive; many internet search engines, like Yahoo, offer both at once, searching through the hierarchy and by keywords. The problem with keywords in personal collections is that the average user is not disciplined enough to add the keywords to each new image he/she collects.

To reduce the amount of work needed to create and maintain a collection of images, artificial intelligence methods can be employed. Images in the collection can be described in some language (as vectors of (pre)computed features or by some more complicated descriptions). The idea of our approach is to let the user query by classifying some of the objects as *"wanted"* or *"not wanted"* and the machine learning algorithm tries to learn to distinguish between them. The obtained classifier is then used to observe the images from entire collection and those that most probably have the desired contents are presented to the user.

The user is not required to describe the image in any way, like by placing into a hierarchy or by attaching keywords. Descriptions of collected images are extracted automatically. This "inteligent" approach is not very useful in practice since it is, at the moment, harder to control. The "manual" approaches are, despite their limitations, reliable in the sense that we always know what to expect, while the machine learning approach might perform better but it might also not work at all. Our opinion is, however, that the amount of multimedia data will soon become (or it already is?) infeasible for the manual handling and that the automatic methods must be explored and improved.

This work is limited to collections of images. Also, our intention is not to develop a super-fast and super-accurate image retrieving system but merely to explore the usability of machine learning methods in this area and find the solutions for overcoming the encountered drawbacks. We shall first describe our previous work with a straightforward application of the machine learning algorithm. After discussing the major difficulties, we shall present a new method for querying image databases. The experimental part will try to explore the actual usefulness of the system.

2 Related Work

Existing systems for content based image retrieving (CBIR) generally use attributes that are manually or automatically extracted from images and then stored and managed in conventional database systems [1]. Besides that, precalculated attributes are often too domain specific or too general. Chabot System [2] for example, integrates a relational database containing keyword and other conventional data with color analysis technique to allow searching by keywords and dominant colors. It allows queries as "*mostlyOrange* and *someBlue*" which should, presumably, describe images of sunset over seas and lakes. The problem with this approach is in finding the right combination of attributes; query must be often refined. Also, those queries do not seem to describe the content of the image accurately enough.

Searching of an image database requires the user to select or grade some initial images according to their likeliness to the searched images or to set some boundaries for the values of attributes. Attributes used in this context as well as the distances between the attributes must be fairly simple and fast for computation. Different types of attributes can be used. The most popular are *Color attributes* as they can be computed fast and in a straightforward manner. Color attributes are not sensitive to location, rotation, scale and resolution. On the average they give good results but they miss images which are to a human observer very similar but of different colors. *Texture* [3] is somewhat more difficult to define and compute than color and is more sensitive to resolution. *Shape* (composition, structure) is much more difficult to define and compute than color attributes.

Over the Web several commercial products and research systems for content based image retrieval can be tested. *QBIC (Query by image content)* is an IBM product [4] which is based mostly on color, color layout and texture attributes. *VIRAGE* [5] uses also composition and structure. *MetaSEEK* [6] combines the previous two search systems with the home grown *Vseek* using color and texture.

3 Our Previous Approach and its Limitations

In our first system, we used the machine learning in a straightforward and, as it showed up, inappropriate manner. We took a learning algorithm ID3 and, for the experiment, learned it to distinguish an image, containing a human face from other images. We used a set of simple attributes, mostly describing proportions of basic colors in the image or in the central area of the image. Our addition to the basic ID3 algorithm was a search for informative colors. The system used a local optimization to find a color which could be used as an attribute in a decision tree. In the case of face queries, it usually found a color which we recognized as an approximation for the skin color to be the most informative and used it in decisions of type 'if there is less than 10% of this color, the image does not represent a face'.

We experimented on a set of 167 images, of which 67 were images of a human face and the rest had different contents. Randomly chosen 70% of images were given to the learning algorithm as a learning data and the remaining examples were used for testing. Although results of our experiments measured by the proportion of correctly classified images were good, the method has been quite unsuitable for practical use. We shall list its limitations.

- Due to the fragmentation problem, ID3 is quite sensitive to a small number of examples, comparing to some other algorithms. Our system has been given 70% of 167 images, that is 119 exam-

ples as learning data. No user would be prepared to manually classify such a great number of examples to perform a query.

