10. Visual Alphabets: Video Classification by End Users

Menno Israël, Egon L. van den Broek, Peter van der Putten, Marten J. den Uyl

Summary. The work presented here 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. The end user or domain expert thus builds a visual alphabet that can be used to describe the image in features that are relevant for the task at hand. We present results for experiments on a variety of patch and image classes. The scene classifier approach has been successfully applied to other domains of video content analysis, such as content based video retrieval in television archives, automated sewer inspection, and porn filtering.

10.1 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 [1, 2].

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 chapter 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. In essence, the end user builds a visual alphabet that can be used to describe the world in terms that matter to the task at hand.

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 [3]. 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 chapter is expository and meant to give a non-technical introduction into our methodology. A high level overview of our approach is given in Section 10.2. Section 10.3 provides more detail on the low level color and texture features used and Section 10.4 specifies the classifying algorithms used. Experimental results for patch and scene classification are given in Sections 10.4.1 and 10.4.2. Next, we highlight three applications in which scene classification technology has been embedded (Section 10.6). We finish with a discussion and conclusion (Sections 10.5 and 10.7).

10.2 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.

10.2.1 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 visual 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 10.3). See Figure 10.1, 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.

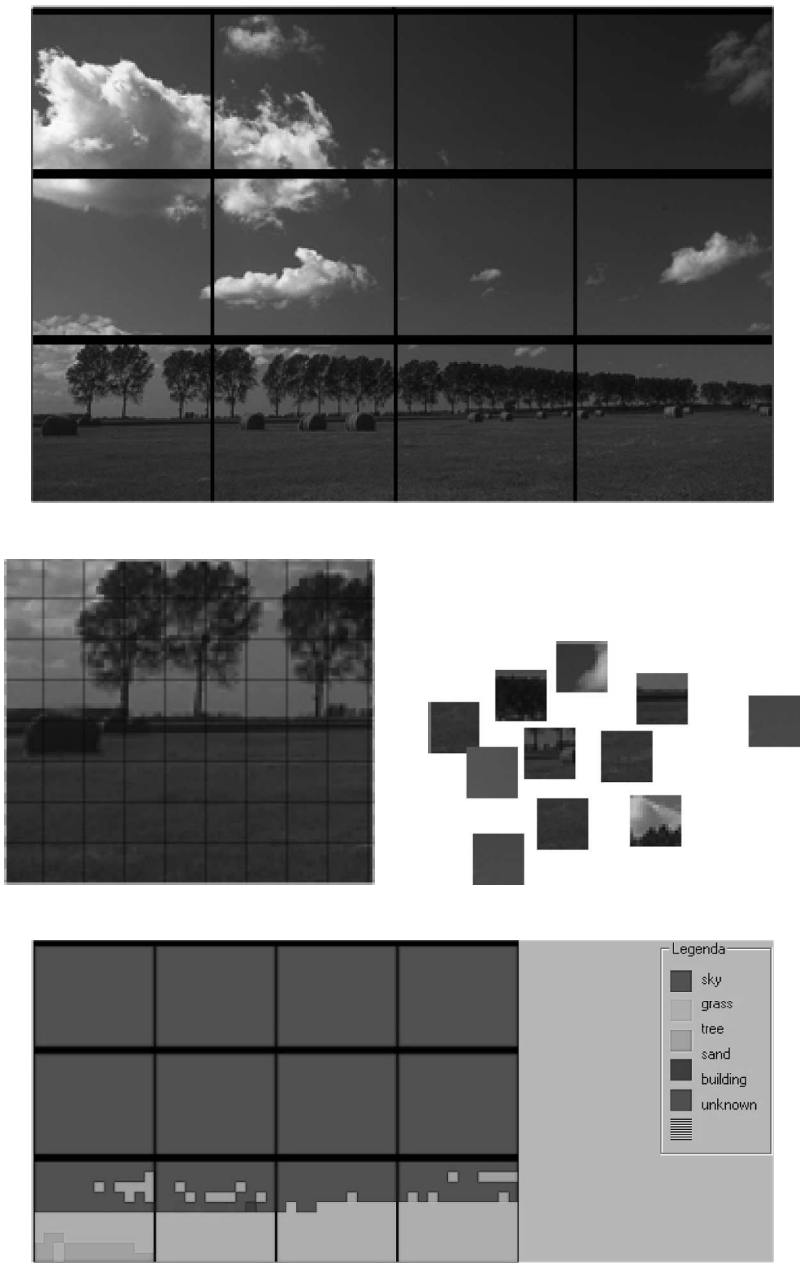

Fig. 10.1. Screenshots visualizing the first phase of the scene classification process. From top to bottom and from left to right: The images with a 4×3 grid over it, extraction of the patches from a grid cell, classification of the patches, and the resulting "patch image" with its legend.

