

Image annotation using SVM

Claudio Cusano^{*ab}, Gianluigi Ciocca^{*ab}, Raimondo Schettini^{♦a}

^aDISCo, University of Milano Bicocca, Via Bicocca degli Arcimboldi 8, 20126 Milano, Italy
ITC, Consiglio Nazionale delle Ricerche, Via Bassini 15, 20131 Milano, Italy

ABSTRACT

The paper describes an innovative image annotation tool for classifying image regions in one of seven classes - sky, skin, vegetation, snow, water, ground, and buildings - or as unknown. This tool could be productively applied in the management of large image and video databases where a considerable volume of images/frames there must be automatically indexed. The annotation is performed by a classification system based on a multi-class Support Vector Machine. Experimental results on a test set of 200 images are reported and discussed.

Keywords: Image annotation, support vector machines, rejection options, joint histogram

1. INTRODUCTION

Various systems have been proposed for content-based retrieval, and for image classification and indexing^{1,2,3,4,5,6,7,8}. Most of these systems are based on low-level features such as color and texture statistics. We believe that the detection of semantically meaningful regions in digital photographs could be productively employed in classification and retrieval systems for image databases, through the definition of medium-level features. Automatic image annotation would also be useful in image processing algorithms, in intelligent scanners, digital cameras, photocopiers, and printers.

In this paper we propose a tool capable of automatically annotating digital photographs, by assigning the regions to seven different classes: sky, skin, vegetation, snow, water, ground, and buildings. These visual categories mirror high level human perception, permitting the design of intuitive and effective image retrieval strategies. Briefly, our tool works as follows: the images are processed by taking for each pixel a fixed number of partially overlapping image subdivisions (tiles) that contain it, each of which is then independently classified by a multi-class Support Vector Machine (SVM). The results are used to assign the pixel to one of the categories. We have chosen to experiment with Support Vectors Machines because they do not require any distributional assumption about the features, and have provided good generalization accuracy in other imaging applications, even in the case of feature spaces of large dimensions^{9,10}.

The paper is organized as follows. Section 2 briefly presents the SVM framework, with extensions to multi-class classification problems. Section 3 illustrates the database of images used in our experimentation, and the strategies adopted to describe the tiles. In Section 4 we report the experimental results obtained by several Support Vector Machine classifiers. These results are discussed in Section 5 where we then present our plans for future work.

2. SUPPORT VECTOR MACHINES

The SVM methodology comes from the application of statistical learning theory to separating hyperplanes for binary classification problems^{10,11}. The central idea of SVM is to adjust a discriminating function so that it makes optimal use of the separability information of boundary cases. Given a set of cases which belong to one of two classes, training a linear SVM consists in searching for the hyperplane that leaves the largest number of cases of the same class on the same side, while maximizing the distance of both classes from the hyperplane. If the training set is linearly separable, then a separating hyperplane, defined by a normal \mathbf{w} and a bias b , will satisfy the inequalities:

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1, \quad \forall i \in \{1, \dots, N\}, \quad (1)$$

* cusano@itc.cnr.it, *ciocca@itc.cnr.it, ♦schettini@disco.unimib.it

where $\mathbf{x}_i \in \mathfrak{R}^d$ is a case of the training set ($i = 1, \dots, N$), d being the dimension of the input space, and $y_i \in \{-1, +1\}$ is the corresponding class. Among the separating hyperplanes, the SVM approach selects the one for which the distance to the closest case is maximal. Since such a distance is $1/\|\mathbf{w}\|$, finding the optimal hyperplane is equivalent to minimizing $\|\mathbf{w}\|^2$ under constraints (1). The cases closest to the hyperplane are called Support Vectors, while the quantity $2/\|\mathbf{w}\|$ is called the margin (see Figure 1). This can be considered a measure of the generalization ability of the SVM: the larger the margin, the better the generalization is expected to be.

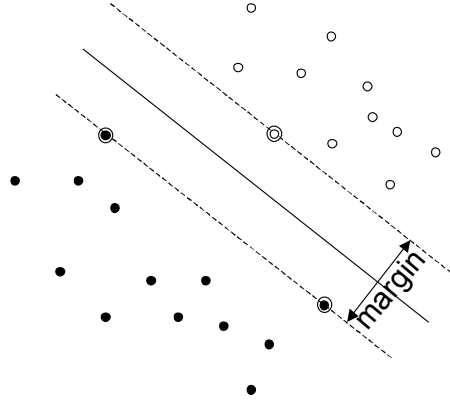


Figure 1. Separating hyperplane. The Support Vectors are circled.

