Computer Vision and Image Understanding
Vol. 75, Nos. 1/2, July/August, pp. 133–149, 1999
Article ID cviu.1999.0769, available online at http://www.idealibrary.com on IDEAL®

Features and Classification Methods to Locate Deciduous Trees in Images

Niels Haering and Niels da Vitoria Lobo

School of Computer Science, University of Central Florida, Orlando, Florida 32816
E-mail: haering@cs.ucf.edu, niels@cs.ucf.edu

We compare features and classification methods to locate deciduous trees in images. From this comparison we conclude that a back-propagation neural network achieves better classification results than the other classifiers we tested. Our analysis of the relevance of 51 features from seven feature extraction methods based on the graylevel co-occurrence matrix, Gabor filters, fractal dimension, steerable filters, the Fourier transform, entropy, and color shows that each feature contributes important information. We show how we obtain a 13-feature subset that significantly reduces the feature extraction time while retaining most of the complete feature set's power and robustness. The best subsets of features were found to be combinations of features of each of the extraction methods. Methods for classification and feature relevance determination that are based on the covariance or correlation matrix of the features (such as eigenanalyses or linear or quadratic classifiers) generally cannot be used, since even small sets of features are usually highly *linearly* redundant, rendering their covariance or correlation matrices too singular to be invertible. We argue that representing deciduous trees and many other objects by rich image descriptions can significantly aid their classification. We make no assumptions about the shape, location, viewpoint, viewing distance, lighting conditions, and camera parameters, and we only expect scanning methods and compression schemes to retain a "reasonable" image quality. © 1999 Academic Press

1. INTRODUCTION

Locating trees in images is useful for processing World Wide Web images and image and video databases and in robot navigation and may generally help us with the image understanding problem. We will probably deny a robot the attribute "intelligent" unless it can act and react in our environment. To achieve this, robots may need to perceive their environment similarly to the way humans do so. Most certainly it would help if robots had a notion of the different types of objects around them, deciduous trees being one such class of objects.

Over the past few years we got used to searching the World Wide Web for textual information. Similarly the efficient location of visual information in images and videos on the World Wide Web is desirable. While interfaces already exist that allow users to specify types and shapes of objects of interest [17, 33], our contribution provides a reliable tool that enables the robust detection of deciduous trees in unconstrained images. For our work we use images found on line, like those in Figs. 10 and 11. These images demonstrate that we cannot make assumptions about the shape and location of deciduous trees in images, the viewpoints or viewing distances, the lighting conditions, or the camera parameters. Likewise, we must allow imperfect scanning methods, lossy compression schemes, and other postprocessing, with the only constraint that the resulting images retain a "reasonable" quality and are not unreasonably distorted.

1.1. Previous Work

Previous work can be grouped into four basic categories. For each category we briefly mention representative approaches.

Remote sensing. The remote sensing community has a continuing interest in terrain classification [26, 27]. Their work is similar in that they too have to deal with the fusion of multimodal information. On the other hand, their work differs in that images are obtained by the same camera and through a fixed image formation process. Images are taken from a fixed viewpoint and viewing distance. But most significantly, they use electromagnetic waves from outside the visual spectrum to help in the classification of terrain.

Texture-based classification. Work by Haralick *et al.* ([16]) used texture measures to classify regions of aerial and satellite images, photographed in the visual spectrum, into eight classes: Old Residential, New Residential, Lake, Swamp, Marsh, Urban, Rail, and Scrub/Wood. Scrub and wood were classified together as "Scrod" due to their visual similarity.

Color-based classification. Work by Fischler ([10]) dealt with terrain classification solely based on the color of individual pixels. He classified natural scenes into five classes: Sky, Water, Rocks, Ground, and Live Vegetation. Work in the area of image and video retrieval describes color-histogram-based methods for simple retrieval.

Image and video retrieval. With little processing per image we can obtain relatively meaningless but fast image descriptors like color histograms. With minutes, hours, or even days of processing, existing computer vision systems can be used to locate and recognize one of a small number of known objects.

Image and video retrieval methods try to provide more information about a larger number of objects faster. Recent approaches describe images in terms of blobs [3] of coherent color and texture attributes to facilitate the interpretation of image content in a meaningful way, or they focus on dichotomies like indoor vs outdoor scenes [37] or city images vs landscapes [38].

We present a method that focuses on the task of locating deciduous trees, while significantly relaxing the constraints on the imaging conditions under which they can be recognized.

1.2. Object Recognition

Much of object recognition research has concerned itself with the problem of locating a small number of known objects in a scene containing one or more of these objects [18, 29]. Viewer- and object-centered approaches to object recognition tend to focus on the geometric arrangement of simple features obtained using a unique method (edges/lines from edge detectors, "interest points" from interest operators, etc). These feature values are either taken directly as "signatures" for the depicted objects [1] or taken indirectly through the derivation of "invariants" from them [18, 32]. An alternative approach uses Shape-from-X methods to infer the shape of the depicted objects before registering 3D models to the inferred object shape. Such descriptions are then used to generate hypotheses as to the size and orientation/viewpoint of the known objects. Apart from the "Shape-from-X," "point correspondence," and "segmentation" problems with these approaches, they mostly assume that one of the known objects *is* present and visible in the image. Therefore, objects are even detected and recognized in images that do not show them. The uniformity of the extracted features leads to an intractable number of possible alignments between noisy feature points in the image and potential candidates from the model(s). Our approach attempts to deal with all the mentioned problems through the use of richer bottom-up descriptions of images and by training a classifier to tell deciduous trees from everything but deciduous trees. Admittedly the latter class (nondeciduous-trees) is rather large and diverse, and the classifier must generalize well to achieve good classification. Richer descriptions of the depicted object form "signatures" even before they are related to each other by their geometric relationships (of course geometric combinations of such signatures can be used to enhance the power of this approach further). For example, while many objects may be green, fewer objects both are green and exhibit little directional energy, and even fewer objects are green, exhibit little directional energy, have high entropy, few collinear step edges, appear similar at a range of scales, etc. Our approach aims to exploit the fact that in many cases objects are more easily identified by an array of different observable measures than by the measurements of one kind of feature at a number of "key" locations.

1.3. Our Approach

Thus, we argue that a single measure, observation, or model is unlikely to enable robust recognition of deciduous trees. Variations in the objects themselves, the imaging conditions, and image compression complicate the classification task.

The *appearance* and *shape* of trees, even if unoccluded and from only one deciduous species, vary greatly. Images of trees vary in *viewpoint* (frontal views, views from below trees, panoramic views, or aerial or elevated views), *viewing distance* (anything between a distant forest and a number of individual leaves is considered a tree), *lighting conditions* (natural vs artificial lighting, broad daylight vs sunrise/sunset, or clear vs overcast vs foggy/misty). Additionally, we have no information about the camera parameters, and images are obtained or transformed by noisy processes such as scanning and lossy compression. GIF compressed images, for example, only have 256 unique colors. If texture features are based on grayvalue representations of images, some of the colors may collapse to the same gray level. The resulting severe quantization has complex consequences for many texture measures. The range of JPEG compression schemes, their lossy representations of 8×8 blocks in the image, and the aliasing effects at the borders of these blocks further complicate the task for texture measures.

Hence, we believe that no one feature, such as texture, color, size, or inferred shape, can robustly locate trees in unconstrained images. We derived 51 features from seven methods based on the gray-level co-occurrence matrix [16], Gabor filters [25], box-counting-based fractal dimension estimators [6], Fourier-transform-based features [5], color, entropy, and steerable filters [31]. The underlying methods are described in detail in [6, 14, 16, 22, 31, 39].

We use a back-propagation neural network to combine the features and to obtain a consistent and robust classification. Since extracting the feature space representation for images takes a long time we show how a greedy algorithm can improve on good random selections of features. This approach produces a subset of 13 features which reduces the computation time significantly while retaining the classification accuracy and robustness.

1.4. Deciduous, Coniferous, and Evergreen Trees

The appearance of deciduous trees during spring, summer, and fall is dominated by their foliage cover. Their branches are largely hidden. During the winter season, on the other hand, the foliage of deciduous trees is not visible. Instead their appearance is now dominated by the branches that were hidden during the other seasons. Visually deciduous trees during winter and the other seasons have little in common. Therefore, their recognition during the winter season should probably be treated as a separate problem. We believe that telling deciduous trees from evergreen trees (including most coniferous trees) in winter should be easier than their detection during the remaining seasons, but we have not attempted this so far.

Our intuitive emphasis on tree color turns out to be a rather week cue for the detection of deciduous trees. The following two observations may help to explain this somewhat surprising result:

• Throughout the year their leaves may take on all colors except blue.