10.2.2 Related Work

Literature on scene classification is relatively limited. Early retrieval systems like QBIC [4, 5], VisualSEEk [6], PicHunter [7], PicToSeek [8], and SIMPLIcity [9] as well as recent systems such as MARVEL [10], M4ART [11], and the system proposed by Wu, Rahman, and Chow [12], use color, shape, and texture representations for picture search. Minka and Picard [13], Picard [14], and Picard and Minka [15] 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).

In Blobworld, Belongie, Carson, Greenspan, and Malik [16, 17] 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 [16, 17]. Campbell, Mackeown, Thomas, and Troscianko [18, 19] 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, Grimson, and Sinha [20] 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 [21] 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 and Torralba [22] 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 [23] classified city versus nature scenes. The algorithms used to extract the relevant features were specific for these scenes (i.e., global texture orientation). In addition, Szummer and Picard [24] 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.

10.2.3 Positioning the Visual Alphabet Method

Our method uses the local patch classification as input for the classification of the scene as a whole. To our knowledge only Fung and Loe [25, 26] reported a

similar approach. 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.

10.3 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, and histogram configurations (ii) human color categories, color spaces, and the segmentation of the HSI color space, and (iii) an algorithm used to determine the textural features used.

10.3.1 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:

$$\beta_i = \frac{x_j}{s}, \tag{10.1}$$

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

$$s = \frac{\max(x) - \min(x)}{n}, \tag{10.2}$$

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

For convenience, Equation 10.2 is substituted into Equation 10.1, which yields:

$$\beta_i = \frac{n \cdot x_j}{\max(x) - \min(x)}, \tag{10.3}$$

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:

$$b_i \quad += 1 \tag{10.4}$$

$$b_{i-1} += 1 - \frac{|x_j - \min(b_i)|}{\max(b_i) - \min(b_i)} \tag{10.5}$$

$$b_{i+1} += 1 - \frac{|x_j - \max(b_i)|}{\max(b_i) - \min(b_i)} \tag{10.6}$$

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.

10.3.2 Histogram Configurations

Several histogram configurations have been presented over the years [27]. For example, the PicHunter [7] image retrieval engine uses a HSV($4 \times 4 \times 4$) (i.e., 4 Hues, 4 Saturations, and 4 Values) quantization method. In [28] a HSV($18 \times 3 \times 3$) bin quantization scheme is described. The QBIC configuration used 4096 bins [4, 5]: RGB($16 \times 16 \times 16$). For more detailed discussions concerning color quantization we refer to [27, 29, 30, 31, 32, 33].

Histogram matching on a large number of bins, has a big advantage: Regardless of the color space used during the quantization process, the histogram matching will have a high precision. Disadvantages of our approach are its high computational complexity and poor generalization.

When a coarse color quantization is performed, these disadvantages can be solved. So, since the system should work real-time and the classifiers have to be able to generalize over images, a coarse color quantization, is needed.

However, to ensure an acceptable precision, it is of decisive importance that human color perception is respected during quantization. Hence, the combination of color space and the histogram configuration is crucial for the acceptance of the results by the user.

10.3.3 Human Color Categories

As mentioned by Forsyth and Ponse [34]: "It is surprisingly difficult to predict what colors a human will see in a complex scene; this is one of the many difficulties that make it hard to produce really good color reproduction systems."

From literature [35, 30, 36, 37, 38, 39, 40, 41] 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. Research from various fields of science emphasizes the importance of focal colors in human color perception. The use of this knowledge may provide the means for bridging the semantic gap that exists in image and video classification.

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 [35, 29, 30] and 30 [37], where most evidence is found for 11 color categories. We conducted some limited experiments with subjective categories (categories indicated by humans) but these did not give better results to 16 evenly distributed categories, so for simplicity we used this categorization. Now that we have defined a coarse 16 bin color histogram to define color with, we need a color space on which it can be applied.

10.3.4 Color Spaces

No color quantization can be done without a color representation. Color is mostly represented as tuples of (typically three) numbers, conform certain specifications (that we name a color space). One can describe color spaces using two important notions: perceptual uniformity and device dependency. Perceptually uniform means that two colors that are equally distant in the color space are perceptually equally distant. A color space is device dependent when the actual color displayed depends on the device used.

The RGB color space is the most used color space for computer graphics. It is device dependent and not perceptually uniform. The conversion from a RGB image to a gray value image simply takes the sum of the R,G, and B values and divides the result by three.

The HSI / HSV (Hue, Saturation, and Intensity / Value) color spaces are more closely related to human color perception than the RGB color space, but are still not perceptual uniform. In addition, they are device-dependent.