When the training set is not linearly separable (i.e. when no hyperplane satisfies the constraints (1)), the optimal separating hyperplane is found, solving an optimization problem relaxed by introducing a set of slack variables \mathbf{e} and a penalization for cases that are misclassified or inside the margin. This leads to the minimization of the following objective function, with respect to \mathbf{w} , b , and \mathbf{e}

$$\min \left(\frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^N \mathbf{e}_i \right), \quad (2)$$

under the constraints

$$y_i (\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 - \mathbf{e}_i, \quad \mathbf{e}_i \geq 0, \quad \forall i \in \{1, \dots, N\}. \quad (3)$$

Parameter C determines a trade-off between the error on the training set and the separation of the two classes.

Before introducing a non-linear extension to SVMs, we must reformulate the linear problem in its dual form, introducing a Lagrange multiplier for each constraint¹². The dual problem consists in maximizing the following objective function, with respect to the Lagrange multipliers α

$$\max \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j (\mathbf{x}_i \cdot \mathbf{x}_j), \quad (4)$$

under the constraints

$$\sum_{i=1}^N \alpha_i y_i = 0, \quad (5)$$

$$0 \leq \alpha_i \leq C, \quad \forall i \in \{1, \dots, N\}. \quad (6)$$

The label g , assigned to a new case \mathbf{x} , can be obtained without explicitly computing \mathbf{w} :

$$g(\mathbf{x}) = \text{sgn}\left(b + \sum_{i=1}^N \alpha_i y_i (\mathbf{x}_i \cdot \mathbf{x})\right), \quad (7)$$

and the value of b can be determined using any case \mathbf{x}_j in the training set, provided its multiplier α_j is not at the bounds 0 or C :

$$b = y_j - \sum_{i=1}^N \alpha_i y_i (\mathbf{x}_i \cdot \mathbf{x}_j), \quad (8)$$

The SVM approach can be extended to non-linear decision surfaces through a non-linear function Φ which maps the original feature space \mathfrak{R}^d into a higher dimensional space H . Since the only operation needed on H is the inner product, if we have a kernel function k :

$$k : \mathfrak{R}^d \times \mathfrak{R}^d \rightarrow \mathfrak{R}, \quad k(\mathbf{x}', \mathbf{x}'') = \Phi(\mathbf{x}') \cdot \Phi(\mathbf{x}''), \quad (9)$$

mapping Φ is never explicitly used. Given a kernel function, and provided some non stringent conditions are fulfilled, the existence of a space H and a mapping Φ is guaranteed.

In non-linear SVM, the kernel function k replaces the inner products in Equations 4, 7, and 8, and the objective function becomes

$$\max \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i \cdot \mathbf{x}_j). \quad (10)$$

Examples of widely used kernel functions are the polynomial kernel:

$$k(\mathbf{x}', \mathbf{x}'') = (\mathbf{x}' \cdot \mathbf{x}'' + 1)^p, \quad (11)$$

and the gaussian kernel:

$$k(\mathbf{x}', \mathbf{x}'') = \exp\left(-\frac{\|\mathbf{x}' - \mathbf{x}''\|^2}{\sigma}\right), \quad (12)$$

where p (a positive integer) and σ (a positive real) are parameters of the kernels.

2.1 Multi-class SVM

Although SVMs are mainly designed for the discrimination of two classes, they can be adapted to multi-class problems. A multi-class SVM classifier can be obtained by training several classifiers and combining their results. There are several strategies for combining SVMs; two common methods are “one per class” and “pairwise coupling”^{13,14}. The former consists in training one classifier for each class to discriminate between that class and the other classes. Each classifier defines a discrimination function f_i that should assume positive values when the cases belong to the class i and negative values otherwise. These values are then compared; and the output of the combined classifier is the index i for which the value of the discriminating function f_i is the largest. The most commonly used discrimination function is the signed distance between the case to classify and the hyperplane

$$f_i(\mathbf{x}) = \frac{\mathbf{w}_i \cdot \mathbf{x} + b_i}{\|\mathbf{w}_i\|}, \quad (13)$$

where \mathbf{w}_i and b_i are, respectively, the normal and the bias of the hyperplane related to the i -th SVM. The label c , assigned to a case \mathbf{x} , by the multi-class SVM is:

$$c(\mathbf{x}) = \arg \max_{i \in \{1, \dots, K\}} f_i(\mathbf{x}), \quad (14)$$

where K is the number of classes.