• Areas exposed to direct sunlight appear slightly more yellow and orange than their surface hues. Areas that are not exposed to direct sunlight often show a *significant bias* toward blue. In absence of direct sunlight the ambient blue of the sky becomes the second most powerful light source, coloring all shaded objects.

1.5. Outline

We describe the features we extract from images in Section 2, and the classifier, used to recognize deciduous trees, in Section 3. We show the results of that classifier on a number of different feature sets in Section 4.

In Section 5 we contrast the performance of a number of different classifiers and we compare a number of alternative methods to determine the relevance of features. Finally, Section 6 concludes with a summary of all our findings.

2. THE FEATURES

We classify trees using features extracted from color images. Some of these features are representations of individual pixels in the image, such as the hue, saturation, and value; others are the outputs of grayimage-patch-based filters.

2.1. Fourier-Transform-Based Features

Some measures commonly used with Fourier-transform-based methods are (i) wedge sampling, (ii) annular-ring sampling, and (iii) parallel-slit sampling.

Many textures differ significantly in the domains of the annular-ring and parallel-slit measures; however, for our purpose of discriminating tree and non-tree areas, angular wedge sampling is most expressive. Fourier transforms (FTs) of images and image patches containing humanmade structures often have line- or wedge-shaped areas of high spectral power that pass through the center as shown in the image/FT pairs in Figs. 1a and 1b. Summing the power in fixed angular intervals for all directions in the FT of the image lets us separate common from uncommon orientations in the image. The shaded wedge in Fig. 2 shows such an angular interval. A circular mask has been imposed so that the power in the diagonal directions is not unfairly biased. Once the power in each angular interval has been determined, we obtain the minimum and maximum angular power and use the normalized ratio $(\max - \min)/(\max + \min)$ to determine the amount of structure in the patch.

Larger values for this wedge measure indicate greater "regularity" in some direction in the image patch; smaller values indicate less "regularity," in terms of parallel lines, bars, and edges. Since we are comparing the ratios between the maximum and minimum values, this measure is rotation invariant.

Performing the above procedure on fixed-size image patches, we obtain local measures of the regularity of these patches. We obtained very similar results for patch sizes 16×16, 32×32, and 64×64 pixels.

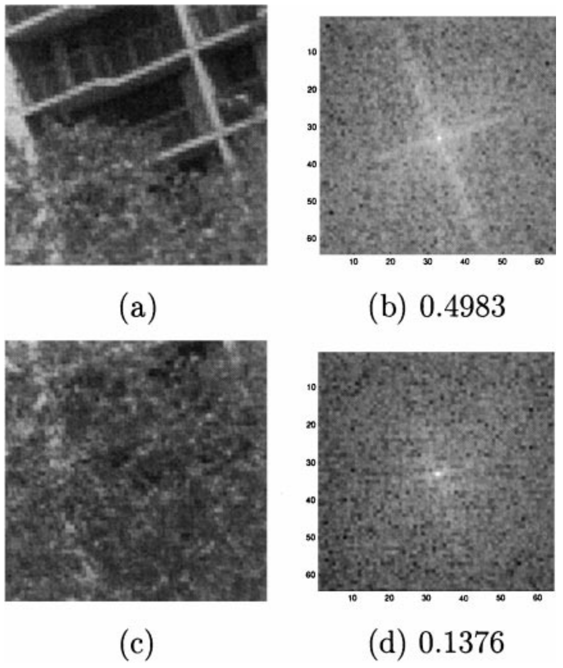

FIG. 1. An image containing humanmade and tree areas (a) and its Fourier transform (b). An image of leaves of a tree (c) and its Fourier transform (d). The numbers associated with (b) and (d) are the structure measure (described in Section 2.1) for images (a) and (c). Images (a) and (c) are copyright Gerhard Ortner, with whose permission they are used.

2.2. Gabor Filter Measures

The image (in the spatial domain) is described by its 2D intensity function. The Fourier transform of an image represents the same image in terms of the coefficients of sine- and cosine-basis functions at a range of frequencies and orientations. Similarly, the image can be expressed in terms of coefficients of other basis functions. Gabor [13] used a combined representation of space and frequency to express signals in terms of "Gabor" functions,

$$F_{\theta,\nu}(x) = \sum_{i=1}^{n} a_i(x) g_i(\theta, \nu), \tag{1}$$

where θ represents the orientation and ν the frequency of the

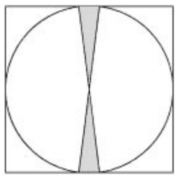

FIG. 2. The sum of the power of the Fourier transform inside the shaded vertical angular interval is a measure of the "structure" present in an image patch.

complex Gabor function

$$g_i(\theta, \nu) = \exp(i\nu(x\cos(\theta) + y\sin(\theta)))\exp\left(-\frac{x^2+y^2}{\sigma^2}\right). \quad (2)$$

Gabor filters have gained popularity in multiresolution image analysis [11, 13], despite the fact that they do not form an orthogonal set, which means that their coefficients cannot be obtained by convolution of the image with the basis functions as is done in the next section, where we employ steerable bar and step-edge detectors. Gabor-filter-based wavelets have recently been shown by Manjunath and Ma [22] to be fast and useful for the retrieval of image data.

We convolve each image patch with Gabor filters tuned to four different orientations at three different scales. Next, the average and range of the four measures at each scale are computed. Finally, to also make the measurements somewhat scale-invariant, we obtain the following four texture measures:

- The average of the orientation responses at all scales.
- The average of the scales' orientation response range.
- The range of the scales' averaged orientation responses.
- The range of the scales' orientation response range.

2.3. *Steerable Bar- and Step-Edge Filters*

Since many humanmade structures exhibit a large amount of regularity in the form of parallel lines and bars, patches with few dominant orientations are less likely to represent trees. On the other hand, the irregular leaf and branch structure of trees often exhibits a greater variety of weak orientations.

By binning orientations appropriately, we can use the *number* and *strength* of different orientations in an image patch to distinguish between patches belonging to humanmade scenes (which usually have fewer but stronger distinct orientations) and those belonging to natural scenes.

Steerable bar and step-edge detecting filters are used to obtain the dominant orientation for each image patch. The result of this routine is an orientation image indicating the orientation of the predominant step or bar edge at each location.

To measure the energy of an image (in terms of line and step edges), the convolutions with two filters that are 90° out of phase (i.e., filters that form a quadrature pair) can be squared and summed. To detect lines, one filter type is an even function that can be decomposed solely into cosine terms, while for the detection of step edges, the other is an odd function that can be decomposed solely into sine terms.

In order to obtain a good orientation resolution of lines and step edges in the image, one could convolve the image with a large number of orientations of the quadrature pair. However, convolutions are slow, and the accuracy of such an approach depends on (a) the sampling frequency and (b) the interpolation of the sampled convolutions.

Canny showed [4] how the outputs of only two orthogonal filters (the first derivatives of a Gaussian in the x and y directions) are sufficient to find the orientation and power of step edges. In order to handle both kinds of edges, Freeman and Adelson [12] use a quadrature pair consisting of the second derivative of a standard Gaussian (the even part of the filter pair) and its Hilbert transform (the odd part of the filter pair). Instead of finding the outputs at all orientations, the concept of *steering* was introduced, in which the convolution with a filter at any orientation can be synthesized by a linear combination of the convolutions with a small basis set of filters,

$$F_\theta^{[n]} = \sum_{i=1}^{n} \sigma_i a_i(x) b_i(\theta). \quad (3)$$

Using three basis and interpolating functions for the Gaussian part and four for the Hilbert transform part of the filter, they can locate both lines and step edges exactly (as opposed to the double edges detected at a line by using the methods of Canny and others). Since the standard Gaussian has a low orientation selectivity, the kernels are only optimal in the presence of a single line or step edge in the image and blur the responses in the presence of more than one edge or line. To improve the orientation selectivity, they suggested using higher-order derivatives of Gaussians as kernels. However, even a fourth-order derivative of a Gaussian together with its Hilbert transform does not yield good results if lines or step edges cross each other at angles other than 90°. Perona [31] demonstrated a general constructive method to construct basis and interpolating functions and showed that all functions that are polar-separable with sinusoidal θ components are steerable. Examples of such functions are shown in Fig. 3.

We used this method to obtain a steerable function set for a quadrature pair (G_{yy}, H_{yy}), where G_{yy} is the second derivative along the y-axis of an elongated Gaussian kernel $G(x, y, \sigma_x, \sigma_y) = \exp(-((x/\sigma_x)^2 + (y/\sigma_y)^2))$ shown in Fig. 4a and H_{yy} is the Hilbert transform of G_{yy} shown in Fig. 4b.

For multiple occurrences of lines and step edges, good angular resolution (orientation selectivity) was obtained when the ratio $\frac{\sigma_x}{\sigma_y}$ was at least $\frac{1}{4}$. Perona [31] showed an efficient method that places the second derivative of the Gaussian in the real part of the complex kernel and its Hilbert transform in the imaginary part.