- The algorithm classified each image as having the desired contents or not having it. Estimating the probabilities for having the desired contents would be more appropriate since it would enable the image browser to sort the images and present them to the user with the 'best' images first, instead of presenting only the images which the classifier guesses to have the desired contents and hiding the others.

- The same holds for the user part: user should not be forced to classify each image as being or not being what he/she searches for. Instead, he/she must be given a chance to grade the images according to how close they are to the desired image. The grades would than be converted to weights of examples. The image with greater positive or negative grade is given a greater weight.

4 A New System for Image Retrieval

Based on experiences from the first system, we built a new system, which we believe to be of practical value. This section describes its details.

4.1 Selection of the Learning Algorithm

ID3 is a strong learning algorithm that presents the knowledge in a 'brain-compatible' form. It is especially popular when we are interested in the obtained decision tree and its explanation. The image retrieval problem is, however, of a different kind. Although we admired the interpretability of the trees derived by our previous system, the user does not really care about it. The knowledge of how the system works is of no interest to the occasional user and there is not a lot of chance that he/she would understand the decision tree or even like to modify it to "refine" the query. This main advantage of ID3 over some other machine learning algorithm is irrelevant for our problem.

From the discussion in previous section it is obvious that we need a learning algorithm with the ability to:

- learn from a small example set,
- predict probability instead of the class,
- handle example weights.

We decided to try out three popular algorithms: ID3, naive Bayesian classifier and k-nearest neighbours.

ID3 is, as already mentioned, quite sensitive to the size of the learning example set. Decision trees can predict probabilities: if the tree is pruned the relative frequency of a class C in the leaf can be used as the estimation for the probability that an example of that leaf is of class C. Alternatively, some other estimate, like Laplace or m-estimate can be used instead of relative frequency on prunned or non-prunned trees. However, in a classical decision tree all the images corresponding to the same leaf are predicted the same probability, which means that the number of different probabilities is limited by the number of leaves, which is in turn limited by the number of examples in the learning set. The decision tree induced from fifteen images would merely divide the whole collection onto at most fifteen groups of images and the whole "most probable" group would have to be presented as the answer to the query. The solution would be to induce more than one decision tree and use some voting technique, or to combine ID3 with some other learning algorithm. The only satisfactory aspect of ID3 is that it can use the weighted examples. This, however, is not an unique feature of this algorithm.

Naive Bayesian classifier is much more robust on small learning example sets. It supports example weighting. It also estimates probabilities instead of predicting classes. The number of different possible probabilities is much larger than that for ID3. Its major limitation is that it cannot handle continuous attributes. Continuous attributes are discretized and then treated as if they were unordered, which discards some possibly of useful information.

K-nearest neighbours uses a distance measure (Euclidean distance, Manhattan distance, or some other) to find the k nearest neighbours of an example which is being classified. The algorithm usually selects the most frequent class among the neighbours as a prediction for the example's class. In our case, we are interested in probabilities of classes so the system returns relative frequency of the class as an estimate for the probability. Examples are also weighted by their prescribed weight and by their distance from the reference example.

From the discussed algorithms, the k-nearest neighbours seems to satisfy the requirements better than the other two.

4.2 The Query Procedure

To avoid the need for manual classification of a larger number of examples when querying, the query examples are added as needed. In the beginning, the system presents the user a small number (say 15) of randomly selected images and asks him/her to grade them. Each image can be given one of five grades, with the lowest meaning that the image is completely different from what he/she looks for and the highest meaning that the image is of exactly the right type. The user is not required to classify all the presented images, but the greater number of images in the query usually means a better accuracy of the answer.

Grades are converted to classes and weights. The images having the middle grade are skipped. The lower two grades are converted to 'NO' class, with the lowest having weight 1 and the other 0.5. The higher two grades correspond to 'YES' class with the highest having weight 1 and the other 0.5. The precalculated attributes together with the just constructed class and weight values are given to the learning algorithm. The obtained classifier is used to estimate the probabilities of 'YES' class for all other images in the database, which have not been presented to the user yet. The fifteen images with the highest probability of 'YES' class are presented to the user as an answer to his/her query and examples for its refinement.

The process than continues by the user classifying the obtained images again. It is hoped (and usually indeed happens) that those fifteen images contain more images that are closer to the desired theme. The grades are submitted again and are added to the previous learning data. The learning algorithm re-learns with the new examples and a new selection of fifteen images is presented to the user again.