The pairwise coupling method trains $K(K-1)/2$ binary SVM classifiers. Each of these is a SVM trained to discriminate between two classes. To classify a case, the pairwise coupling method combines the discrimination functions of these $K(K-1)/2$ classifiers, using some voting scheme. When the number of the classes is high the pairwise coupling method requires the training of a huge number of SVM.

3. DATA SELECTION AND DESCRIPTION

We applied different SVM-based classification strategies and feature sets to semantically classify regions in digital photographs. The seven classes considered - sky, skin, vegetation, snow, water, ground, and buildings - are briefly defined in Table 1.

Class	Description
Buildings	Man made structures, such as houses, vehicles, ...
Ground	Terrain: sand, rocks, beaches ...
Skin	Skin of different human races: caucasian, asian, negroid
Sky	Sky, including clouds and sun
Snow	Snow and ice
Vegetation	Grass, trees, woods and forests
Water	Sea, oceans, lakes, rivers ...

Table 1. Description of the classes.

The set of the images used for our experiments consisted of 500 photographs downloaded from the Web, or acquired by scanner or digital cameras. The images varied widely in compression ratio and size, ranging from about 75000 pixels to 1,4 million pixels; most of them were outdoor photographs. Each of the seven classes were well represented, although sky was the most frequent, since almost every outdoor photograph contains a region of this kind. The dataset was randomly subdivided into a training set of 300 photographs, and a test set composed of the remaining 200 images. The salient regions of the images in the training and in the test set have been manually labeled with the correct class.

Our tool has been designed to classify image tiles, that is, square subdivisions of the image the size of a fixed fraction of the total area of the image. The length l of the side of a tile of an image of width w and height h , is computed as

$$l = \sqrt{p \cdot w \cdot h}, \quad (15)$$

meaning that the area of a tile is p times the area of the whole image (here $p=0.004$).

Two sets of tiles were selected at random from the training set and the test set, and used respectively for the learning and for the validation of the classifier. More in detail, we first drew an image, then a region of that image, and, finally, we selected at random a tile that was entirely inside that region. This process was repeated until we had 1500 tiles for each class extracted from the training set, and 1500 for each class extracted from the test set. At the end of the selection process we had two sets of tiles, consisting of $1500 \times 7 = 10500$ tiles each.

3.1 Tiles description

Before submitting a tile to a classifier, in the training or in the testing phase, we computed a description of it in terms of low-level features. We used histograms as feature vectors, because they are very simple to compute and have given good results in practical applications, where feature extraction must be as simple and rapid as possible¹⁵. We initially used a simple histogram based on the quantization of the HSV color space in eleven bins¹⁶ (see Table 2).

H_{min}	H_{max}	S_{min}	S_{max}	V_{min}	V_{max}	Color
0	360	0	15	0	31	Black
0	360	0	15	32	69	Gray
0	360	0	15	70	100	White
-18	18	16	100	32	100	Red
19	40	16	100	32	100	Orange
41	62	16	100	32	100	Yellow
63	158	16	100	32	100	Green
159	208	16	100	32	100	Cyan
209	288	16	100	32	100	Blue
289	330	16	100	32	100	Magenta
331	341	16	100	32	100	Pink

Table 2. Partition of the HSV color space: hue ranges from 0 to 359, and is an angular measure (so, -18=342). Saturation and value are in the range 0 - 100. The color column reports the name of some average color in the bin.

Due to their properties of efficiency and invariance properties in rotation and translation, color histograms are widely used for content-based image indexing and retrieval. However, a color histogram only records color distribution; other potentially useful properties of the images are lost. We have used what is called a joint histogram¹⁷. This is a multidimensional histogram which can incorporate additional information about color distribution, edge and texture statistics and any kind of local pixel feature without sacrificing the robustness of color histograms. Every entry in a joint histogram contains the fraction of pixels in the image that are described by a particular combination of feature values. feature. The joint histogram used here combines color distribution with gradient statistics. Color distribution is described by the quantization of the HSV color space. The horizontal and vertical components of the gradient are computed by the application of Sobel’s filters to the luminance image. For each component, the absolute value is taken and then quantized in four bins on the basis of comparison with three thresholds. The thresholds have been selected taking the 0.25, 0.50, and 0.75 quantiles of the distribution of the absolute value of gradient components, estimated by a random selection of over two million pixels in the images of the training set (see Table 3).