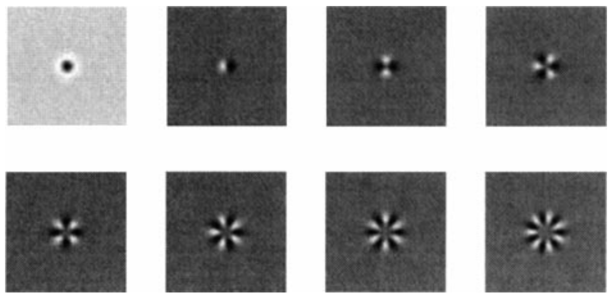

FIG. 3. Examples of polar separable functions with sinusoidal θ component corresponding to $a_0, \ldots, a_7$.

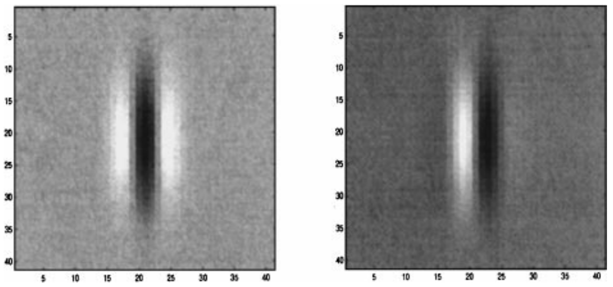

FIG. 4. Filters used to measure the edge energy in an image. The second derivative of an elongated Gaussian (left) is used to detect lines in the image. Its Hilbert transform (right) is used to detect step edges in the image.

The n-term approximation of the function we want to steer can be written as

$$F_\theta^{[n]} = \sum_{i=1}^{n} \sigma_i a_i(\mathbf{x}) b_i(\theta), \quad \forall\theta \in S^1, \forall\mathbf{x} \in \mathbb{R}^2, \tag{4}$$

where the σ_i weight the product of the ith filter basis function a_i (the coefficients of the 2D Fourier series) and the corresponding interpolating function b_i (note that the b_i are the frequency basis functions of the Fourier series).

The values for σ_i, a_i, and b_i are obtained by finding the Fourier series of the function $h(\theta)$, which is the integral of the product of the function with rotated versions of itself,

$$h(\theta) = \int_{\mathbb{R}^2} F_\theta(x)\overline{F_{\theta'=0}(x)}\, dx, \tag{5}$$

where the integral ranges over all 2D space ($\mathbb{R}^2$) and $\overline{(\cdot)}$ represents the complex conjugate. Note that $F_{\theta'=0}(x) = F(x)$.

Expanding $h(\theta)$ as a Fourier series we can read off the filter's (2D) basis functions a_i and the corresponding interpolating functions b_i:

$$\sigma_i = \sqrt{h(\nu_i)} \tag{6}$$

$$b_i(\theta) = \exp(i\nu\theta) \tag{7}$$

$$a_i(\mathbf{x}) = \sigma_i^{-1} \int_{S^1} \overline{F_\theta(\mathbf{x})} \exp(i\nu\theta\, d\theta). \tag{8}$$

The σ_i terms are used only for error analysis. For details see [31].

These filters are used to obtain the oriented energy of both step and bar edges. Although we initially envisaged them as aiding in the recognition of deciduous trees in winter, when their leaves are missing, the orientation analysis also turned out to be useful for the recognition of leaves and trees in summer.

2.4. *Graylevel Co-occurrence Matrix Measures*

Let $p(i, j, d, \theta) = P(i, j, d, \theta)/R(d, \theta)$, where $P(\cdot)$ is the graylevel co-occurrence matrix (GLCM) of pixels separated by distance d whose orientation is θ and where $R(\cdot)$ is a normalization constant that causes the entries of $P(\cdot)$ to sum to 1.

In texture classification, the following measures have been defined, see for example [5, 16]: The *angular second moment (E)* (also called the *energy*) assigns larger numbers to textures whose co-occurrence matrix is sparse,

$$E(d, \theta) = \sum_{j=1}^{N_g} \sum_{i=1}^{N_g} [p(i, j, d, \theta)]^2.$$

The *difference angular second moment (DASM)* assigns larger numbers to textures containing only a few graylevel patches,

$$DASM(d, \theta) = \sum_{n=0}^{N_g} p_{x-y}(n, d, \theta)^2,$$

where $p_{x-y}(n, d, \theta) = \sum_{j=1}^{N_g} \sum_{i=1 |i-j|=n}^{N_g} p(i, j, d, \theta)$.

The *contrast (Con)* is the moment of inertia around the co-occurrence matrix's main diagonal. It is a measure of the spread of the matrix values and indicates whether pixels vary smoothly in their local neighborhood,

$$Con(d, \theta) = \sum_{n=0}^{N_g-1} n^2 \left[\sum_{\substack{j=1 \\ |i-j|=n}}^{N_g} \sum_{i=1}^{N_g} p(i, j, d, \theta) \right].$$

The *inverse difference moment (IDM)* measures the local homogeneity of a texture. It weights the contribution of the co-occurrence matrix entries inversely proportional to their distance to the main diagonal,

$$IDM(d, \theta) = \sum_{i=1}^{N_g-1} \sum_{j=1}^{N_g-1} \frac{1}{1 - (i - j)^2} p(i, j, d, \theta).$$

The *mean (M)* is similar to the contrast measure above but weights the off-diagonal terms linearly with the distance from the main diagonal, rather than quadratically as for the contrast,

$$M(d, \theta) = \sum_{n=0}^{N_g-1} n \left[\sum_{\substack{j=1 \\ |i-j|=n}}^{N_g} \sum_{i=1}^{N_g} p(i, j, d, \theta) \right].$$

Similar to the angular second moment, the *entropy (H)* is small for textures that give rise to co-occurrence matrices whose sparse entries have strong support in the image. It is maximal for matrices whose entries are all equally large,

$$H(d, \theta) = -\sum_{j=1}^{N_g} \sum_{i=1}^{N_g} p(i, j, d, \theta) \log(p(i, j, d, \theta)).$$

Other measures are *sum entropy (SH),*

$$SH(d,\theta) = -\sum_{n=0}^{2*N_g-1} p_{x+y}(n,d,\theta)\log(p_{x+y}(n,d,\theta)),$$

where $p_{x+y}(n,d,\theta) = \sum_{j=1}^{N_g}\sum_{i=1 | i+j|=n}^{N_g} p(i,j,d,\theta)$, *difference entropy (DH),*

$$DH(d,\theta) = -\sum_{n=0}^{N_g} p_{x-y}(n,d,\theta)\log(p_{x-y}(n,d,\theta)),$$

and *difference variance (DV),*

$$DV = -\sum_{n=2}^{2N_g}(n-DH)^2 p_{x-y}(n,d,\theta).$$

The *correlation (Cor)* measure is an indication of linear structure of a texture. This and the next two measures use $\mu_x = \sum_i i \sum_j p(i,j,d,\theta)$ and $\mu_y = \sum_j j \sum_i p(i,j,d,\theta)$,

$$Cor(d,\theta) = \frac{\sum_{i=1}^{N_g-1}\sum_{j=1}^{N_g-1} ij p(i,j,d,\theta) - \mu_x * \mu_y}{\sigma^2}.$$

Shade (S):

$$S(d,\theta) = \sum_i^{N_g}\sum_j^{N_g}(i+j-\mu_x-\mu_y)^3 p(i,j,d,\theta).$$

Prominence (P):

$$P(d,\theta) = \sum_i^{N_g}\sum_j^{N_g}(i+j-\mu_x-\mu_y)^4 p(i,j,d,\theta).$$

Note that the directionality of a texture can be measured by comparing the values obtained for a number of the above measures as θ is changed. The above measures were computed at four angles (0°, 45°, 90°, and 135°) using $d = 1$. To make the measures rotation-invariant, we use the average and range over the four orientations to obtain two features for each type of measure. For further discussion of these graylevel co-occurrence matrix measures, see [5, 16].

2.5. Fractal Dimension Measures

The underlying assumption for the use of the *fractal dimension (FD)* for texture classification and segmentation is that images or parts of images are self similar at some scale.

Various methods that estimate the FD of an image have been suggested:

- Fourier-transform-based methods [28],
- box-counting methods [6, 21], and
- 2D generalizations of Mandelbrot's methods [30].

The principle of self-similarity may be stated as: If a bounded set A (object) is composed of N_r nonoverlapping copies of a set similar to A, but scaled down by a reduction factor r, then A is self-similar. From this definition, the fractal dimension D is given by

$$D = \frac{\log N_r}{\log r}.$$

The FD can be approximated by estimating N_r for various values of r and then determining the slope of the least-squares linear fit of $(\log N_r / \log(1/r))$. The differential box-counting method outlined in Chaudhuri *et al.* [6] is used to achieve this task.