To deal with the improving "correctness" of the retrieved images, weights of images from old queries can be gradually decreased. When the system works like this, user can request a larger concept in the beginning and narrow it (become more strict) later, without the good grades from the first rounds of the process interfering in the later rounds.

4.3 Implementation

The learning part of the system is performed by our general machine learning system ML*. ML* is a modular system which incorporates all of the listed learning algorithms, all of them also support example weighting and probability estimation. The program is implemented as a Web's CGI application and can be tested at `http://diana.fri.uni-lj.si/bsq/`.

5 Experiments

This section describes the used attribute set. An example of the query is presented, followed by a more formall assesment of methods successfullness.

5.1 Attributes

The attributes were defined and extracted by Dragan Radolović [8]. We present a brief description of each group.

First and second moment of color histogram, that is the average RGB color and its dispersion, are measured on the whole image and in the central part (middle three fifths) of the image. Although the averaging discards a lot of information, the attribute seems to be quite useful anyway.

Compactness of colors measures the proportion of pixels of 'mostly red', 'mostly green', 'mostly blue' and 'other' colors which are surrounded by pixels of a similar color.

Proportions of basic colors are measured for red, green, blue and 'gray' (that is 'none of these') colors. Each pixel is classified to one of basic colors and the proportion of those corresponded to each color is computed.

The original set of attributes also contained other features such as *the most frequent colors*, which cannot be directly used in machine learning algorithms and were therefore omitted. The used attribute set is rather small and thus leaves a lot of space for future improvements.

5.2 An Example of a Query

Instead of statistical analyses which can distract the attention from actual usability of the system, we tested the system "visually" by using it to retrieve images from a relatively small database containing 1000 images.

Figure 1 shows an example of a query for images of faces and the figure 2 shows the answer. Note that from the fifteen random images that were initialy chosen by the system, only two presented a positive and

one half-positive example, while other images were negative examples. The answer is relatively accurate; only three of fifteen images are complete misses. We can get the next fifteen images with or without refining the query by assigning the grades to the newly presented images.

This result was obtained by using the k-nearest neighbours method with $k = 5$. K-NN was by far the best of the three learning methods tested. The reasons for ID3's failure were already discussed. Naive Bayes classifier's poor performance was probably due to its inability to handle continuous or at least ordered attributes. A drawback of the k-NN method was its slowness in comparison with the other two methods. It could be improved by organizing the database in a more suitable way, like kd-trees.

5.3 Statistical Evaluation

For testing our previous system, we manually classified all the images in the database and then compare our classification by the learning algorithm's predictions. The classification accuracy was measured by the percentage of correctly classified images. This does not work when the classifier returns the probabilities instead of classes. A solution would be to compute the U statistics: if the images are ordered by the probabilities of the correct class, U counts the number of "correct" images before each "incorect" one. For a random sequence of images, U would be distributed approximately by $N(\frac{mn}{2}, \sqrt{\frac{mn(m+n+1)}{12}})$ where m and n are numbers of examples in each class. If the hypothesys $U = \frac{mn}{2}$ can be rejected, it can be concluded that the learning algorithm is better than random and has therefore learned the concept.

This test is still impractical because it requires the manual classification of the test examples. Instead, we used a simple measure based on the user's response to the retrieved images. After the grades are converted to weights, the weights are summed; the grades of positive examples are added and the grades of negative examples are subtracted. For fifteen images, the sum is between -15 (none of the images is of required content) and +15 (all of the images have required content). This measure can be seen absolutely or relatively (with comparison to the initial value, i.e. the grade of the fifteen randomly chosen images). This way, the user himself implicitly asseses the success of the query.

The graphs in Figure 3 show the results of experiments with the k-nearest neighbours algorithm, performed by a test person who was unaware of the underlying algorithm and image descriptors. The test person was asked to perform queries for four themes (except for the faces, she choses the theme herself). For each theme, she performed four searches from the beginning. Each curve on a graph corresponds to one such search and consecutive points of the curve present the consecutive sums of weights.

Images of *faces* are obviously an easy target. Surprisingly, the system was quite successful on images of *animals*, and some progress can be observed even on images of *houses and cityscapes*, which were rare in our collection. Queries for images of *seas and lakes* seem to be unsuccesful.

6 Conclusions and Further Work

Our goal was to adopt the general machine learning methods for the use in image retrieving systems. Different methods were examined and incorporated in a web based search engine.