Component of the gradient	Quantized value
$ g < 0.02$	<i>Null gradient</i>
$0.02 \leq g < 0.06$	<i>Low gradient</i>
$0.06 \leq g < 0.17$	<i>Medium gradient</i>
$ g \geq 0.17$	<i>High gradient</i>

Table 3. Quantization of the gradient of the images: the thresholds are the same for the horizontal and the vertical component. Here, the luminance of a pixels ranges from 0 to 1.

Since the joint histogram combines information regarding color and gradient, the number of bins of the joint histograms is $11 \times 4 \times 4 = 176$. Although the dimension of this joint histogram is quite small (compared with the typical joint histogram of thousands of bins), we think that the information conveyed can suffice for applications of image annotation.

4. EXPERIMENTAL RESULTS

For classification, we used a multi-class SVM, constructed according to “one per class” strategy. To train each of the seven SVM, we used the 1500 tiles of the related class, taken from the training set, and a random selection of 1500 tiles of the other classes. Each SVM was thus trained to discriminate between one class and the others. In the test phase we used all the 10500 tiles extracted from the test set to evaluate the performance of the classification engine. The first classifier tested was a simple linear multi-class SVM, that is, the seven binary classifiers were linear SVM. In our first set of experiments we used a penalization coefficient C of 60 and an HSV histogram with eleven bins as feature vectors. Consequently, all the information carried by the feature vectors was related to color distribution. Table 4 shows the confusion matrix of the first classifier on the test set.

		Predicted class						
		Buildings	Ground	Skin	Sky	Snow	Vegetation	Water
True class	Buildings	<u>0.2653</u>	0.2493	0.1527	0.0000	0.0953	0.2040	0.0333
	Ground	0.1073	<u>0.1307</u>	0.4680	0.0000	0.2073	0.0660	0.0207
	Skin	0.0080	0.0213	<u>0.9553</u>	0.0000	0.0153	0.0000	0.0000
	Sky	0.0113	0.0007	0.0013	<u>0.0067</u>	0.2867	0.0020	0.6913
	Snow	0.0320	0.0093	0.0260	0.0060	<u>0.7887</u>	0.0007	0.1373
	Vegetation	0.1393	0.0853	0.0447	0.0000	0.1427	<u>0.4880</u>	0.1000
	Water	0.0140	0.0093	0.0120	0.0027	0.1307	0.0153	<u>0.8160</u>

Table 4. Confusion matrix of a linear multi-class SVM trained with the HSV histogram as feature vectors. The penalization coefficient C is set to 60. The values are computed on the test set.

Although the results are quite different from those of random guessing, they are still far from satisfactory. Further experimentation with different values of the penalization coefficient did not bring the generalization accuracy of the classifier to an acceptable level. Slightly better results have been achieved by non-linear multi-class SVM with a gaussian kernel; the confusion matrix of this classifier for the test set is reported in Table 5.

		Predicted class						
		Buildings	Ground	Skin	Sky	Snow	Vegetation	Water
True class	Buildings	<u>0.4193</u>	0.1527	0.1520	0.0073	0.0220	0.2273	0.0193
	Ground	0.1153	<u>0.1727</u>	0.4660	0.0213	0.0980	0.1040	0.0227
	Skin	0.0033	0.0527	<u>0.9393</u>	0.0000	0.0040	0.0007	0.0000
	Sky	0.0007	0.0053	0.0013	<u>0.2687</u>	0.0533	0.0187	0.6520
	Snow	0.0313	0.0440	0.0153	0.4940	<u>0.2613</u>	0.0413	0.1127
	Vegetation	0.0520	0.2367	0.0500	0.0020	0.0213	<u>0.5647</u>	0.0733
	Water	0.0033	0.0120	0.0140	0.0373	0.1380	0.0367	<u>0.7587</u>

Table 5. Confusion matrix of a non-linear multi-class SVM trained with the HSV histogram as feature vectors. The penalization coefficient C is set to 50, and the kernel parameter σ is set to 0.8. The values are computed on the test set.

The overall error is nearly the same, but the misclassification probability is now more uniform among the classes. We selected the penalization coefficient and the kernel parameter empirically after several tests.

