Automating the Construction of Scene Classifiers for Content-Based Video Retrieval

Menno Israël
ParaBot Services b.v.
Singel 160
1054 TA Amsterdam
The Netherlands
m.israel@parabots.nl

Egon L. van den Broek
NICI, University of Nijmegen
P.O. Box 9104
6500 HE Nijmegen
The Netherlands
e.vandenbroek@nici.kun.nl

Peter van der Putten
LIACS, University of Leiden
P.O. Box 9512
2300 RA Leiden
The Netherlands
putten@liacs.nl

ABSTRACT

This paper introduces a real time automatic scene classifier within content-based video retrieval. In our envisioned approach end users like documentalists, not image processing experts, build classifiers interactively, by simply indicating positive examples of a scene. Classification consists of a two stage procedure. First, small image fragments called patches are classified. Second, frequency vectors of these patch classifications are fed into a second classifier for global scene classification (e.g., city, portraits, or countryside). The first stage classifiers can be seen as a set of highly specialized, learned feature detectors, as an alternative to letting an image processing expert determine features a priori. We present results for experiments on a variety of patch and image classes. The scene classifier has been used successfully within television archives and for Internet porn filtering.

Categories and Subject Descriptors

H.3.1 [**Information storage and retrieval**]: Content Analysis and Indexing - Indexing methods. H.3.3 [**Information storage and retrieval**]: Information Search and Retrieval - Retrieval models. I.2.10 [**Artificial Intelligence**]: Vision and Scene Understanding - Intensity, color, photometry, and thresholding, Texture, Video analysis. I.4.7 [**Image processing and computer vision**]: Feature Measurement - Feature representation, Moments, Texture. I.4.8 [**Image processing and computer vision**]: Scene Analysis - Color. I.4.9 [Image processing and computer vision]: Applications. I.5.1 [Pattern Recognition]: Models - Neural Nets. I.5.4 [**Pattern Recognition**]: Applications - Computer vision.

General Terms

Algorithms, Performance, Experimentation, Human Factors.

MDM/KDD'04, August 22, 2004, Seattle, WA, USA.

Keywords

Content-Based Video Retrieval, CBVR, Content-Based Image Retrieval, CBIR, Image Mining, Vicar, Scene Classification, Patch Classification, Color, Texture, real time, automatic annotation, television archives, porn filtering.

1. INTRODUCTION

This work has been done as part of the EU Vicar project (IST). The aim of this project was to develop a real time automated video indexing, classification, annotation, and retrieval system. Vicar was developed in close cooperation with leading German, Austrian, Swedish, and Dutch broadcasting companies. These companies generally store millions of hours of video material in their archives. To increase sales and reuse of this material, efficient and effective video search with optimal hit rates is essential. Outside the archive, large amounts of video material are managed as well, such as news feeds and raw footage [35].

Generally, only a fraction of the content is annotated manually and these descriptions are typically rather compact. Any system to support video search must be able to index, classify, and annotate the material extensively, so that efficient mining and search may be conducted using the index rather than the video itself. Furthermore, these indices, classifications, and annotations must abstract from the pure syntactical appearance of the video pixels to capture the semantics of what the video is about (e.g., a shot of Madonna jogging in a park).

Within Vicar a variety of visual events is recognized, including shots, camera motion, person motion, persons and faces, specific objects, etc. In this paper we will focus on the automated classification of visual scenes. For searching and browsing video scenes, classifiers that extract the background setting in which events take place are a key component. Examples of scenes are indoor, outdoor, day, night, countryside, city, demonstration, and so on. The amount of classes to be learned is generally quite large - tens to hundreds - and not known beforehand. So, it is generally not feasible to let an image processing expert build a special purpose classifier for each class.

Using our envisioned approach, an end user like an archive documentalist or a video editor can build classifiers by simply showing positive examples of a specific scene category.

In addition, an end user may also construct classifiers for small image fragments to simplify the detection of high level global scenes, again just by showing examples (e.g., trees, buildings, and road).

We call these image fragments patches. The patch classifiers actually provide the input for the classification of the scene as a whole. The patch classifiers can be seen as automatically trained data preprocessors generating semantically rich features, highly relevant to the global scenes to be classified, as an alternative to an image processing expert selecting the right set of abstract features (e.g., wavelets, Fourier transforms). Additionally, the interactive procedure is a way to exploit a priori knowledge, the documentalist may have about the real world, rather than relying on a purely data driven approach.

Note that the scene is classified without relying on explicit object recognition. This is important because a usable indexing system should run at least an order of magnitude faster than real time, whereas object recognition is computationally intensive. More fundamentally, we believe that certain classes of semantically rich information can be perceived directly from the video stream rather than indirectly by building on a large number of lower levels of slowly increasing complexity. This position is inspired by Gibson's ideas on direct perception [14]. Gibson claims that even simple animals may be able to pick up niche specific and complex observations (e.g., prey or predator) directly from the input without going through several indirect stages of abstract processing.

This paper is expository and meant to give a non-technical introduction into our methodology. A high level overview of our approach is given in Section 2. Section 3 provides more detail on the low level color and texture features used and Section 4 specifies the classifying algorithms used. Experimental results for patch and scene classification are given in Sections 4.1 and 4.2. Next, we highlight two applications in which scene classification technology has been embedded (Section 6). We finish with a discussion and conclusion (Sections 5 and 7).

2. OVERALL APPROACH

In Vicar a separate module is responsible for detecting the breaks between shots. Then for each shot a small number of representative key frames is extracted, thus generating a storyboard of the video. These frames (or a small section of video around these key frames) are input to the scene classifier.

2.1 Scene Classification Procedure

The scene classifier essentially follows a two stage procedure: (i) Small image segments are classified into patch categories (e.g., trees, buildings, and road) and (ii) these classifications are used to classify the scene of the picture as a whole (e.g., interior, street and forest). The patch classes that are recognized can be seen as an alphabet of basic perceptual elements to describe the picture as a whole.