Hue is the color component of the HSI color space. When Saturation is set to 0, Hue is undefined and the Intensity / Value-axis represents the gray-scale image.

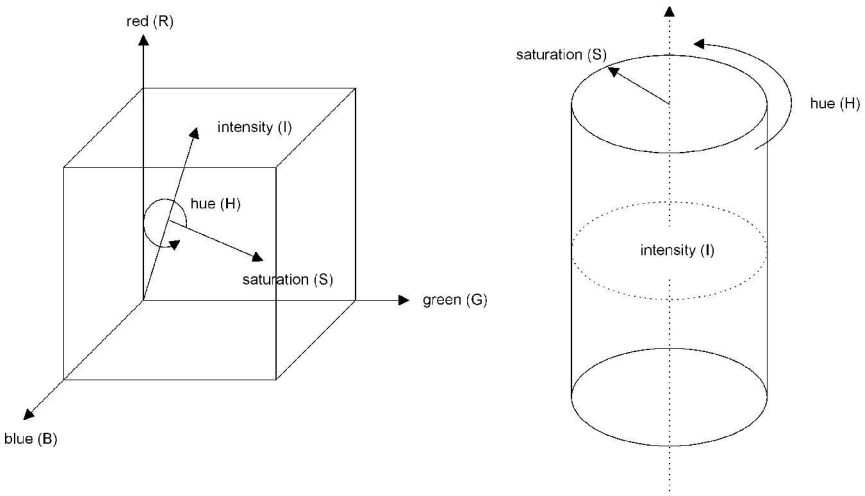

Fig. 10.2. Left: The relation between the RGB and the HSI color space, from the perspective of the RGB color space. Right: The cylinder shaped representation of the HSI (hue, saturation, and intensity) color space, as used in this research. The figure is adopted from Van den Broek [30].

Despite the fact that the HSI and HSV color spaces are not perceptually uniform, they are found to perform as good of better than perceptual uniform spaces such as CIE LUV [42]. Therefore, we have chosen to use the HSI color space.

Hereby, 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. This, 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 [43]:

$$H = \arctan\left(\frac{\frac{\sqrt{3}}{2}(G-B)}{R-\frac{1}{2}(G+B)}\right) \tag{10.7}$$

$$S = \sqrt{\left(R-\frac{\sqrt{3}}{2}(G-B)\right)^2+\left(\frac{1}{2}(G+B)\right)^2} \tag{10.8}$$

$$I = \frac{R+G+B}{3} \tag{10.9}$$

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

10.3.5 Segmentation of the HSI Color Space

Our 16 color categories are defined by an equal division of the Hue axis of the HSI color space, since the Hue represents color. So far, only color was defined and luminance is ignored.

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 original RGB color coordinates were converted to Hue and Intensity coordinates by Equations 10.7 and 10.9, as adopted from Gevers and Smeulders [43]. Next, for both the Hue and the Intensity histogram, using Equation 10.3 each pixel is assigned to a bin. Last, Equations 10.4, 10.5, 10.6 are applied on both histograms to update them. Since both histograms were a coarse quantization this method (i) is computationally cheap (making real-time classification possible) and (ii) facilitates in generalization by classifiers.

10.3.6 Texture

Next to color, texture can be analyzed. Jain and Karu [44] 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 [36, 45, 46, 47]. Several techniques are used to determine the patterns that may be perceived from the image [27, 30, 36, 46, 47, 48]. With most texture analyzes, 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 10.7, 10.8, and 10.9.

Using the algorithm below, 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 [49] on structure tensor. Recent work on nonlinear structure tensors has been presented by Brox, Weickert, Burgeth, and Mrázek [50].

$$
\begin{array}{l}
\text{foreach}(p,q) \in \text{Image} \\
\quad \text{foreach}(i,j) \in \text{Mask}(p,q) \\
\qquad \text{Sum} \mathrel{+}= f(i,j) \\
\qquad \text{SqSum} \mathrel{+}= f(i,j)^2 \\
\qquad M_{11} \mathrel{+}= (f(i+1,j) - f(i-1,j))^2 \\
\qquad M_{12} \mathrel{+}= (f(i,j+1) - f(i,j-1))^2 \\
\qquad M_{22} \mathrel{+}= (f(i+1,j) - f(i-1,j)) \cdot (f(i,j+1) - f(i,j-1))
\end{array}
$$

Given this algorithm, three textural features can be determined:

$$F_1 = \text{SqSum} - \text{Sum}^2 \tag{10.10}$$

$$F_2 = \frac{\min\{ev_1, ev_2\}}{\max\{ev_1, ev_2\}} \tag{10.11}$$

$$F_3 = \max\{ev_1, ev_2\} \tag{10.12}$$

F_1 (see Equation 10.10) 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 10.11 and 10.12), 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 10.11), 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 10.11 and 10.12), 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 10.11 and 10.12), 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.