Three features are calculated based on

- The actual image patch $I(i,j)$,
- the high-graylevel transform of

$$I(i,j),\ I_1(i,j) = \begin{cases} I(i,j) - L_1 & I(i,j) > L_1 \\ 0 & \text{otherwise,} \end{cases}$$

- the low-graylevel transform of

$$I(i,j),\ I_2(i,j) = \begin{cases} 255 - L_2 & I(i,j) > 255 - L_2 \\ I(i,j) & \text{otherwise,} \end{cases}$$

where $L_1 = g_{\text{min}} + (g_{\text{avg}}/2)$, $L_2 = g_{\text{max}} - (g_{\text{avg}}/2)$, and g_{min}, g_{max}, and g_{avg} are the minimum, maximum, and average gray values in the image patch, respectively.

The fourth feature is based on multifractals, which are used for self-similar distributions exhibiting nonisotropic and inhomogeneous scaling properties. Let k and l be the minimum and maximum gray level in an image patch centered at position (i,j), let $n_r(i,j) = l - k + 1$, and let $\mathcal{N}_r = (n_r/N_r)$; then the multifractal, D_2 is defined by

$$D_2 = \lim_{r\to 0}\frac{\log\sum_{i,j}\mathcal{N}_r^2}{\log r}.$$

A number of different values for r are used, and the linear regression of $(\log\sum_{i,j}\mathcal{N}_r^2)/\log r$ yields an estimate of D_2.

2.6. Color Measures

While the intensities of the red, green, and blue components of a color image are highly correlated, the hue, saturation, and value decomposition offers a more independent representation that captures complementary information about the image.

The hue component (θ) can be computed by finding the angle between the color of a pixel and the *red* corner of the color triangle in Fig. 5a (see for example [19] for details),

$$\theta = \cos^{-1}\left(\frac{2r-g-b}{2\sqrt{(r-g)^2+(r-b)(g-b)}}\right),$$

where r, g, and b are the intensities of the red, green, and blue components of the corresponding pixel.

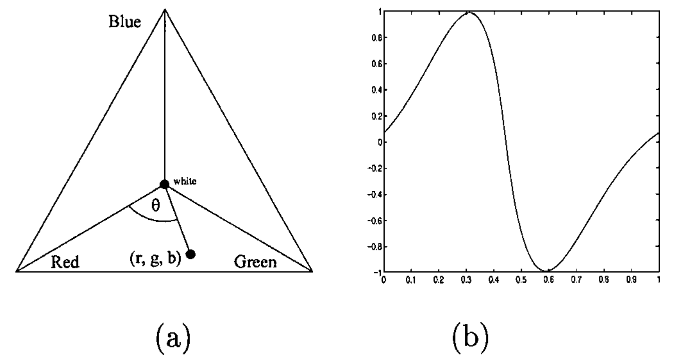

FIG. 5. The color triangle and the function mapping a pixel's hue to its probability of being a leaf pixel.

The *saturation (S)* and *value (V)* components are also defined in terms of r, g, and b:

$$S = 1 - \frac{3}{r+g+b}\min(r, g, b),$$
$$V = \frac{1}{3}(r+g+b).$$

The color–value discontinuity between magenta-red and orange-red (which have maximally different feature values on the hue scale but appear very similar in images) complicates the task unnecessarily. Therefore, each hue value is assigned the corresponding value of a function that transforms the hue into the likelihood of the pixel being a leaf pixel (see Fig. 5(b)).

We also use opponent color measures that contrast the intensities of red vs green ($red/(green+\alpha)$), red vs blue ($red/(blue+\alpha)$), and green vs blue ($green/(blue+\alpha)$), where we used $\alpha = 0.01$ to bound the ratios.

2.7. Entropy Measures

Since leaves and branches appear as rough and "messy" areas at most scales at which trees can be identified, we can use the entropy of image patches to separate them from uniform, smooth, and smoothly varying object surfaces. If $V_{\max}$ is the maximum value in an image patch, the entropy is defined as

$$Entropy = -\sum_{i=0}^{V_{\max}} h_i \log(h_i),$$

where $h_i = n_i/N$ is the ith histogram count n_i divided by the total number of pixels in the image patch (N). We measure the entropy in both the gray-value image and the orientation image described above; both measures are largely rotation-invariant.

3. CLASSIFICATION

A number of factors prevent zero error results. *Often the object identity is unclear*. For instance, should bushes be labeled as tree or nontree areas? What if a bush is actually a small tree? *There is no correct class for class border pixels*. How should an image patch be labeled if roughly half of it depicts a tree and the other half does not?

A related issue concerns the importance of the classification result. Misclassifying a distant coniferous tree as a distant deciduous tree is not as severe as, for example, classifying humanmade structures as trees.

3.1. The Back-Propagation Neural Net

We use a back-propagation neural network to arbitrate among the different features describing the image. Some alternative classification methods are discussed in Section 5. Our back-propagation neural network [8] has a single hidden layer and uses the sigmoidal activation function $\Phi(act) = 1/(1+\exp(-act)) - 0.5$, where *act* is the activation of the unit before the activation function is applied. A single hidden layer in a back-propagation neural network has been shown to be sufficient to uniformly approximate any function (mapping) to arbitrary precision [7]. Although this existential proof does not state that the best network for some task has a single hidden layer, we found one hidden layer adequate. The architecture of the network is shown in Fig. 6. The back-propagation algorithm propagates the (input) function values layer by layer, left to right (input to output) and back-propagates the errors layer by layer, right to left (output to input) as shown in Fig. 6. As the errors are propagated back to the input units, part of each unit's error is being corrected.

3.2. Reducing Feature Redundancy

An important consequence of our approach of combining a large number of different measures to locate tree regions in images is that it is very time-consuming if done sequentially. Parallelism could be exploited easily at the feature and pixel level to produce speedier solutions.

To find the 51D feature space representation for an image of size 512×512 takes just under 1 day on a 200-MHz UltraSPARC. Labeling each pixel of an image of this size as belonging

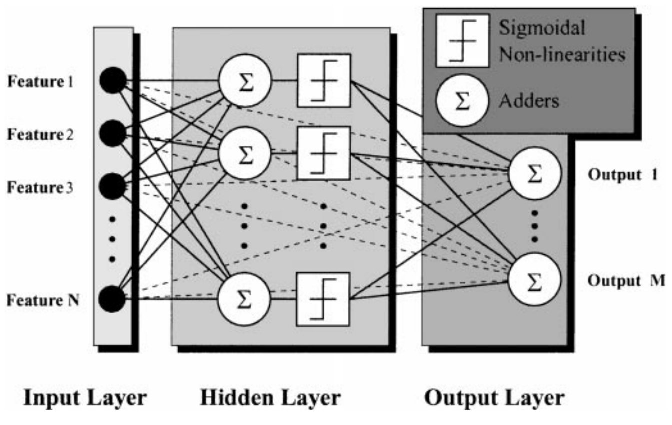

FIG. 6. The network architecture.

to a tree or non-tree region takes about 20 s. The elimination of redundancy in the feature set is therefore an important task. While finding the best set of n features for our classification problem is an intractable problem, we will present a method (in Section 4) that can be used to determine good subsets of features, thus speeding classification.

4. RESULTS

As a means for comparison, we contrast the classification results using all 51 features and those using various subsets of features. In Section 4.4 we show how the architecture of the neural network affects the classification results.

4.1. *Subsets of Features*

There is a prohibitively large number of subsets even for moderate numbers of features. To establish a lower bound of performance for subsets of varying size we first averaged the performance of the classifier on random feature samples. We assume that the discriminatory power of a set of features can be estimated by averaging over the differences due to initialization. In Figs. 7, 9, 17, and 18 the average performance of a given feature set is represented by horizontal lines, while the variation of these solutions is shown by vertical lines.