In more detail, first a high level segmentation of the image takes place. This could be some intelligent procedure recognizing arbitrarily shaped segments, but for our purposes we simply divide images up into a regular n by m grid, say 3 by 2 grid segments for instance. Next, from each segment patches (i.e., groups of adjacent pixels within an image, described by a specific local pixel distribution, brightness, and color) are sampled. Again, some intelligent sampling mechanism could be used to recognize arbitrarily sized patches. However, we divided each grid segment by a second grid, into regular size image fragments, ignoring any partial patches sampled from the boundary. These patches are then classified into several patch categories, using color and texture features (see Section 3). See Figure 1, for a visualization of this approach.

For each segment, a frequency vector of patch classifications is calculated. Then, these patch classification vectors are concatenated to preserve some of the global location information (e.g., sky above and grass below) and fed into the final scene classifier. Various classifiers have been used to classify the patches and the entire picture, including kNN, naive Bayes, and back-propagation neural networks.

2.2 Related Work

Literature on scene classification is relatively limited. Early retrieval systems like QBIC [10], VisualSEEk [30], PicHunter [7], PicToSeek [13], and SIMPLIcity [36] use color, shape, and texture representations for picture search. Picard extended Photobook with capabilities for classifying patches into so-called 'stuff' categories (e.g., grass, sky, sand, and stone), using a set of competing classification models (society of models approach) [22, 25, 26].

In Blobworld, Belongie et al. [1, 5] segment pictures into regions with coherent texture and color of arbitrary shape ('blobs') and offer the user the possibility to search on specific blobs rather than the low level characteristics of the full picture. However, these blobs are not classified into stuff nor scene categories [1, 5]. Campbell et al. [3] also segment pictures into arbitrarily shaped regions and then use a neural network to classify the patches into stuff-like categories like building, road and vegetation.

Some papers are available on classification of the scene of the picture as a whole. Lipson et al. ([21]) recognize a limited set of scenes (mountains, mountain lakes, waterfalls, and fields) by deriving the global scene configuration of a picture and matching it to a handcrafted model template. For example, the template for a snowy mountain states that the bottom range of a picture is dark, the middle range very light and the top range has medium luminance. Ratan and Grimson [27] extend this work by learning the templates automatically. The templates are built using the dominant color-luminance combinations and their spatial relations in images of a specific scene category. They present results for fields and mountains only. Both papers only report results for retrieval tasks, not for classification.

Oliva et al. [23] defined global characteristics (or semantic axes) of a scene (e.g., vertical - horizontal, open - closed, and natural - artificial), for discriminating between, for example, city scenes and nature scenes. These characteristics are used to organize and sort pictures rather than classify them. Gorkani and Picard [16] classified city versus nature scenes. The algorithms used to extract the relevant features were

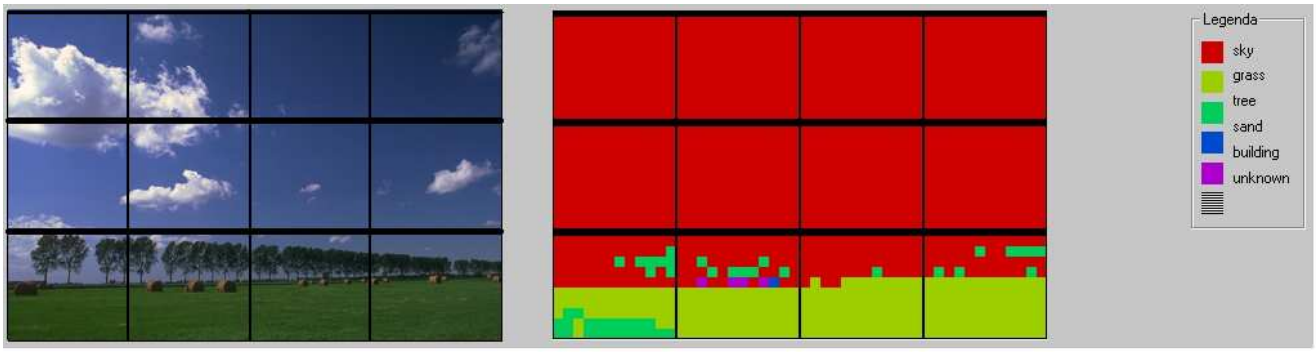

Figure 1: A screenshot of the automatic scene classifier, using a 4x3 grid. The right image shows the classified patches.

specific for these scenes (i.e., global texture orientation). In addition, Szummer and Picard [31] classified indoor and outdoor scenes. They first classified local segments as indoor or outdoor, and then classified the whole image as such. Both classifiers performed well, but it is not known whether these approaches generalize to other scene categories.

2.3 Positioning the Vicar method

Our method uses the local patch classification as input for the classification of the scene as a whole. To our knowledge only Fung et al. reported a similar approach [11, 12]. Note that the final scene classifier has only access to patch class labels. From the point of view of the final classifier, the patch classifiers are feature extractors that supply semantically rich and relevant input rather than generic syntactic color and texture information. Moreover, the patch classifiers are trained rather than being feature extractors a priori selected by an image processing expert.

So, our method differs and improves on the general applicability for a variety of scene categories, without the need to select different and task specific feature extraction algorithms, for each classification task. Moreover, we used computationally cheap algorithms, enabling real time scene classification. A more fundamental difference is that we allow end users to add knowledge of the real world to the classification and retrieval engines, which means that it should be possible to outperform any purely data driven approach, even if it is based on optimal classifiers. This is important given the fact that image processing expertise is scarce and not available to end users, but knowledge of the world is abundant.

3. PATCH FEATURES

In this section, we discuss the patch features as used for patch classification. They provide the foundation for the scene classifier. In order of appearance, we discuss: (i) color quantization using a new distributed histogram technique, (ii) color spaces, the segmentation of the HSI color space, and human color categories, and (iii) an algorithm used to determine the textural features used.