10.4 Experiments and Results

In the previous section (Section 10.3) 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 10.4.1) and the scene classification (Section 10.4.2).

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.

10.4.1 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 2,500 patches per category. The neural network architecture was as follows: 25 input, 30 hidden, and 9 output nodes. The network ran 5,000 training cycles with a learning rate of 0.007.

With a patch size of 16×16, 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 10.1.

Table 10.2 shows the results for a 8×8 patch classifier in one of our experiments. The 16×16 patch classifier clearly outperforms the 8×8 patch classifier with an overall precision of 87.5% versus 74.1%. So, the overall precision for the 8×8 patch classifier decreases with 13.4% compared to the precision of the 16×16 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 10.1 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.

10.4.2 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 3×2 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 10.4.1).

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 2,000 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 either city/street (6.0%) or 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 10.3 the complete results for each category separately are presented.

Table 10.1. Confusion matrix of the patch (size: 16×16) 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 10.2. Confusion matrix of the patch (size: 8×8) 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 10.3. 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**

10.5 Discussion and Future Work

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 16×16 patch classifier gave clearly a much higher precision than the 8×8 patch classifier. Our explanation is that a 16×16 patch can contain more information of a (visual) category than a 8×8 patch. Therefore, some textures can not be described in a 8×8 patch (e.g., patches of buildings). A category such as grass, on the other hand, performed well with 8×8 patches. This is due to its high frequency of horizontal lines that fit in a 8×8 patch.

Therefore, the final tests were carried out with the 16×16 patch size, resulting in an average result of 87,5% correct. Campbell and Picard [19, 14, 15] reported similar results. However, our method has major counterexamples 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.

A major challenge is the collection of training material for the patch classes to be recognized. The patches with which the classifiers are trained have to be manually classified. Consequently, the development of an automatic scene classifying system requires substantial effort since for all relevant patch classes, sets of reference patches have to be manually collected. For a given class, the other classes act as counterexamples. We are currently looking into several directions to reduce this burden. One approach would be to generate more counterexamples by combining existing patches. Another direction is the use of one class classification algorithms that only require positive examples [51].

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 10.2, 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 misclassification 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.

10.6 Applications

The scene classifier has been embedded into the VICAR system for content based video retrieval. In addition, the same visual alphabet approach has been used for other video classification applications such as porn filtering, sewage inspection and skin infection detection. The initial versions of these classifiers were built within very short time frames and with sufficient classification accuracy. This provides further evidence that our approach is a generally applicable method to quickly build robust domain specific classifiers.

One of the reasons for its success in these areas, is its user-centered approach: the system can easily learn knowledge of the domain involved, by showing it new patch types and so creating a new visual alphabet, simply by selecting the relevant regions or areas in the image. In this section we will describe a number of these applications in a bit more detail.

10.6.1 Vicar

The scene classifier has been integrated into the Vicar Video Navigator [2]. 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 Figure 10.3 and 10.4, an example search is shown from a custom 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.

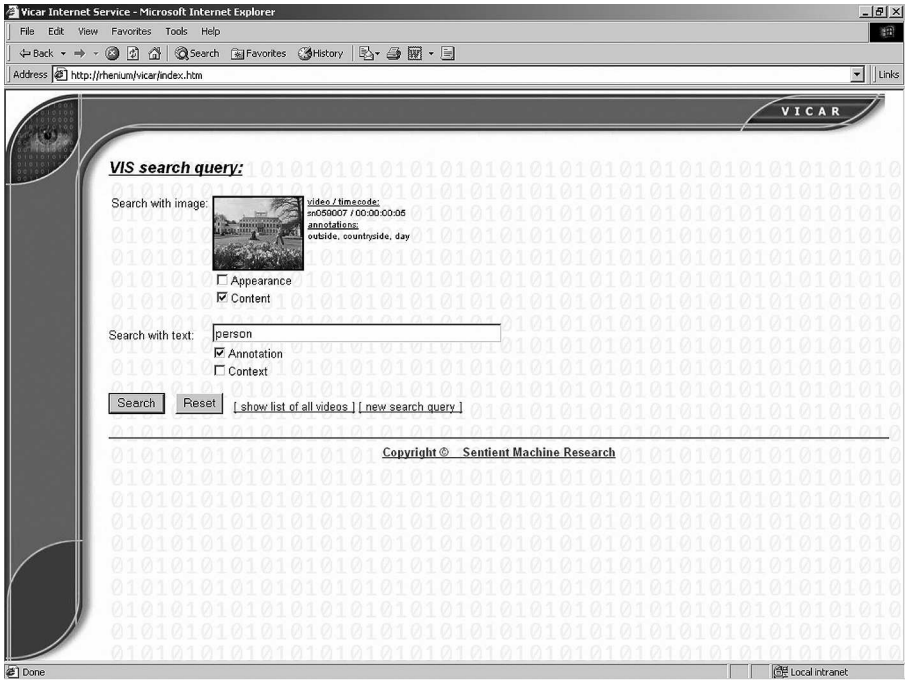

Fig. 10.3. A query for video material.