4.1.1. Random sets of features. Figure 7 shows the errors on the training set as the number of randomly selected features is increased from 1 to all (51) features. Horizontal lines show the average error for the feature sets of the various sizes; vertical lines indicate the range of errors for each feature set. These results should form a lower bound on the performance of any feature selection method. Difficulties with the correct label for each pixel, problems with pixels on the border between tree and nontree regions, and problems due to compression prevent us

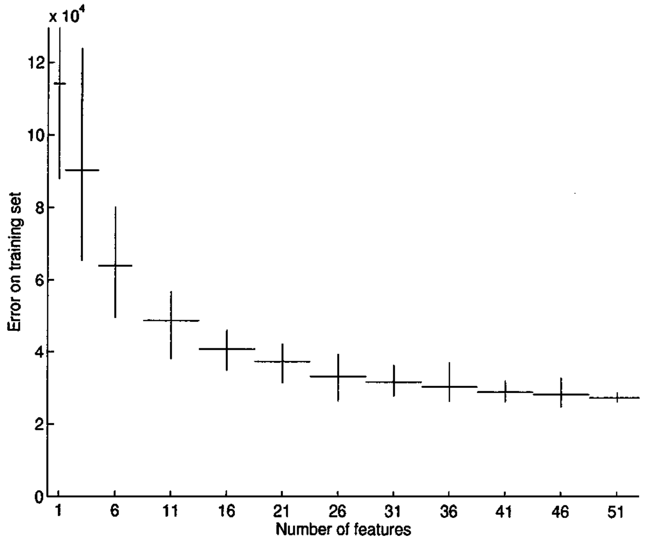

FIG. 7. The performance of random feature sets of various sizes.

from eliminating the error completely. This partly accounts for the fact that the exponential error curve does not approach zero as the number of features is increased. This is not to say that, using additional/alternative methods, we cannot reduce the error further.

4.2. *Good Subsets of Features*

Transitivity does not hold for subsets of features: Given an optimal subset with N features we cannot expect that the addition of the "best" feature not yet in the set will produce an optimal set of size $N + 1$. Likewise, eliminating the "least useful" feature from an optimal set of size $N + 1$ need not produce the best subset of size N. Determining an optimal subset of N features from the entire feature set hence requires the consideration of 2^N such subsets.

4.2.1. A greedy algorithm. Since determining the set of features that enables the best classification performance is an intractable task, we have to settle for methods that can find us good, but suboptimal, feature sets. One such method is a greedy algorithm that starts from the empty set and includes the single most "useful" feature at each step (or its opposite, a greedy algorithm that starts with the entire set and discards the single least "useful" feature at each step). Since this strategy makes local decisions it cannot be expected to find globally optimal solutions. In fact we found that after a number of steps the greedy algorithm has made so many suboptimal decisions that the resulting set cannot be improved further by the inclusion of new features.

We can reduce this effect somewhat by adding *delete* and *replace* operations to the algorithm that are used to check whether the deletion or replacement of features in the current set improves the performance further (many classifiers perform worse in the presence of "distracting" and redundant features).

4.2.2. Combining random feature sets and a greedy algorithm. We ran the greedy algorithm from scratch (the empty set) to produce a 15-feature set that achieved results equivalent to about 31 randomly selected features. Since the performance increases due to the greedy algorithm are significant during the first few steps of its execution we also started it from a good random collection of features.

The performance of a set of features varied strongly with the initialization of the search. Therefore, we averaged over the performance of differently initialized searches. We generated 30 random sets containing six features. For each of the 30 sets we averaged the performance of the classifier over 10 runs. The set that achieved the best average classification results was taken as a starting point of an incremental greedy algorithm. As mentioned earlier we also checked at each step whether deleting a feature or replacing a feature was helpful.

Using this strategy we determined 13 features that achieve a total error of 28,373 on the training set, which is only about 3% worse than that found in the collection of all 51 features combined (27,500). The error figures are the number of

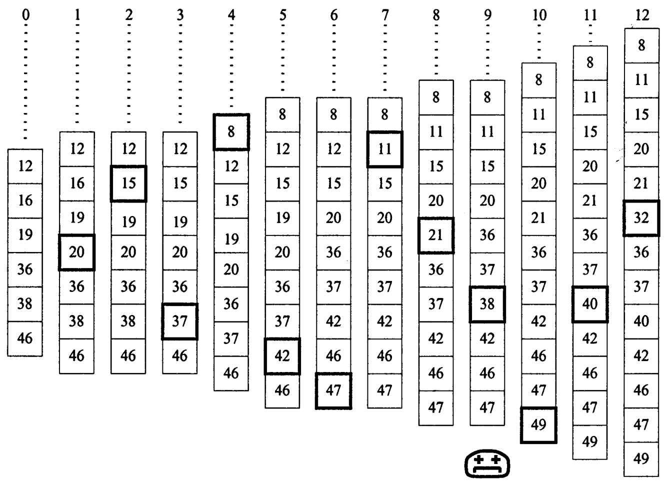

FIG. 8. Stages of the greedy algorithm.

misclassified pixels in the entire training set of over half a million pixels. The execution time of the feature extraction stage, on the other hand, is reduced from about 1 day to under 5 h for an image of size 512 × 512 pixels.

Figure 8 illustrates the steps of the method for a particular set of features. The initial six-feature set is shown in column 0 of feature IDs (the features are numbered 1 through 51). In the first step it was found that feature 20, which was not in the six-feature set, reduced the error the most among all features not in the six-feature set. The resulting seven-feature set is shown in column 1. In the next two steps, features 16 and 38 were replaced by features 15 and 37, respectively. In step 4 feature 8 was found to be the best addition to the existing set. In steps 6 and 7, features 19 and 12 were replaced by features 47 and 11, respectively. The feature sets in steps 8 and 9 both achieved roughly equally good classification. Therefore, both were kept as possible expansions of the previous feature set. At the next step (10), however, it turned out that reintroducing feature 38, which was replaced with feature 37 in step 3, was not as beneficial as adding feature 21 to the set. The dead Tamagotchi marks the elimination of an alternative feature set. Finally, in steps 11 and 12, features 40 and 32 were added to the set of features.

The first five of these features are based on the graylevel co-occurrence matrix, the 6th is a Gabor filter measure; the 7th is a multifractal measure; the 8th, 11th, 12th, and 13th are based on color measurements; the 9th is based on entropy; and the 10th, is based on the Fourier transform. With the exception of the steerable filter-based features all types of measures are represented. This confirms the need for the combination of features of different types for good classification.

The main reason for the inclusion of the steerable feature measures was to facilitate the detection of branches and forks in images in the future (not used in this work). But we also used the orientation and edge type information to see if they could contribute directly to the classification process, rather than just indirectly through the detection of branches and forks. The absence of these features from the final 13-feature set indicates that without further processing they are not very useful.

Efficient subsets of features, like the one used for this study, generally contain representatives of most types of feature extraction methods. We found that combinations of features of different methods clearly outperform features based on only one feature extraction method (e.g., using only Gabor filters).

Figure 9 compares the performance of random feature sets from Fig. 7 and the performance of the feature sets shown in Fig. 8 (the circles in the graph).

From Fig. 9 it can be seen that the performance of these 13 features is roughly equivalent to the performance of the entire feature set. As we mentioned earlier, it also can be seen that the performance increases taper off as the number of steps of the greedy algorithm increases.

4.3. *The Performance of the Resulting Feature Set*

Figures 10 and 11 show previously unseen images with their classification results horizontally adjacent to them. Brighter regions in the labeled images represent areas that are likely to depict deciduous trees and darker regions represent areas that are less likely to depict deciduous trees. Due to the filter width of the texture measures we do not generate labels around the edges of the images where the filters have incomplete image support.

Measures of every seventh pixel of 25 training images were obtained and combined with labeled images to train the network. Note that from these 25 images we obtained well over half a million data points. The 51D data set is about 119 MB; the 13D data set is about 30 MB. Subsampling speeds up the training process without (noticeably) affecting its outcome, since neighboring pixel locations are highly correlated.

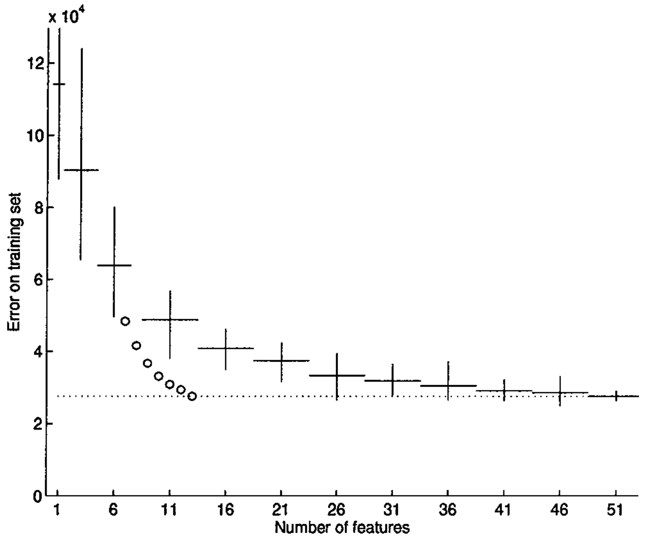

FIG. 9. The circles in the graph show the average performance of seven feature sets obtained starting from the best six-feature set (the lower tip of the vertical line corresponding to the performance range of random feature sets of size 6).