3.1 Distributed color histograms

At the core of many color matching algorithms lies a technique based on histogram matching. This is no different for the current scene classification system.

Let us, therefore, define a color histogram of size n. Then, each pixel j present in an image, has to be assigned to a bin (or bucket) b. Each pixel is assigned to a bin, as follows:

The bin b_i, with $i \in \{0, n-1\}$, for a pixel j with value x_j, is determined using:

$$\beta_i = \frac{x_j}{s}, \tag{1}$$

where x_j is the value of pixel j and s is the size of the intervals, with s determined as follows:

$$s = \frac{max(x) - min(x)}{n}, \tag{2}$$

where *max(x)* and *min(x)* are respectively the maximum and minimum value x_j can take.

For convenience, Equation 2 is substituted into Equation 1, which yields:

$$\beta_i = \frac{n \cdot x_j}{max(x) - min(x)}, \tag{3}$$

Now, b_i is defined as the integer part of the decimal number β_i.

As for each conversion from a originally analog to a digital (discrete) representation, one has to determine the precision of the discretization and with that the position of the boundaries between different elements of the discrete representation. In order to cope with this problem, we distributed each pixel over three bins, instead of assigning it to one bin.

Let us consider an image with p pixels that has to be distributed over n bins. Further, we define $min(b_i)$ and $max(b_i)$ as the borders of bin i (b_i). Then, when considering an image pixel by pixel, the update of the histogram for each of these pixels, is done as follows:

$$b_i \quad += \quad 1 \tag{4}$$

$$b_{i-1} \quad += \quad 1 - \frac{|x_j - min(b_i)|}{max(b_i) - min(b_i)} \tag{5}$$

$$b_{i+1} \quad += \quad 1 - \frac{|x_j - max(b_i)|}{max(b_i) - min(b_i)} \tag{6}$$

where $min(b_i) \leq x_j \leq max(b_i)$, with $i \in \{0, n-1\}$ and $j \in \{0, p-1\}$

Please note that this approach can be applied on all histograms, but its effect becomes stronger with the decline in the number of bins a histogram consists of.

3.2 Color

No color quantization can be done without a color representation. The RGB color space is the most used color space for computer graphics. However, the HSI / HSV (Hue, Saturation, and Intensity / Value) color spaces are more closely related to human color perception than the RGB color space [20, 34]. Therefore, we have chosen to use the HSI color space.

Here, we took into account human perceptual limitations. If Saturation was below 0.2, Intensity was below 0.12, or Intensity was above 0.94, then the Hue value has not been taken into account, given that, since for these Saturation and Intensity values the Hue is not visible as a color.

Since image and video material is defined in the RGB color space, we needed to convert this color space to the HSI color space. This was done as follows:

$$H \quad = \quad arctan\left(\frac{\frac{\sqrt{3}}{2}(G-B)}{R - \frac{1}{2}(G+B)}\right) \tag{7}$$

$$S \quad = \quad \sqrt{\left(R - \frac{\sqrt{3}}{2}(G-B)\right)^2 + \left(\frac{1}{2}(G+B)\right)^2} \tag{8}$$

$$I \quad = \quad \frac{R+G+B}{3} \tag{9}$$

Note that, all H, S, and I values were normalized to values between 0 and 1.

But how to quantize this HSI color space? From literature [2, 8, 9, 15, 19, 28, 32] is known that people use a limited set of color categories. Color categories can be defined as a fuzzy notion of some set of colors. People use these categories when thinking of or speaking about colors or when they recall colors from memory.

No exact definition of the number nor the exact content of the color categories is present. However, all research mentions a limited number of color categories: ranging between 11[2, 32, 33] and 30[8], where most evidence is found for 11 color categories. We conducted some limited experiments with subjective categories (i.e., categories indicated by humans) but these did not give better results to 16 evenly distributed categories, so for simplicity we used this categorization.

Our 16 color categories are defined by an equal division of the Hue axis of the HSI color space, since the Hue represents color. Luminance is represented by the Intensity axis of the HSI color space. Again we have chosen for a coarse quantization: the Intensity-axis is divided into six equal segments. The Saturation-axis was not segmented.

The original RGB color coordinates were converted to Hue and Intensity coordinates by Equations 7 and 9. Next, for both the Hue and the Intensity histogram, using Equation 3 each pixel is assigned to a bin. Last, Equations 4, 5, and 6 are applied on both histograms to update them. Note that, due to the circular character of the Hue, the last bin and the first bin of our Hue histogram are neighbors. Our algorithm takes into account this implication of Hue's circularity. Since both histograms are a coarse quantization this method (i) is computationally cheap (making real time classification possible) and (ii) facilitates in generalization by classifiers.

3.3 Texture

Next to color, texture can be analyzed. Jain and Karu [18] state: "Texture [eludes] a formal definition". Let us define texture as follows: A repetitive arrangement of pixels values that either is perceived or can be described as such.

For texture analysis, in most cases the Intensity of the pixels is used, hereby ignoring their color [24, 34]. Several techniques are used to determine the patterns that may be perceived from the image [29]. With most texture analyses, textural features are derived from the image, instead of describing arrangements of the individual pixels. This reduces the computational costs significantly, which is essential for applications working real time.

Therefore, we used a texture algorithm that extracts three textual features for each position of a mask that is run over the image. Here, the size of the mask determines the ratio between local and global texture analysis. The position of the mask is defined by its central pixel. Note that the mask is a square of $n \times n$ pixels, with n being an odd integer.