Since the results obtained with several classifiers, varying kernels and parameters, were not satisfactory, we considered a more complex description of the tiles. In our next experiments we used a joint histogram as feature vector, combining information related to color and gradient statistics. The results obtained on the test set using a linear multi-class SVM are reported in the following table.

		Predicted class						
		Buildings	Ground	Skin	Sky	Snow	Vegetation	Water
True class	Buildings	<u>0.3687</u>	0.1560	0.0533	0.0213	0.1513	0.1793	0.0700
	Ground	0.1473	<u>0.3347</u>	0.2393	0.0093	0.0733	0.1933	0.0027
	Skin	0.0193	0.0460	<u>0.9167</u>	0.0007	0.0167	0.0007	0.0000
	Sky	0.0047	0.0000	0.0000	<u>0.7787</u>	0.1180	0.0027	0.0960
	Snow	0.0913	0.0227	0.0040	0.0793	<u>0.6920</u>	0.0233	0.0873
	Vegetation	0.0880	0.1127	0.0580	0.0153	0.0133	<u>0.6787</u>	0.0340
	Water	0.0360	0.0413	0.0360	0.0980	0.1067	0.0380	<u>0.6440</u>

Table 6. Confusion matrix of a linear multi-class SVM trained with the joint histogram as feature vectors. The penalization coefficient C is set to 100. The values are computed on the test set.

The classifier obtained using the joint histogram is far more accurate than the classifiers based on the simple color histogram, but the results are still far from an acceptable level. Our next step was to train a non-linear classifier based on the joint histogram. The results on the test set are reported in Table 7.

		Predicted class						
		Buildings	Ground	Skin	Sky	Snow	Vegetation	Water
True class	Buildings	<u>0.8447</u>	0.0453	0.0160	0.0107	0.0227	0.0447	0.0160
	Ground	0.0287	<u>0.8660</u>	0.0300	0.0080	0.0113	0.0547	0.0013
	Skin	0.0000	0.0040	<u>0.9947</u>	0.0000	0.0007	0.0007	0.0000
	Sky	0.0013	0.0013	0.0000	<u>0.9347</u>	0.0267	0.0000	0.0360
	Snow	0.0133	0.0080	0.0007	0.0547	<u>0.9000</u>	0.0067	0.0167
	Vegetation	0.0353	0.0500	0.0120	0.0053	0.0053	<u>0.8813</u>	0.0107
	Water	0.0100	0.0187	0.0060	0.0380	0.0360	0.0100	<u>0.8813</u>

Table 7. Confusion matrix of a non-linear multi-class SVM trained with the HSV histogram as feature vectors. The penalization coefficient C is set to 25, and the kernel parameter σ is set to 0.1. The values are computed on the test set.

The improvement with respect to the linear classifier is quite impressive: the accuracy in the classification has increased for each class (more than 50% for the ground class). Some errors are still present, ground tiles in particular have often been misclassified as vegetation tiles (5,4% of cases) and vice-versa (5%), snow tiles have been misclassified as sky tiles (5,4%), and buildings tiles as ground tiles (4,5%).

4.1 The image annotation tool

After the satisfactory training of a classifier, we designed a strategy for annotating whole images. In order to label each pixel of the image as belonging to one of the classes, we needed a way to select the tiles and then combine multiple classification results. In our approach, the tiles are sampled at fixed intervals. Since several tiles overlap, every pixel of the image is found in a given number of tiles. Each tile is independently classified, and the pixel's final label is decided by majority vote. The size of the tiles is determined by applying Equation 15.

Frequently an area of the image cannot be labeled with one of the seven classes selected, and in this case different classes are often assigned to overlapping tiles. To correct this kind of error and to achieve in general, a more reliable annotation strategy, we introduced a rejection option: when the fraction of concordant votes related to overlapping tiles lies below a certain threshold, the pixels inside these tiles are labeled as belonging to an unknown class. In practice, the rejection option selects the pixels that cannot be assigned to any class with sufficient confidence.

Our method of annotation has performed with success on the 200 images in the test set. The tiles are sampled at one fifth of their size in both the x and y dimensions. As a result the same pixels are found in 25 tiles. Pixels on the image borders are classified using only the available tiles. Figures 2,3,4, and 5 show the annotations obtained applying this tool to some images of the test set, using the non-linear classifier, joint histograms, and a rejection threshold of 0.65.