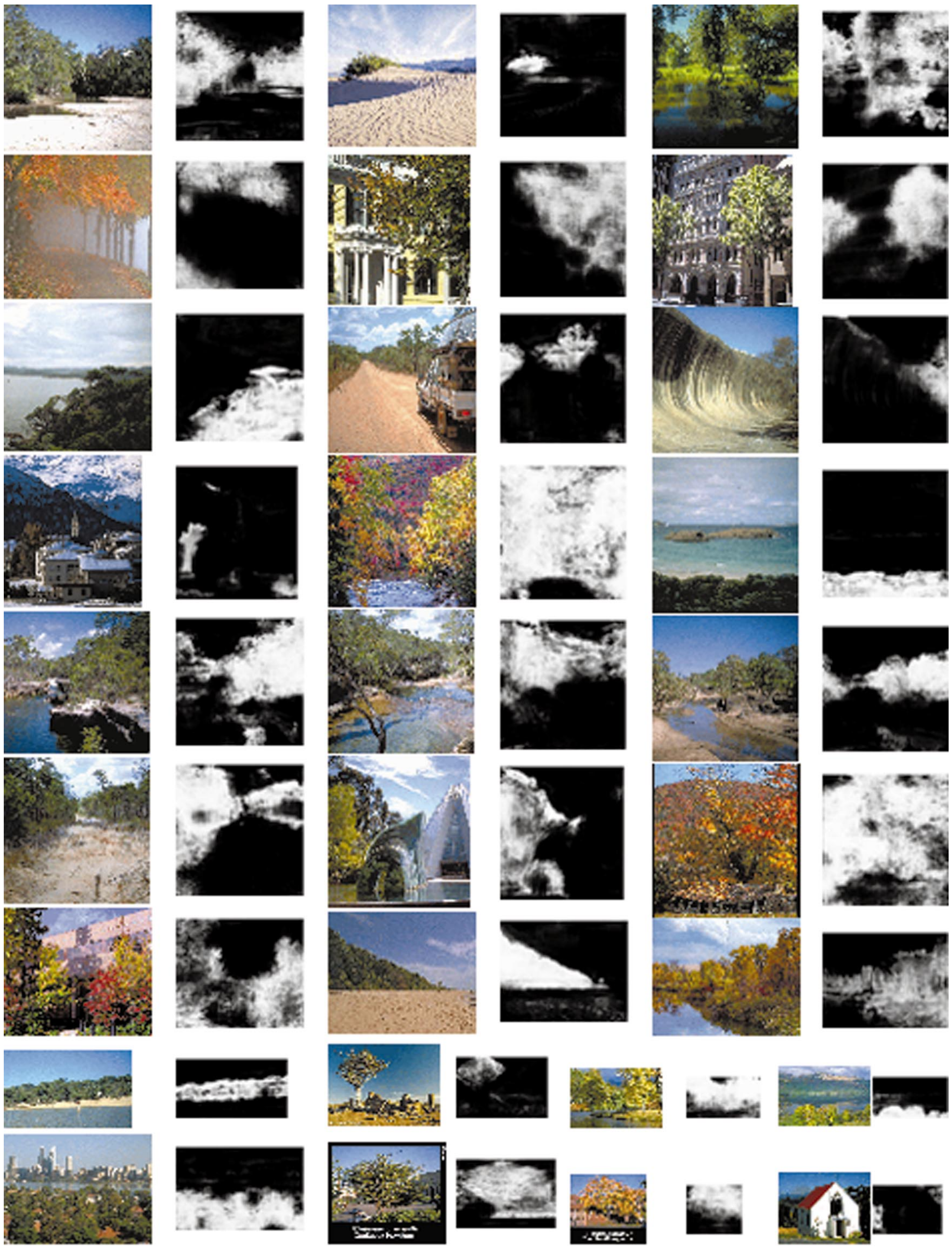
FIG. 10. Test images and the corresponding classification results. The following images are from Compact Discs of Corel Professional Photos, copyright Corel Corporation, and used under license: row 1 right and row 4 left. The following image is used with the permission of the Corporation for National Research Initiatives (http://www.cnri.reston.va.us/), with whose permission it is used: row 7 left. The following images are used with the permission of John Frett (http://bluehen.ags.udel.edu/homepage/plsc/plscstaff/frett.html) and Betsy Mackenzie (http://bluehen.ags.udel.edu/betsy) and the University of Delaware Botanic Gardens: row 9 mid left and mid right. The following images are courtesy Philip Greenspun (http://photo.net/philg): row 2 left and center, row 4 center, row 6 right. The following images are copyright Gerhard Ortner, with whose permission they are used: row 1 left and center, row 2 right, row 3 all, row 4 right, row 5 all, row 6 left and center, row 7 center and right, row 8 all, row 9 extreme left and extreme right.

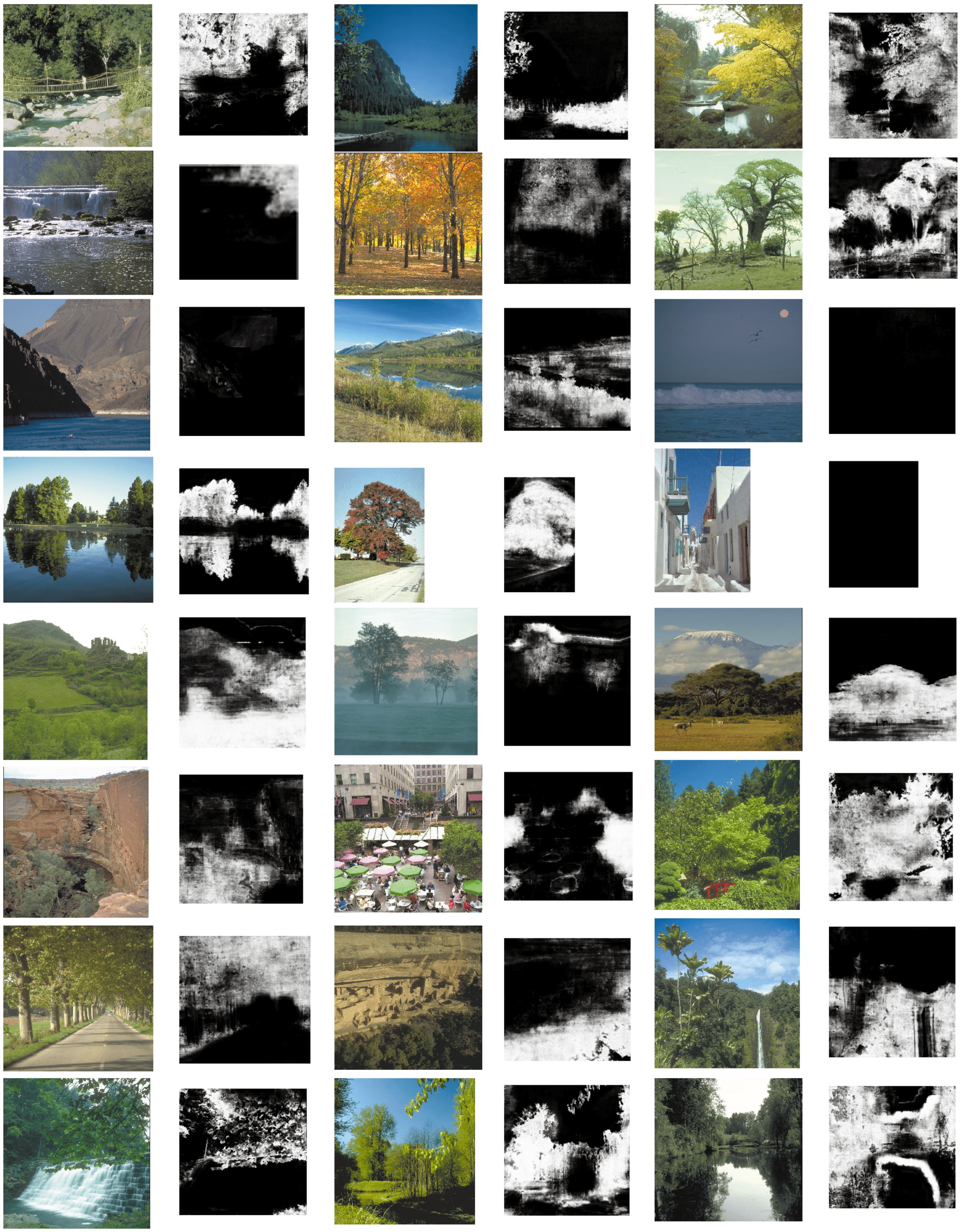

FIG. 11. Test images and the corresponding classification results. The following images are from Compact Discs of Corel Professional Photos, copyright Corel Corporation, and used under license: row 1 all, row 2 all, row 3 all, row 4 left, row 5 all, row 6 right, row 7 all, and row 8 all. The following image is used with the permission of John Frett (http://bluehen.ags.udel.edu/homepage/plsc/plscstaff/frett.html) and Betsy Mackenzie (http://bluehen.ags.udel.edu/betsy) and the University of Delaware Botanic Gardens: row 4 center. The following image is courtesy Philip Greenspun (http://photo.net/philg): row 6 center. The following images are copyright Gerhard Ortner, with whose permission they are used: row 4 right, row 6 left.