For each pixel of the mask, the difference between both its horizontal neighbors as well as the difference between its vertical neighbors is determined. (p, q) denotes the elements (i.e., pixels) of the image with (i, j) being the coordinates of the pixels located in a mask, surrounding an image pixel (p, q). Function f determines the normalized value of pixel (i, j) for a chosen color channel (i.e., H, S, or I), using Equations 7, 8, and 9.

$$\begin{aligned}
&foreach(p,q) \in Image \\
&\quad foreach(i,j) \in Mask(p,q) \\
&\quad\quad Sum+ = f(i,j) \\
&\quad\quad SqSum+ = f(i,j)^2 \\
&\quad\quad M_{11}+ = (f(i+1,j) - f(i-1,j))^2 \\
&\quad\quad M_{12}+ = (f(i,j+1) - f(i,j-1))^2 \\
&\quad\quad M_{22}+ = f(i+1,j) - f(i-1,j)) * \\
&\quad\quad\quad\quad\quad\quad (f(i,j+1) - f(i,j-1))
\end{aligned}$$

So, for each mask M_{11}, M_{12}, and M_{22} are determined, defining the symmetric covariance matrix M. Let ev_1 and ev_2 be the eigenvalues of M (for more details, see for example Jähne [17] on structure tensor).

Given this algorithm, three textural features can be determined:

$$F_1 = SqSum - Sum^2 \quad (10)$$

$$F_2 = \frac{min\{ev_1, ev_2\}}{max\{ev_1, ev_2\}} \quad (11)$$

$$F_3 = max\{ev_1, ev_2\} \quad (12)$$

F_1 (see Equation 10) can be identified as the variance (σ^2), indicating the global amount of texture present in the image. The other two features, F_2 and F_3 (see Equations 11 and 12), indicate the structure of the texture available. If ev_1 and ev_2 differ significantly, stretched structures are present (e.g., lines). When ev_1 and ev_2 have a similar value (i.e., F_2 approximates 1; see Equation 11), texture is isotropic. In the case both ev_1 and ev_2 are large (i.e., both F_2 and F_3 are large; see Equation 11 and 12), clear structure is present, without a clear direction. In the case ev_1 and ev_2 are both small (i.e., F_2 is large and F_3 is small; see Equation 11 and 12), smooth texture is present. Moreover, F_2 and F_3 are rotation-invariant.

Hence, this triplet of textural features provides a good indication for the textural properties of images, both locally and globally. In addition, it is computationally cheap and, therefore, very useful for real time content-based video retrieval.

4. EXPERIMENTS AND RESULTS

In the previous section (Section 3) the features used were introduced. These features were used for the first phase of classification: the classification of patches, resulting in a frequency vector of patch classes for each grid cell.

In the second phase of classification, a classifier is used to classify the whole image. The input for the classifier is the concatenation of all frequency vectors of patch classes for each grid cell.

So, two phases exist, each using their own classifier. We have experimented with two types of classifiers: A K-nearest neighbors classifier (kNN) and a neural network. We will now discuss both the patch classification (Section 4.1) and the scene classification (Section 4.2).

The advantage of kNN is that it is a lazy method, i.e. the models need no retraining. This is an important advantage given that we envisage an interactively learning application. However, given that kNN does not abstract a model from the data, it suffers more from the curse of dimensionality and will need more data to provide accurate and robust results. The neural network needs training, parameter optimization and performance tuning, however it can provide good results on smaller data sets providing that the degrees of freedom in the model are properly controlled.

The experiments discussed in the next two subsections all used the Corel image database as test bed.

4.1 Patch classification

In this section we will discuss the patch classification. In the next section, the classification of the image as a whole is discussed.

Each of the patches had to be classified to one of the nine patch categories defined (i.e., building, crowd, grass, road, sand, skin, sky, tree, and water). First, a kNN classifier was used for classification. This is because it is a generic classification method. In addition, it could indicate whether a more complex classification method would be needed. However, the classification performance was poor. Therefore, we have chosen to use a neural network for the classification of the grid cells, with nine output nodes (as much as there were patch classes).

On behalf of the neural network, for each of the nine patch classes both a train and a test set were randomly defined, with a size ranging from 950 to 2500 patches per category. The neural network architecture was as follows: 25 input, 30 hidden, and 9 output nodes. The network ran 5000 training cycles with a learning rate of 0.007.

With a patch size of 16x16, the patch classifier had an overall precision of 87.5%. The patch class crowd was confused with the patch class building in 5.19% of the cases. Sand and skin were also confused. Sand was classified as skin in 8.80% of the cases and skin was classified as sand in 7.16% of the cases. However, with a precision of 76.13% the patch class road appeared the hardest to classify. In the remaining 23.87% of the cases road was confused with one of the other eight patch classes, with percentages ranging from 1.55% to 5.81%. The complete results can be found in Table 1.

Table 2 shows the results for a 8x8 patch classifier in one of our experiments. The 16x16 patch classifier clearly outperforms the 8x8 patch classifier with an overall precision of 87.5% versus 74.1%. So, the overall precision for the 8x8 patch classifier decreases with 13.4% compared to the precision of the 16x16 classifier. The decline in precision for each category, is as follows: sand 22.16%, water 21.26%, building 17.81%, skin 17.48%, crowd 17.44%, tree 16.8% and road 7.16%. Only for the categories grass and sky the classification was similar for both patch sizes.

Note that Figure 1 presents a screenshot of the system, illustrating both the division of an image into grids. The classified patches are resembled by little squares in different colors.

So far, we have only discussed patch classification in general. However, it was applied on each grid cell separately: For each grid cell, each patch was classified to a patch category. Next, the frequency of occurrence of each patch class, for each grid cell, was determined. Hence, each grid cell could be represented as a frequency vector of the nine patch classes. This served as input for the next phase of processing: scene classification, as is discussed in the next subsection.