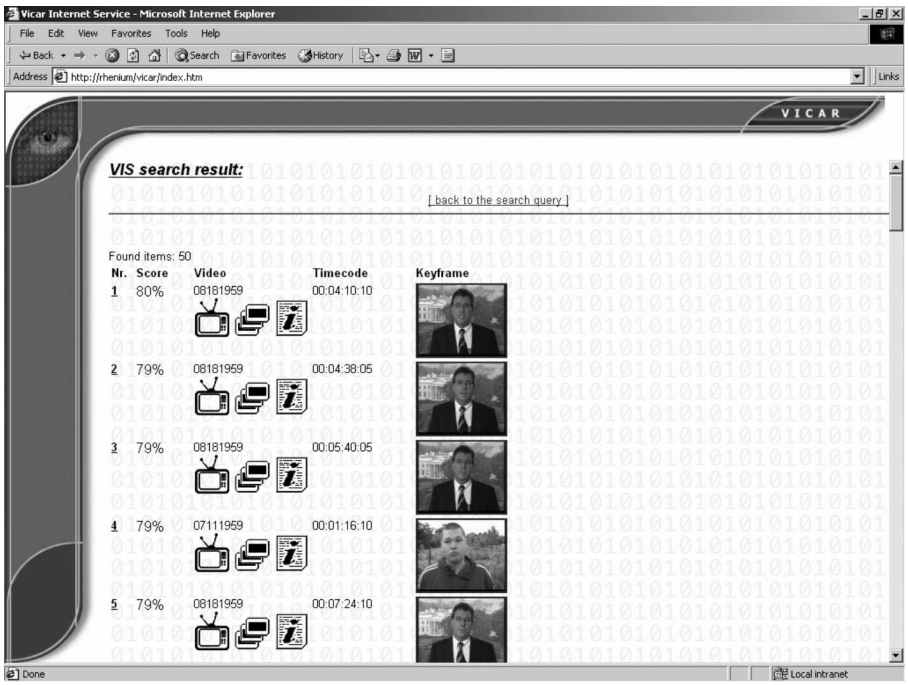

Fig. 10.4. The result of a query for video material.

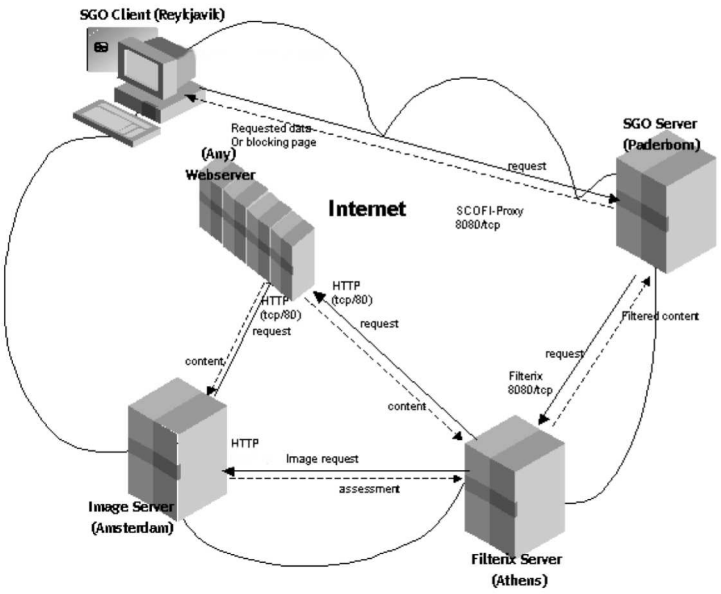

Fig. 10.5. Different components of the SCOFI system: authentication server, text filtering server and porn image classification server

10.6.2 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 smart cards enabled authentication server to enable safe web surfing (see Figure 10.5). The text classifier and the proxy server have been developed by Demokritos, Greece, and are part of the FilterX system [52].