We would like to point out that some of the test images show trees in fall with the leaves' colors ranging from green, through yellow, orange, and red, to a magenta-ish red. Thus the color-based features illustrate a typical problem the classifier has to solve; color is often a useful cue, but leaves are not always green and not everything green depicts leaves.

The first image pair on the second row in Fig. 10 shows the performance of the approach for an image taken on a foggy day, with low contrast and low color saturation. The second image pair on row four in the same figure shows the approach's robustness with respect to scale and color. This fall image shows trees at distances ranging between 5 m and over 500 m whose colors range from magenta to green. The output image shows that almost all regions were correctly labeled. Not all deciduous trees are green and not everything green is a deciduous tree; in fact deciduous trees take most hues except those from a thin band around blue. The same is true for the other measures we use; they all perform poorly in isolation. The classifier combines these measures to improve on the performance of the individual measures.

4.4. *Neural Network Issues*

While varying the number and kind of features has a strong impact on the performance of the classifier, the architecture of the back-propagation neural network affects the performance very little. The network we use for all our work has a single hidden layer with 25 hidden units in it. This is a safe architecture. Generally we found that adding more hidden layers or more hidden units in the hidden layer slows the convergence while producing little performance improvement. Twenty-five hidden units in a single hidden layer produced good results for feature sets (input units) from the smallest (six-feature) set up to the full (51-feature) set.

5. ALTERNATIVE METHODS FOR CLASSIFICATION AND FEATURE RELEVANCE ESTIMATION

Here, we discuss alternative classifiers and alternative approaches to the determination of the most relevant features. Since the preprocessing stage is the slowest of the modules, we consider classification without preprocessing in the first part of this section.

Since the "relevance" of a feature depends on the method used for classification, there is little benefit in using one approach to decide on the importance of features and another approach to classify images given these features. Therefore, we consider linear, quadratic, and eigenanalysis techniques to both the determination of good subsets of features and classification. We show how other approaches to feature selection compare to the greedy approach presented in Section 4.

5.1. *Classification without Preprocessing*

We considered classification without preprocessing and trained a convolutional neural network (CNN) [20] on raw image data. Unfortunately the results were discouraging. The training took significantly longer than for the back-propagation network with separate preprocessing and labeling stages, and the performance on both the training and the test set was worse (see Figs. 14 and 16). One of the motivating factors for the use of CNNs is that they provide a degree of translation invariance by sharing weights among input units; at the same time this reduces the number of free parameters (weights) and hence speeds convergence. In [20] a CNN is used for handwritten digit recognition. In order to reduce the number of weights in the network all units at the same level use a common set of weights (a convolution kernel) to connect them to the units of the next higher level of the network. Our comparison shows that this is not enough to obtain good classification results. Training a classifier on preprocessed input therefore seems to be the better approach.

5.2. *Linear Relationships between Pairs of Variables*

Covariance and correlation matrices only measure *linear* relationships between *pairs* of variables. Therefore, methods based on them do not accurately capture (i) nonlinear relationships between pairs of variables (see Fig. 12), and (ii) linear or nonlinear relationships between more than two variables at the same time.

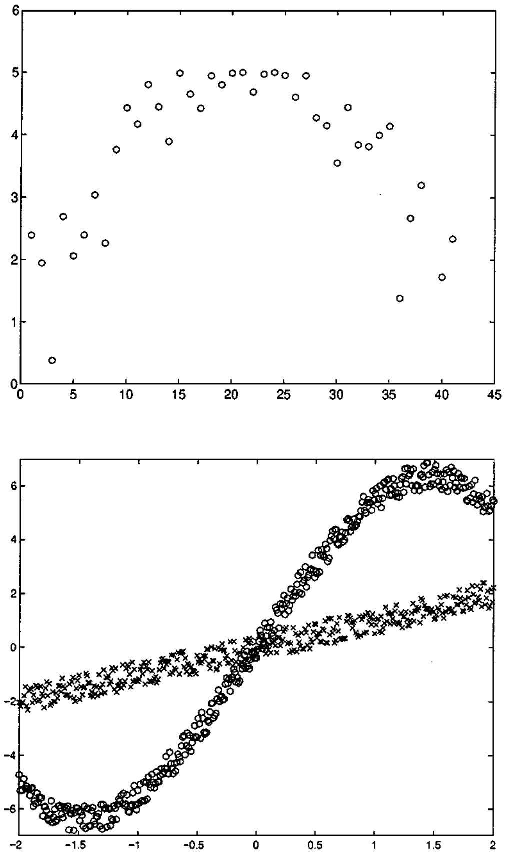

FIG. 12. The upper plot shows two variables on the x- and y-axes that have zero covariance and yet are dependent. The lower plot shows how the same linear relationship between the variables on the x- and y-axes might represent very different nonlinear properties between the variables.

Since the values of covariance and correlation matrices do not even reflect the true relation between pairs of variables, classification methods that are based on these matrices are fundamentally flawed and we should not be surprised to see that their performance is inferior to that of the suggested nonlinear classifier (see Fig. 14). For example, a cubic relationship between two variables might make them linearly dependent even though the nonlinear relationship might contain valuable and exploitable information for the classification problem at hand.

5.3. Linear Analysis

If we restrict ourselves to linear classification methods, Fisher's Linear Discriminant Functions (LDF) [9], Maximally Discriminating Functions [24], or the Best Linear Discriminant Function [35] can be used. To determine the redundancy and contribution of features we can use the F-test or the Wilks test [34] or the discriminant functions themselves. Fisher's Linear Discriminant Function is widely used to classify an observation into one of two classes. An observation is taken to belong to class 1 if

$$(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)^{\mathrm{T}} \mathbf{S}_{\mathrm{pl}}^{-1} \left(\mathbf{x} - \frac{1}{2}(\bar{\mathbf{x}}_1 + \bar{\mathbf{x}}_2) \right) > \ln\left(\frac{\hat{\pi}_2}{\hat{\pi}_1} \right)$$

and to belong to class 2 otherwise. Here $\bar{\mathbf{x}}_i$ denotes the mean of class i, $\mathbf{S}$ is the pooled sample covariance matrix, $\mathbf{x}$ is the observation to be classified, and $\hat{\pi}_i$ reflects the prior information we have about the likelihood of the observation belonging to class i.

Linear classifiers are inferior in power to nonlinear classifiers, and both the linear classifiers and the tests for redundancy require the calculation of the inverse of the class or pooled covariance matrices. Since our set of features is very *linearly* redundant, the associated covariance matrices are too singular to enable the stable computation of their inverse.

5.4. Quadratic Analysis

According to the quadratic discriminant function an observation is taken to belong to class 1 if

$$\mathbf{x}^{\mathrm{T}}\big(\mathbf{S}_2^{-1} - \mathbf{S}_1^{-1}\big)\mathbf{x} - 2\mathbf{x}\big(\mathbf{S}_2^{-1}\bar{\mathbf{x}}_2 - \mathbf{S}_1^{-1}\bar{\mathbf{x}}_1\big) + \big(\bar{\mathbf{x}}_2^{\mathrm{T}}\mathbf{S}_2^{-1}\bar{\mathbf{x}}_2 - \bar{\mathbf{x}}_1\mathbf{S}_1^{-1}\bar{\mathbf{x}}_1\big)$$
$$> \ln\left(\frac{|\mathbf{S}_2|}{|\mathbf{S}_1|}\right) + 2\ln\left(\frac{\hat{\pi}_2}{\hat{\pi}_1}\right)$$

and to belong to class 2 otherwise. As before $\bar{\mathbf{x}}_i$ denotes the mean of class i, $\mathbf{S}_1$ and $\mathbf{S}_2$ are the pooled sample covariance matrices, $\mathbf{x}$ is the observation to be classified, and π_i reflects the prior information we have about the likelihood of the observation belonging to class i.

While quadratic discriminant functions [36] generally yield better results than linear discriminant functions, they too require the calculation of the inverses of the near-singular class covariance matrices. Therefore, they too are computationally too unstable to allow their use for a feature redundancy analysis or classification.

5.5. Eigenanalyses

A global or class eigenanalysis of the covariance matrices is useful to describe the features but it is not designed to classify feature vectors into their corresponding classes and hence, often performs poorly as a basis for classification. Recent successful applications of eigenanalyses for classification, e.g., [29], normalize images of training and test objects prior to the classification to maximize the likelihood that differences in the description are indeed due to differences in the identity of the objects. In our case it is unclear how to standardize/normalize the appearance of an arbitrarily shaped tree of unknown size in arbitrary images, especially if it has not already been segmented or recognized.