Table 1: Confusion matrix of the patch (size: 16x16) classification for the test set. The x-axis shows the actual category, the y-axis shows the predicted category.

	building	crowd	grass	road	sand	skin	sky	tree	water	unknown
building	89.23	3.02	0.09	1.11	1.02	0.60	0.38	3.70	0.85	0.00
crowd	5.19	87.25	0.19	1.81	0.44	0.50	0.38	2.94	0.06	1.25
grass	0.00	0.00	94.73	0.73	0.60	0.00	0.00	3.00	0.93	0.00
road	1.55	5.48	2.84	76.13	1.55	1.74	1.81	5.81	3.10	0.00
sand	1.84	0.88	2.24	1.44	83.68	8.80	0.24	0.00	0.64	0.24
skin	0.32	2.53	0.00	0.63	7.16	89.37	0.00	0.00	0.00	0.00
sky	0.21	0.00	0.00	2.57	0.93	0.00	91.71	0.36	3.86	0.36
tree	1.12	3.44	2.60	0.32	0.16	0.24	0.56	88.44	0.84	2.28
water	0.00	0.00	4.00	4.44	0.52	0.00	3.04	0.44	87.26	0.30

Table 2: Confusion matrix of the patch (size: 8x8) classification for the test set. The x-axis shows the actual category, the y-axis shows the predicted category.

	building	crowd	grass	road	sand	skin	sky	tree	water	unknown
building	71.42	9.00	0.85	2.69	2.43	2.86	0.26	6.53	0.77	3.20
crowd	10.38	69.81	1.13	1.56	2.13	5.56	0.69	6.44	0.19	2.13
grass	0.80	0.07	93.87	0.73	0.07	0.73	1.20	1.20	0.87	0.47
road	2.65	5.81	2.45	68.97	2.97	1.87	5.48	3.10	4.52	2.19
sand	3.44	3.12	2.88	1.84	61.52	15.20	8.80	0.16	2.80	0.24
skin	1.16	7.79	0.42	0.11	13.47	71.89	4.42	0.11	0.11	0.53
sky	0.00	0.00	0.00	0.29	1.36	2.57	91.43	0.07	4.07	0.21
tree	4.56	11.08	8.20	1.88	0.52	0.76	0.24	71.64	0.56	0.56
water	0.37	0.52	3.26	9.78	3.85	3.85	11.41	0.52	66.00	0.44

Table 3: Confusion matrix of the scene classification for the test set. The x-axis shows the actual category, the y-axis shows the predicted category.

	Interior	City/street	Forest	Country	Desert	Sea	Portraits	Crowds
Interior	82.0	8.0	2.0	0.0	0.0	0.0	2.0	6.0
City/street	10.0	70.0	4.0	8.0	0.0	0.0	2.0	6.0
Forest	2.0	4.0	80.0	2.0	2.0	8.0	0.0	2.0
Country	0.0	6.0	28.0	54.0	10.0	0.0	0.0	2.0
Desert	8.0	6.0	2.0	10.0	64.0	4.0	4.0	2.0
Sea	4.0	14.0	0.0	2.0	0.0	80.0	0.0	0.0
Portraits	8.0	0.0	0.0	4.0	4.0	2.0	80.0	2.0
Crowds	4.0	14.0	0.0	0.0	2.0	0.0	0.0	80.0

4.2 Scene classification

The system had to be able to distinguish between eight categories of scenes, relevant for the Vicar project: interiors, city/street, forest, agriculture/countryside, desert, sea, portrait, and crowds. In pilot experiments several grid sizes were tested: a 3x2 grid gave the best results. The input of the classifiers were the normalized and concatenated grid vectors. The elements of each of these vectors represented the frequency of occurrence of each of the reference patches, as they were determined in the patch classification (see Section 4.1).

Again, first a kNN classifier was used for classification. Similarly to the patch classification, the kNN had a low precision. Therefore, we have chosen to use a neural network for the classification of the complete images, with eight output nodes (as much as there were scene classes).

For each of the eight scene classes both a train and a test set were randomly defined. The train sets consisted of 199, 198, or 197 images. For all scene classes, the test sets consisted of 50 images. The neural network architecture was as follows: 63 input, 50 hidden, and 8 output nodes. The network ran 2000 training cycles with a learning rate of 0.01.

The image classifier was able to classify 73,8% of the images correct. Interior (82% precision) was confused with city/street in 8.0% and with crowds in 6.0% of the cases. City/street was correctly classified in 70.0% of the cases and confused with interior (10%), with country (8.0%), and with crowds (6.0%). Forest (80% precision) was confused with sea (8.0%). Country was very often (28.0%) confused with forest and was sometimes confused with city/street (6.0%) and with desert (10%), which resulted in a low precision: 54.0%. In addition, also desert had a low precision of classification (64%); it was confused with: interior (8.0%), city/street (6.0%), and with country (10%). Sea, portraits, and crowds had a classification precision of 80.0%. Sea was confused with city/street in 14%, portraits were confused with interior in 8.0% of the cases, and crowds were confused with city/street in 14.0% of the cases. In Table 3 the complete results for each category separately are presented.

5. DISCUSSION

Let us discuss the results of patch and scene classification separate, before providing overall issues. For patch classification, two patch sizes have been applied. The 16x16 patch classifier gave clearly a much higher precision than the 8x8 patch classifier. Our explanation is that a 16x16 patch can contain more information of a (visual) category than a 8x8 patch. Therefore, some textures can not be described in a 8x8 patch (e.g., patches of buildings). A category such as grass, on the other hand, performed well with 8x8 patches. This is due to its high frequency of horizontal lines that fit in a 8x8 patch.

Therefore, the final system tests were done with the 16x16 patch size, resulting in an average result of 87,5% correct. Campbell and Picard [4, 25, 26] reported similar results. However, our method has major advantages in terms of a much lower computational complexity. Moreover, the classified patches themselves are intermediate image representations and can be used for image classification, image segmentation as well as for image matching.