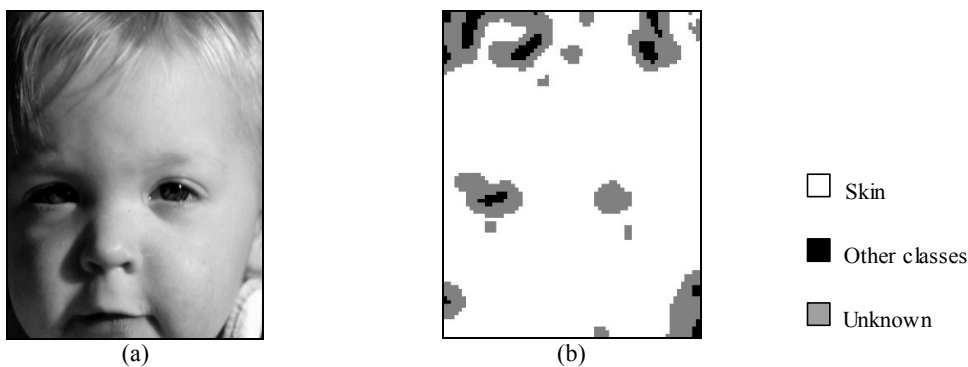


Figure 2. a) original image, b) annotated image.

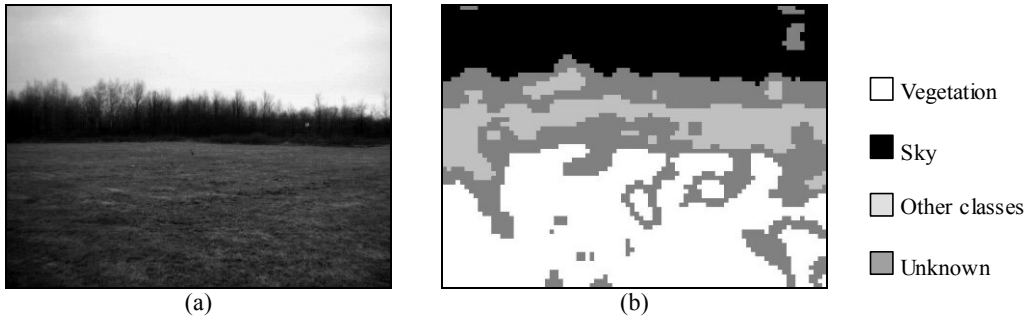


Figure 3. a) original image, b) annotated image.

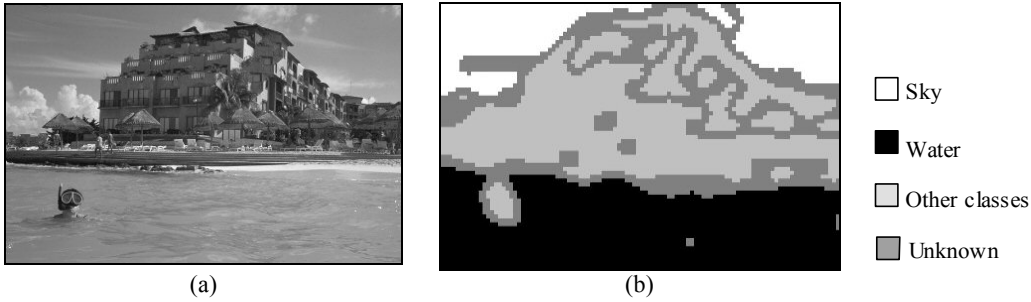


Figure 4. a) original image, b) annotated image.

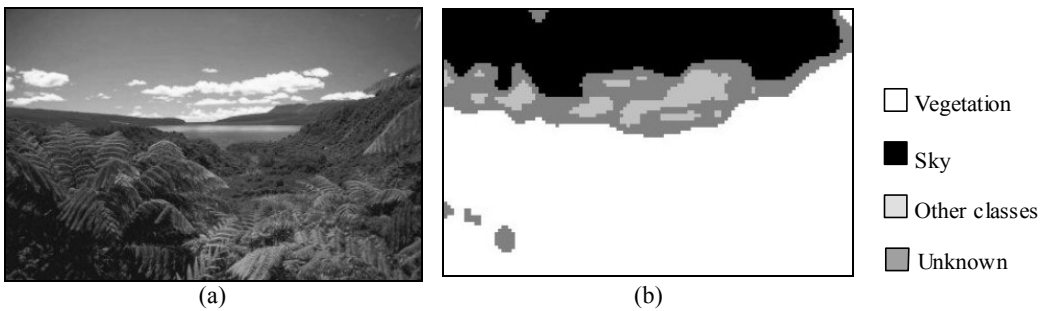


Figure 5. a) original image, b) annotated image.