For this application of the system, we first created image representations using the patch classification network as mentioned in Section 10.4.1. 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

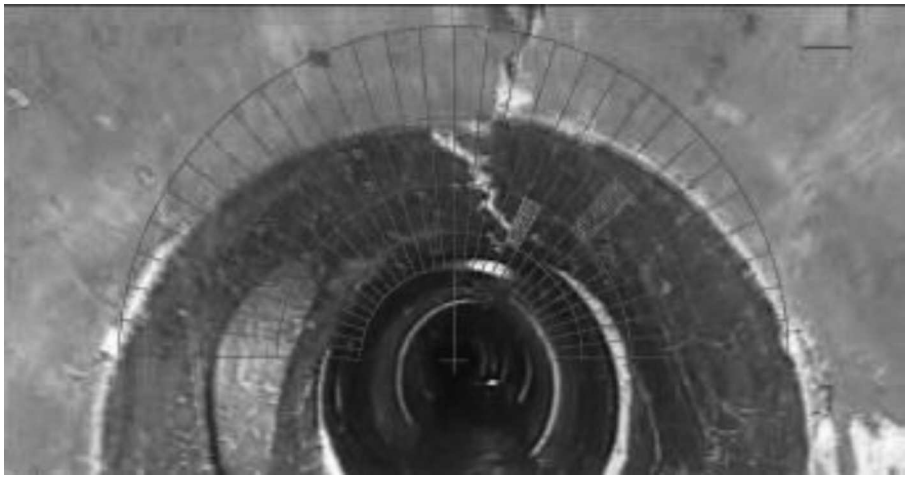

Fig. 10.6. A spherical grid is placed on video footage of a sewer.

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. For a description of the complete results, we refer to Israël [53]. 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.

10.6.3 Sewer Inspection

Our image classification approach is also applied to support the inspection of sewers in the RESEW project (EU GROWTH program for competitive and sustainable growth). Many European cities are spending increasing amounts to improve their sewage systems, so the inspection of deteriorating structures is becoming more and more important.

Currently, robots are used for sewer inspection, but these are completely controlled by operators and the video material that is collected is analyzed manually, which is a costly, time consuming and an error prone process. For instance, a UK based waste water utility company taking part in the project has 7,000 recent tapes of video material available, corresponding to thousands of kilometers of sewers. Real time monitoring of the entire system would increase the need of automated analysis even further.

Automated and integrated systems for damage identification and structural assessment that are based on video analysis can be used to increase the speed and accuracy of the inspection and evaluation process and lower the cost. To prove the feasibility of the above the project partners have develrf an integrated and automated detection, classification, structural assessment and rehabilitation method selection system for sewers based on the processing of Closed Circuit Television (CCTV) inspection tapes. The research prototype provides the user with an easy, fast and accurate method of sewer assessment. It consists of an intuitive interface to the sewage network with typical Geographic Information System functionality, a digital archive of indexed CCTV inspection tapes and a classification module to analyze video material for defects.

The RESEW classification method builds on the approach presented in this chapter. The primary goal of the classifier is to detect longitudinal cracks. First the central 'tunnel eye' is detected and a spherical rather than rectangular grid is placed around it (see Figure 10.6; separate specialized modules extract the sewer joints and any CCTV text).

Neural networks are used to classify the extracted patches into crack and non crack classes. For this local patch classification we achieved an accuracy of 86.9%, with balanced train, validation and test sets of 40,000, 18,562 and 20,262 instances respectively. In the next stage, patch class histograms along the vanishing direction are classified to detect global longitudinal cracks. As an alternative method, a region growing approach is used that takes patch class probabilities as input. The latter approach generally produces more favorable results.

The environment is designed to be utilized in several utility contexts (water networks, sewer networks) where different engineering models are developed (e.g. structural reliability models for water pipes, reliability models taking into account seismic risk, safety models based on digital imagery of sewer interior, rehabilitation models for the previous). The system may be adapted to fit the needs of CCTV inspection of boreholes, shafts, gas and oil pipelines and other construction sectors. Going forward, the methods for analyzing the video material can also be used to build autonomous sewer robots that can explore sewage systems more or less independently.

10.7 Conclusion

In the work presented here, 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 applied as a generic methodology for creating domain specific image classifiers for content-based retrieval and filtering. This is demonstrated by its success in various applications such as the Vicar Video Navigator video search engine, the RESEW sewer inspection system, and the SCOFI real time filter for pornographic image material on the Internet.

Acknowledgments

The work presented in this chapter was partially supported by the EU projects VICAR (IST-24916), SCOFI (IAP-2110; `http://www.scofi.net/`), and RESEW (GRD1-2000-25579). We would like to thank all project team members involved in these projects. We especially thank Robert Maas for his work on the texture algorithm. Further, we gratefully acknowledge the reviewers, for their valuable comments on the manuscript.