Hitherto, the patches with which the classifiers were trained had to be manually classified. So, the development of a general purpose automatic scene classifying system would ask an enormous effort: In principle, for all possible patches, sets of reference patches should be manually classified. To solve the latter problem, we currently develop algorithms for automatic extraction of relevant patch types, to utilize automatic training of our system.

The second phase of the system consists of the classification of the image representation, using the concatenated frequency patch vectors of the grid cells. An average performance of 73.8% was achieved. The least performing class is Country (which includes the categories countryside and agriculture) with 54% correct. What strikes immediately, when looking at the detailed results in Table 2, is that this category is confused in 28% of the times with the category forest and in 10% of the times with the category desert.

The latter confusions can be explained by the strong visual resemblance between the three categories, which is reflected in the corresponding image representations from these different categories. To solve such confusions, the number of patch categories could be increased. This would increase the discriminating power of the representations. Note that if a user searches on the index rather than on the class label, the search engine may very well be able to search on images that are a mix of multiple patches and scenes.

To make the system truly interactive, classifiers are needed that offer the flexibility of kNN (no or very simple training) but the accuracy of more complex techniques. We have experimented with learning algorithms such as naive Bayes, but the results have not been promising yet. Furthermore, one could exploit the interactivity of the system more, for instance by adding any misclassifications identified by the user to the training data. Finally, the semantic indices are not only useful for search or classification but may very well be used as input for other mining tasks. An example would be to use index clustering to support navigation through clusters of similar video material.

6. APPLICATIONS

The scene classifier has been embedded into several applications. In this section we will describe two of them.

6.1 Vicar

The scene classifier has been integrated into the Vicar Video Navigator [35]. This system utilizes text-based search, either through manual annotations or through automatically generated classifications like the global scene labels. As a result, Vicar returns the best matching key frames along with information about the associated video. In addition, a user can refine the search by combining a query by image with text-based search. The query by image can either be carried out on local characteristics (appearance) or may include content based query by image. In the first case, the index consisting of the concatenated patch classification vectors is included in the search. In the latter case, the resulting index of scores on the global scene classifiers is used (content). In Figures 2 and 3 an example search is shown from a custom

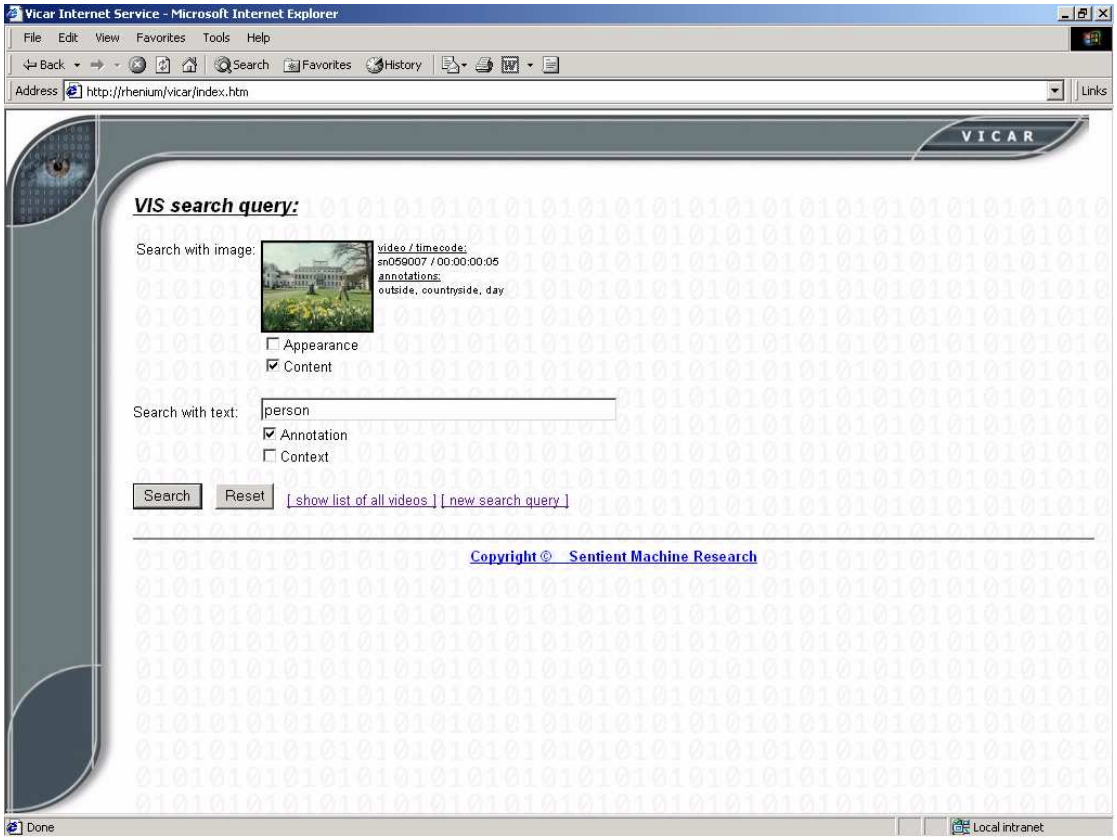

Figure 2: A query for video material.