5. CONCLUSIONS

The experimental results presented here demonstrate that our tool for image annotation can be applied with sufficient reliability to content-based retrieval in image and video databases. Although the accuracy of the tool is quite satisfactory, we plan to further refine the whole strategy. The misclassified pixels are generally on the boundary between two classes, or inside very dark areas, where the information about color and texture is insufficient (see, for example, the trees in Figure 3). The types of the error made by the tool suggests that it cannot be used as a segmentation tool: a more specific segmentation strategy would be needed for this purpose. We also believe that a richest description of the tiles would improve the performances of the classifiers significantly. This will be the main topic of our future research in the area.

REFERENCES

1. C. Faloutsos, R. Barber, M. Flickner, J. Hafner, W. Niblack, D. Petkovic, and W. Equitz, "Efficient and effective querying by image content", *Journal of Intelligent Information Systems*, **3**, pp. 231-262, 1994.
2. A. Pentland, R. W. Picard, and S. Sclaroff, "Photobook: content-based manipulation of image databases", *Proc. SPIE Storage and Retrieval for Image and Video Databases II*, **2185**, pp. 34-47, 1994.

3. A. Hampapur, A. Gupta, B. Horowitz, C. F. Shu, C. Fuller, J. Bach, M. Gorkani, and R. Jain, "Virage video engine", *Proc. SPIE Storage and Retrieval for Image and Video Databases V*, pp. 188-197, 1997.
4. J. R. Smith and S.F. Chang, "Visualeek: a fully automated content-based image query system", *Proc. ACM Multimedia*, pp. 87-98, 1996.
5. S. Mehrotra, Y. Rui, M. Ortega, and T.S. Huang, "Supporting content-based queries over images in MARS", *Proc. IEEE International Conference on Multimedia Computing and Systems*, pp. 632-633, 1997.
6. G. Ciocca, I. Gagliardi, R. Schettini, "Quicklook2: an integrated multimedia system", *International Journal of Visual Languages and Computing, Special issue on Querying Multiple Data Sources*, **12**, pp. 81-103, 2001.
7. A. Vailaya, M. Figueiredo, A. K. Jain, and H. J. Zhang, "Image classification for content-based indexing", *IEEE Transaction on Image Processing*, **10(1)**, pp.117-130, 2001.
8. R. Lienhart and A. Hartmann, "Classifying images on the web automatically", *Journal of electronic imaging*, **11(4)**, pp. 445-454, 2002.
9. T. Hastie, R. Tibshirani, J. Friedman, *The Elements of Statistical Learning*, Springer, 2001.
10. V. Vapnik, *The Nature of Statistical Learning Theory*, Springer, 1995.
11. C. Cortes and V. Vapnik, "Support-Vector Networks", *Machine Learning*, **20(3)**, pp. 273-297, 1995.
12. K. R. Muller, S. Mika, G. Ratsch, K. Tsuda, and B. Scholkopf, "An introduction to kernel-based learning algorithms", *IEEE Transactions on Neural Networks*, **12(2)**, pp. 181-201, 2001.
13. J. Weston and C. Watkins, "Support vector machines for multiclass pattern recognition", *Proc. Seventh European Symposium On Artificial Neural Networks*, 1999.
14. K. Goh, E. Chang, and K. Cheng, "Support vector machine pairwise classifiers with error reduction for image classification", *Proc. ACM workshops on Multimedia: multimedia information retrieval*, pp. 32-37, 2001.
15. M. Stricker, and M. Swain, "The Capacity of Color Histogram Indexing", *Computer Vision and Pattern Recognition*, pp. 704-708, 1994.
16. I. J. Cox, M. L. Miller, S. M. Omohundro, and P.N. Yianilos, "Target Testing and the PicHunter Bayesian Multimedia Retrieval System", *Advances in Digital Libraries*, pp. 66-75, 1996.
17. G. Pass and R. Zabih, "Comparing Images Using Joint Histograms", *Multimedia Systems*, **7(3)**, pp. 234-240, 1999.