A theoretical consideration suggests that the most suitable method for searching for images described by the vectors of attributes is the k-nearest neighbours. Experiments confirms this thesys. This is not surprising, the fact is that most of working systems for image retrieval use a simplified version of this method. Its performance could be further improved by refining probability estimation function and how it is influenced by the learning examples of different classes at different distances from the example which is being classified.

Although the system is able to retrieve images belonging to simple 'concepts', the concepts that can be distinguished from each other are much to wide. For example, an image retrieving system is expected to be able to retrieve not just 'images of faces' but at least 'images of female faces' if not even 'images of faces of middle-aged blondes with green eyes, round glasses and not too much make up'. The given attributes do not describe images precisely enough. The future work shall focus mostly on searching for new image describing features. A more sophisticated language, able to express the relations like *"brown area above a green one"* might be needed for this.

Overall, we believe our procedure for incorporating machine learning tools in image retrieval problem is generally useful and could be, by refining the descriptions language, put into a practical use.

Figure 1: An example of a query for human faces. The user gave the highest grade to the two faces and the second highest to the image of a group of people. All other images have the lowest grade.

References

[1] V. N. Gudivada, V. V. Raghavan. Content-Based Image Retrieval Systems. *Computer*, 28:18–12, 1995.

[2] Ogle V. E. (1995) Chabot: Retrieval from a Relational Database of Images, In *Computer 28*, pp. 40-48.

[3] F. Liu, R. Picard. Periodicity, directionality, and randomness: world features for image modeling and retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(7):722–733, 1996.

[4] `http://wwwqbic.almaden.ibm.com/`

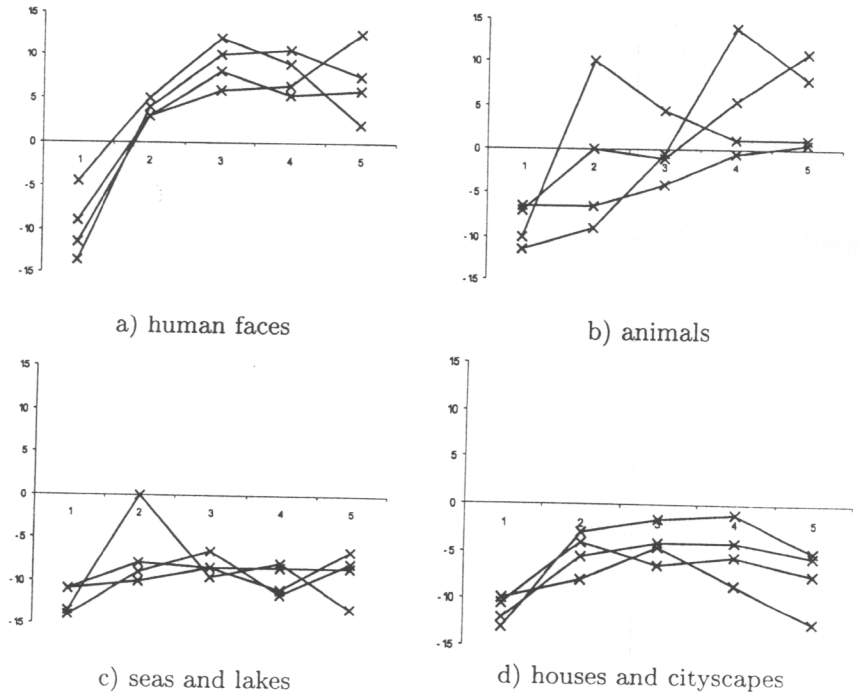
[5] `http://www.virage.com/virdemo.html`

[6] `http://ctr.columbia.edu/metaseek/`

[7] J. Demšar, F. Solina. Using machine learning for content-based image retrieving. In *Proceedings of the 13th International Conference on Pattern Recognition*, volume IV, pages 138–142, Vienna, Austria, August 1996.

[8] Dragan Radolović. *Image database queries based on color information*, B.Sc. Thesis (in Slovene). Faculty of Computer and Information Science, University of Ljubljana, 1998.

Figure 2: Answer of the query from Figure 1. Only three of fifteen images are complete misses, all other images represent human faces.



a) human faces

b) animals

c) seas and lakes

d) houses and cityscapes

Figure 3: Sums of weights from queries for different themes.