References

1. Israël M, Broek E L van den, Putten P van der, Uyl M J den. Automating the Construction of scene classifiers for Content-Based Video Retrieval. In: Khan L, Petrushin VA, editors. Proceedings of the Fifth ACM International Workshop on Multimedia Data Mining (MDM/KDD'04). Seattle, WA, USA; 2004. p. 38–47.
2. Putten P van der. Vicar Video Navigator: Content Based Video Search Engines Become a Reality. Broadcast Hardware International, IBC edition; 1999.
3. Gibson J. The Ecological Approach to Visual Perception. Houghton Mifflin, Boston; 1979.
4. Niblack W, Barber R, Equitz W, Flickner M, Glasman E, Petkovic D, et al. The QBIC project: Querying images by content using color, texture, and shape. Proceedings of SPIE (Storage and Retrieval for Image and Video Databases) 1993;1908:173–87.
5. Flickner M, Sawhney H, Niblack W, Ashley J, Huang Q, Dom B, et al. Query by Image and Video Content: The QBIC System. IEEE Computer 1995;28(9):23–32.
6. Smith JR, Chang SF. Querying by color regions using the VisualSEEk content-based visual query system. The AAAI Press; 1997. p. 23–42.
7. Cox IJ, Miller ML, Minka TP, Papathomas TV. The bayesian image retrieval system, PicHunter: theory, implementation, and psychophysical experiments. IEEE Transactions on Image Processing 2000;9(1):20–37.
8. Gevers Th, Smeulders AWM. PicToSeek: Combining color and shape invariant features for image retrieval IEEE Transactions on Image Processing 2000;9(1):102–19.
9. Wang JZ. Integrated region-based image retrieval. Boston: Kluwer Academic Publishers; 2001.

10. IBM Research. MARVEL: Multimedia Analysis and Retrieval System. Intelligent Information Management Dept., IBM T. J. Watson Research Center; 2005.
11. van den Broek EL, Kok T, Schouten ThE, Hoenkamp E. Multimedia for Art ReTrieval (M4ART). Proceedings of SPIE (Multimedia Content Analysis, Management, and Retrieval) 2006;6073 [in press].
12. Wu S, Rahman MKM, Chow TWS. Content-based image retrieval using growing hierarchical self-organizing quadtree map. Pattern Recognition 2005;38(5):707–22.
13. Minka TP, Picard RW. Interactive learning using a "society of models". MIT Media Laboratory Perceptual Computing Section; 1996.
14. Picard RW. Light-years from Lena: video and image libraries of the future. In: Proceedings of the 1995 International Conference on Image Processing; 1995. p. 310–3.
15. Picard RW, Minka TP. Vision Texture for Annotation. Multimedia Systems 1995;3(1):3–14.
16. Belongie S, Carson C, Greenspan H, Malik J. Recognition of Images in Large Databases Using a Learning Framework. University of California at Berkeley; 1997.
17. Carson C, Belongie S, Greenspan H, Malik J. Blobworld: Image Segmentation Using Expectation-Maximization and Its Application to Image Querying. IEEE Transactions on Pattern Analysis and Machine Intelligence 2002;24(8):1026–38.
18. Campbell NW, Mackeown WPJ, Thomas BT, Troscianko T. The Automatic Classification of Outdoor Images. In: Proceedings of the International Conference on Engineering Applications of Neural Networks. Systems Engineering Association; 1996. p. 339–42.
19. Campbell NW, Mackeown WPJ, Thomas BT, Troscianko T. Interpreting Image Databases by Region Classification. Pattern Recognition 1997;30(4):555–63.
20. Lipson P, Grimson E, Sinha P. Configuration based scene classification and image indexing. In: Proceedings of 16th IEEE Conference on Computer Vision and Pattern Recognition. IEEE Computer Society; 1997. p. 1007–13.
21. Ratan AL, Grimson WEL. Training templates for scene classication using a few examples. In: Proceedings of the IEEE Workshop on Content-Based Analysis of Images and Video Libraries; 1997. p. 90–7.
22. Oliva A, Torralba A. Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope. International Journal of Computer Vision 2001;42(3):145–75.
23. Gorkani MM, Picard RW. Texture Orientation for Sorting Photos at a Glance. In: Proceedings of the International Conference on Pattern Recognition; 1994. p. 459–64.
24. Szummer M, Picard RW. Indoor-Outdoor Image Classification. In: IEEE International Workshop on Content-Based Access of Image and Video Databases (CAIVD). Bombay, India: IEEE Computer Society; 1998. p. 42–51.
25. Fung CY, Loe KF. Learning primitive and scene semantics of images for classification and retrieval. In: Proceedings of the 7th ACM International Conference on Multimedia '99. Orlando, Florida, USA: ACM; 1999. p. 9–12.
26. Fung CY, Loe KF. 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. ACM; 1999. p. 301–2.