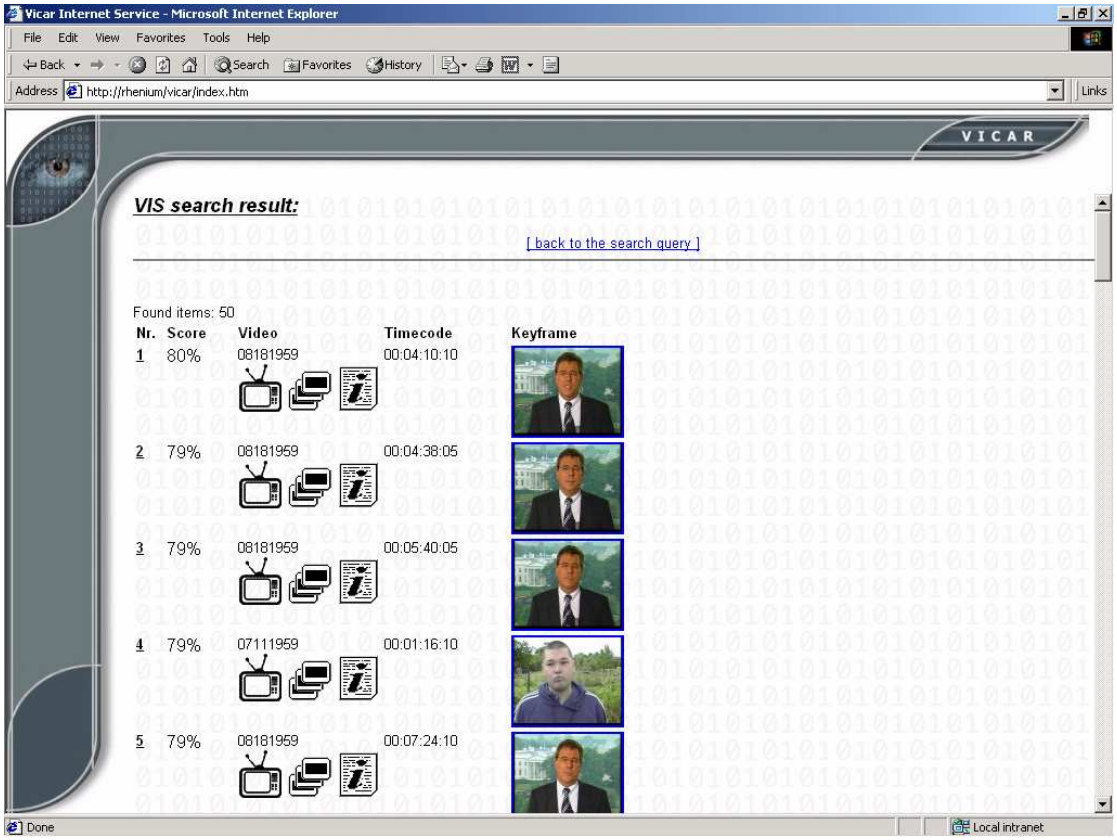

Figure 3: The result of a query for video material.

made web application based on the Vicar technology: the first screenshot shows one of the key frames that has been retrieved from the archive using the (automated annotated) keyword countryside. An extra keyword person (also automated annotated) is added in the search, as well as the content index of the image. In the second screenshot the results of the combined queries are shown: persons with a similar background scene as the query image.

6.2 Porn filtering

To test the general applicability of our approach we built a new classifier to distinguish pornographic from non pornographic pictures. Within half a day a classifier was constructed with a precision of over 80%. As a follow up, a project for porn filtering was started within the EU Safer Internet Action Plan (IAP) program. Within this project, SCOFI, a real time classification system was built, which is currently running on several schools in Greece, England, Germany and Iceland. The porn image classifier is combined with a text classifier and integrated with a proxy server to enable safe web surfing. The text classifier and the proxy server have been developed by Demokritos, Greece, and are part of the Filterix system [6].

For this application of the system, we first created image representations using the patch classification network as mentioned in Section 4.1. With these image representations we trained the second phase classifier, using 8.000 positive (pornographic) and 8.000 negative (non pornographic) examples. The results: the system was able to detect 92% of the pornographic images in a diverse image collection of 2.000 positive examples and 2.000 negative examples (which includes non pornographic pictures of people). There were 8% false positives (images that are not pornographic, are identified as pornographic images) and 8% false negatives. Examples of false positives were close ups of faces and pictures like deserts and fires. To improve results, within the SCOFI project a Vicar module was used that detects close ups of faces.

The integrated SCOFI system that combines text and image classification has a performance of 0% overblocking (i.e., 100% correct on non pornographic web pages) and 1% underblocking (i.e., 99% correct on pornographic web pages). As such it is used as a real time filter for filtering pornography on the Internet, in several schools throughout Europe.

7. CONCLUSION

In this paper a general scene classifier is introduced that does not rely on computationally expensive object recognition. The features that provide the input for the final scene classification are generated by a set of patch classifiers that are learned rather than predefined, and specific for the scenes to be recognized rather than general. Though the results on different scene categories can still be improved, the current system can successfully be used as a tool for generating scene indexes and classifications for content-based image and video retrieval and filtering. This is demonstrated by its success in various applications such as the Vicar Video Navigator video search engine and the SCOFI real time filter for pornographic image material on the Internet.

8. ACKNOWLEDGMENTS

This work was partially supported by the EU projects VICAR (IST-24916) and SCOFI (IAP-2110; http://www.scofi.net/). Further, we gratefully acknowledge the reviewers, for their comments on the manuscript. We thank Robert Maas for his work on the texture algorithm.

9. ADDITIONAL AUTHORS

Additional authors: Marten J. den Uyl, Vicar Vision b.v., The Netherlands, email: m.denuyl@vicarvision.nl

10. REFERENCES

[1] S. Belongie, C. Carson, H. Greenspan, and J. Malik. Recognition of images in large databases using a learning framework. Technical Report CSD-97-939, University of California at Berkeley, 1997.

[2] B. Berlin and P. Kay. *Basic color terms: Their universals and evolution*. Berkeley: University of California Press, 1969.

[3] N. W. Campbell, W. P. J. Mackeown, B. T. Thomas, and T. Troscianko. The automatic classification of outdoor images. In *Proceedings of the International Conference on Engineering Applications of Neural Networks*, pages 339–342. Systems Engineering Association, 1996.

[4] N. W. Campbell, W. P. J. Mackeown, B. T. Thomas, and T. Troscianko. Interpreting image databases by region classification. *Pattern Recognition*, 30(4):555–563, 1997.