27. Broek E L van den, Rikxoort E M van, Schouten ThE. Human-centered object-based image retrieval. Lecture Notes in Computer Science (Advances in Pattern Recognition) 2005;3687:492–501.
28. Smith JR, Chang SF. Single color extraction and image query. In: Liu B, editor. Proceedings of the 2nd IEEE International Conference on Image Processing. IEEE Signal Processing Society. IEEE Press; 1995. p. 528–31.
29. Broek E L van den, Kisters PMF, Vuurpijl LG. Content-Based Image Retrieval benchmarking: Utilizing color categories and color distributions. Journal of Imaging Science and Technology 2005;49(3):293–301.
30. Broek E L van den. Human-Centered Content-Based Image Retrieval. Ph.D. thesis. Nijmegen Institute for Cognition and Information, Radboud University Nijmegen; 2005. Available from: `http://eidetic.ai.ru.nl/egon/PhD-Thesis/`.
31. Prasad B, Gupta S, Biswas K. Color and Shape Index for Region-Based Image Retrieval. In: Arcelli C, Cordella L, di Baja GSanniti, editors. Proceedings of 4th International Workshop on Visual Form. Capri, Italy: Springer Verlag; 2001. p. 716–25.
32. Redfield S, Nechyba M, Harris JG, Arroyo AA. Efficient object recognition using color. In: Roberts R, editor. Proceedings of the Florida Conference on Recent Advances in Robotics. Tallahassee, Florida; 2001.
33. Schettini R, Ciocca G, Zuffi S. A survey of methods for colour image indexing and retrieval in image databases. J. Wiley; 2001.
34. Forsyth DA, Ponse J. Computer Vision: A modern approach. Pearson Education, Inc., Upper Saddle River, New Jerssey, U.S.A.; 2002.
35. Berlin B, Kay P. Basic color terms: Their universals and evolution. Berkeley: University of California Press; 1969.
36. Broek E L van den, Rikxoort E M van. Parallel-Sequential Texture Analysis. Lecture Notes in Computer Science (Advances in Pattern Recognition) 2005;3687:532–41.
37. Derefeldt G, Swartling T. Colour concept retrieval by free colour naming: Identification of up to 30 colours without training. Displays 1995;16(2):69–77.
38. Derefeldt G, Swartling T, Berggrund U, Bodrogi P. Cognitive color. Color Research & Application 2004;29(1):7–19.
39. Goldstone RL. Effects of categorization on color perception. Psychological Science 1995;5(6):298–304.
40. Kay P. Color. Journal of Linguistic Anthropology 1999;1:29–32.
41. Roberson D, Davies I, Davidoff J. Colour categories are not universal: Replications and new evidence from a stone-age culture. Lanham, Maryland: University Press of America Inc.; 2002.
42. Lin T, Zhang HJ. Automatic Video Scene Extraction by Shot Grouping. In: Proceedings of the 15th IEEE International Conference on Pattern Recognition. vol. 4. Barcelona, Spain; 2000. p. 39–42.
43. Gevers Th, Smeulders AWM. Color Based Object Recognition. Pattern Recognition 1999;32(3):453–64.
44. Jain AK, Karu K. Learning Texture Discrimination Masks. IEEE Transactions on Pattern Analysis and Machine Intelligence 1996;18(2):195–205.
45. Palm C. Color texture classification by integrative Co-occurrence matrices. Pattern Recognition 2004;37(5):965–76.

46. Rikxoort E M van, Broek E L van den, Schouten ThE. Mimicking human texture classification. Proceedings of SPIE (Human Vision and Electronic Imaging X) 2005;5666:215–26.
47. Broek E L van den, Rikxoort E M van, Kok T, Schouten ThE. M-HinTS: Mimicking Humans in Texture Sorting. Proceedings of SPIE (Human Vision and Electronic Imaging XI) 2006;6057 [in press].
48. Rosenfeld A. From image analysis to computer vision: An annotated bibliography, 1955-1979. Computer Vision and Image Understanding 2001;84(2):298–324.
49. Jähne B. Practical Handbook on Image Processing for Scientific Applications. CRC Press; 1997.
50. Brox T, Weickert J, Burgeth B, Mrázek P. Nonlinear structure tensors. Image and Vision Computing 2006;24(1):41–55.
51. Tax DMJ. One-class classification; concept learning in the absence of counter-examples. Ph.D. thesis. Delft University of Technology; 2001.
52. Chandrinos KV, Androutsopoulos I, Paliouras G, Spyropoulos CD. Automatic Web Rating: Filtering Obscene Content on the Web. In: Borbinha J, Baker T, editors. Proceedings of the 4th European Conference on Research and Advanced Technology for Digital Libraries; 2000. p. 403–6.
53. Israël M. ParaBot: Text and Image classification for the internet. Amsterdam, The Netherlands: Sentient Machine Research; 1999.