[5] C. Carson, S. Belongie, H. Greenspan, and J. Malik. Blobworld: Image segmentation using expectation-maximization and its application to image querying. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(8):1026–1038, 2002.

[6] K. V. Chandrinos, I. Androutsopoulos, G. Paliouras, and C. D. Spyropoulos. Automatic web rating: Filtering obscene content on the web. In J. Borbinha and T. Baker, editors, *Proceedings of the 4th European Conference on Research and Advanced Technology for Digital Libraries*, pages 403–406, 2000.

[7] I. J. Cox, M. L. Miller, T. P. Minka, and T. V. Papathomas. The bayesian image retrieval system, pichunter: theory, implementation, and psychophysical experiments. *IEEE Transactions on Image Processing*, 9(1):20–37, 2000.

[8] G. Derefeldt and T. Swartling. Colour concept retrieval by free colour naming: Identification of up to 30 colours without training. *Displays*, 16(2):69–77, 1995.

[9] G. Derefeldt, T. Swartling, U. Berggrund, and P. Bodrogi. Cognitive color. *Color Research & Application*, 29(1):7–19, 2004.

[10] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee,

D. P. nad D. Steele, and P. Yanker. Query by Image and Video Content: The QBIC system. *IEEE Computer*, 28(9):23–32, 1995.

[11] C. Y. Fung and K.-F. Loe. Learning primitive and scene semantics of images for classification and retrieval. In *Proceedings of the 7th ACM International Conference on Multimedia '99*, volume 2, pages 9–12, rlando, Florida, USA, 1999. ACM.

[12] C. Y. Fung and K.-F. Loe. A new approach for image classification and retrieval. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 301–302. ACM, 1999.

[13] T. Gevers and A. W. M. Smeulders. Pictoseek: combining color and shape invariant features for image retrieval. *IEEE Transactions on Image Processing*, 9(1):102–119, 2000.

[14] J. Gibson. *The Ecological Approach to Visual Perception*. Houghton Mifflin, Boston, 1979.

[15] R. Goldstone. Effects of categorization on color perception. *Psychological Science*, 5(6):298–304, 1995.

[16] M. M. Gorkani and R. W. Picard. Texture orientation for sorting photos at a glance. In *Proceedings of the International Conference on Pattern Recognition*, volume 1, pages 459–464, 1994.

[17] B. Jähne. *Practical Handbook on Image Processing for Scientific Applications*. CRC Press, 1997.

[18] A. K. Jain and K. Karu. Learning texture discrimination masks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(2):195–205, 1996.

[19] P. Kay. Color. *Journal of Linguistic Anthropology*, 1:29–32, 1999.

[20] T. Lin and H. Zhang. Automatic video scene extraction by shot grouping. In *Proceedings of the 15th International Conference on Pattern Recognition*, volume 4, pages 39–42, Barcelona, Spain, 2000. IEEE Computer Society.

[21] P. Lipson, E. Grimson, and P. Sinha. Configuration based scene classification and image indexing. In *Proceedings of 16th IEEE Conference on Computer Vision and Pattern Recognition*, pages 1007–1013. IEEE Computer Society, 1997.

[22] T. P. Minka and R. W. Picard. Interactive learning using a "society of models". Technical Report 349, MIT Media Laboratory Perceptual COmputing Section, 1996.

[23] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42(3):145–175, 2001.

[24] C. Palm. Color texture classification by integrative co-occurrence matrices. *Pattern Recognition*, 37(5):965–976, 2004.

[25] R. W. Picard. Light-years from lena: video and image libraries of the future. In *Proceedings of the 1995 International Conference on Image Processing*, volume 1, pages 310–313, 1995.

[26] R. W. Picard and T. P. Minka. Vision texture for annotation. *Multimedia Systems*, 3(1):3–14, 1995.

[27] A. L. Ratan and W. E. L. Grimson. Training templates for scene classication using a few examples. In *Proceedings of the IEEE Workshop on Content-Based Analysis of Images and Video Libraries*, pages 90–97, 1997.

[28] D. Roberson, I. Davies, and J. Davidoff. *Colour categories are not universal: Replications and new evidence from a stone-age culture*. Lanham, Maryland: University Press of America Inc., 2002.

[29] A. Rosenfeld. From image analysis to computer vision: An annotated bibliography, 1955-1979. *Computer Vision and Image Understanding*, 84(2):298–324, 2001.

[30] J. R. Smith and S. F. Chang. *Querying by color regions using the VisualSEEk content-based visual query system*, chapter 2, pages 23–42. The AAAI Press, 1997.

[31] M. Szummer and R. W. Picard. Indoor-outdoor image classification. In *IEEE International Workshop on Content-Based Access of Image and Video Databases (CAIVD)*, pages 42–51, Bombay, India, 1998. IEEE Computer Society.

[32] E. L. van den Broek, M. A. Hendriks, M. J. H. Puts, and L. G. Vuurpijl. Modeling human color categorization: Color discrimination and color memory. In T. Heskes, P. Lucas, L. Vuurpijl, and W. Wiegerinck, editors, *Proceedings of the 15th Belgian-Netherlands Conference on Artificial Intelligence*, pages 59–68. Nijmegen: SNN, University of Nijmegen, 2003.

[33] E. L. van den Broek, P. M. F. Kisters, and L. G. Vuurpijl. The utilization of human color categorization for content-based image retrieval. In B. E. Rogowitz and T. N. Pappas, editors, *Proceedings of Human Vision and Electronic Imaging IX*, volume 5292, pages 351–362, 2004.

[34] E. L. van den Broek and E. M. van Rikxoort. Colorful texture analysis. *Pattern Recognition Letters*, [submitted].

[35] P. van der Putten. Vicar video navigator: Content based video search engines become a reality. *Broadcast Hardware International, IBC edition*, 1999.

[36] J. Z. Wang. *Integrated region-based image retrieval*. Boston: Kluwer Academic Publishers, 2001.