In classification, therefore, we are mainly interested in finding the features that maximize the *differences* between the individuals of the different classes. For this purpose the most discriminating features can be obtained by maximizing the ratio of the *between*- and *within*-class sums of squares and products matrices (details in [2, 15, 23, 34]). Unfortunately, this method degenerates to Fisher's Linear Discriminant Function in the two class case (e.g., trees and non-trees, as in our case). Therefore, the problems of Fisher's LDF with the inverse of near-singular matrices are inherited by the eigenanalysis of the class differences.

5.6. Minimally Correlated Features

From the previous sections it is obvious that the discussed alternatives all suffer the same problem when (as in our case) the correlation/covariance matrices are nearly singular. Therefore, we asked the question: How powerful are features that allow the stable calculation of the inverse of the data set's correlation matrix?

The answer is "Not very." Features for which linear classifiers can be used perform roughly as well as randomly selected feature sets of the same size.

We estimated the condition number of the correlation matrices using the 1-norm LINPACK condition estimator. This measure is an indication of the relative distance from a given correlation matrix to the set of singular matrices. The condition number estimate is a defensive approximation; i.e., the actual condition number is likely to be at least as bad.

Figure 13 contrasts the performance (error on the training set) of the feature sets that have the least singular covariance matrix and the average performance of random feature sets of sizes 1, 3, 6, and 11. The numbers in the right graph are the estimates of the covariance matrices' condition. Estimates near one indicate sets of (linearly) uncorrelated variables, while larger estimates approach the set of singular covariance matrices. The rapid increase in the singularity measure indicates that combinations of 20 features or more will cause even the set of features with the least singular covariance matrix to become too singular to allow the stable computation of its inverse.

From the left graph, we can see that features for which linear classifiers can be used perform slightly better than randomly selected feature sets of the same size. Increasing the feature set

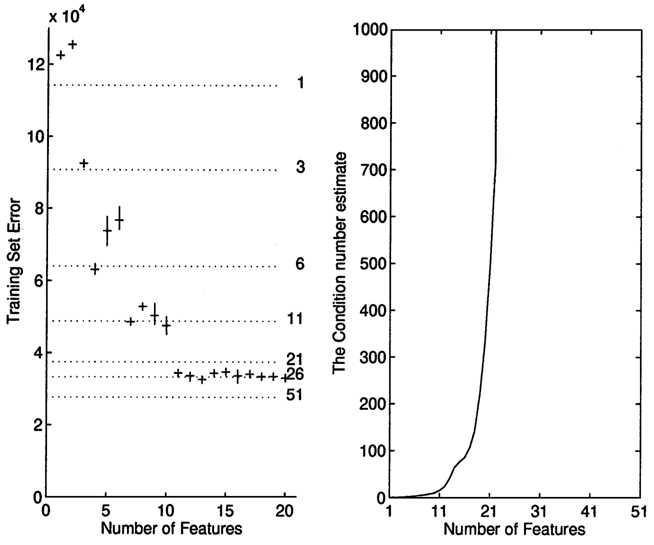

FIG. 13. The performance of feature sets most suitable for linear analysis (left) and the corresponding estimates of their covariances' singularity measure (right).

size to more than 10 features yields no further performance increase. Note that the peak performance of sets selected this way is about 20% worse than that of the nonlinear classifier (the minimum average error on the training set is 33,798). Dotted lines show the performance of random feature sets of the indicated size. The solid horizontal lines show the average performance of the feature sets corresponding to the least singular covariance matrices; the solid vertical lines show the performance variation due to initialization.

Fortunately we can circumvent the problems with singular covariance and correlation matrices by using methods that do not require their use (such as a back-propagation neural network).

The 13-feature set that produced the results in Figs. 10 and 11 have a condition number estimate of 1.9307 × 10.5, indicating that the 13-feature set is highly linearly redundant. Since the set achieves good classification and deleting any of the features from it deteriorates the results noticeably, we see that linear redundancy is not useful for deciding on the importance of features for a nonlinear classifier.

5.7. A Comparison of Classification Methods

Figure 14 shows the performance differences between linear, quadratic, and nonlinear methods for a sample training image. The results for the linear and quadratic classifiers in Fig. 14 are based on the 20-feature set determined in the redundancy analysis described earlier. While other sets might theoretically have better discriminatory powers, the inverses of their covariance matrices cannot be computed stably and their performance is thus unreliable (and likely to be worse).

The results of the linear and quadratic classifiers in Fig. 14 are typical for these classifiers and reflect their discrimininatory power. The convolutional neural network achieves classification results on this training image that are equivalent to those achieved by the back-propagation neural network.

When comparing the performance of both networks on another training image, as in Fig. 15, and a previously unseen (test) image, as in Fig. 16, we can see that the performance of the convolutional neural network is inferior to that of the back-propagation neural network.

Since the performance of the linear and quadratic classifiers is significantly worse than that of the nonlinear classifiers, we did not include their classification results in these figures.

5.8. Each Method in Isolation

As indicated in the Introduction the 51 features are obtained using seven different methods. Since any single feature on its own has very little discriminatory power we compare each of the underlying seven methods in turn. Figure 17 plots the classification error on training images on the y-axis for each of the seven underlying feature extraction methods in isolation on the x-axis. The numbers in parentheses indicate the number of features derived from each feature extraction method. We can see that the methods in isolation are all inferior to feature sets of the same size drawn at random from the entire feature set (dotted

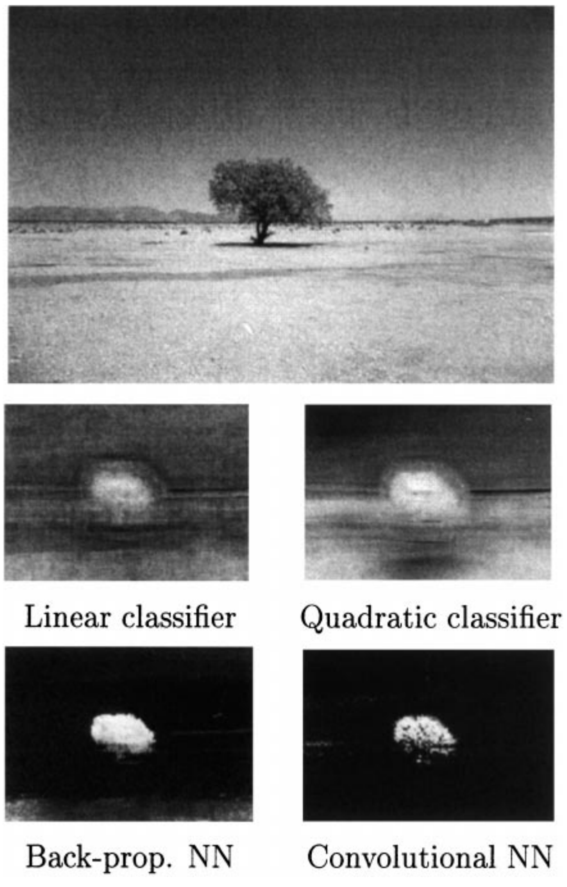

FIG. 14. A sample training image and the segmentation results obtained using various classifiers. The image is copyrigt Gerhard Ortner, with whose permission it is used.

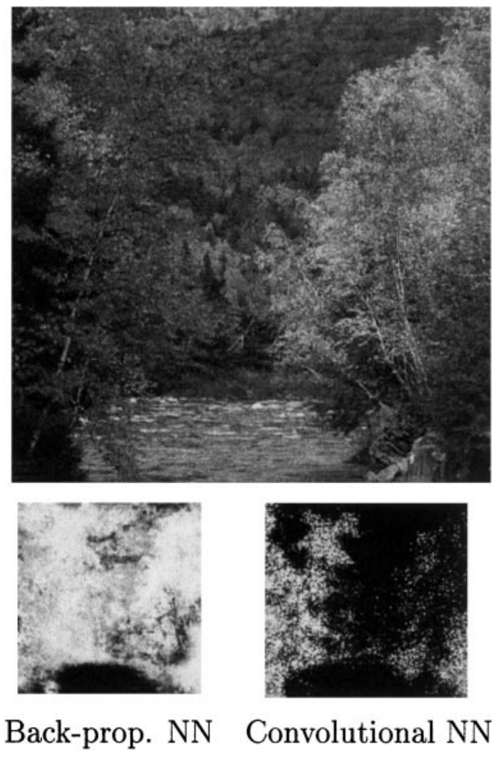

FIG. 15. Another training image and the corresponding segmentation results. Courtesy Philip Greenspun (http://photo.net/philg).

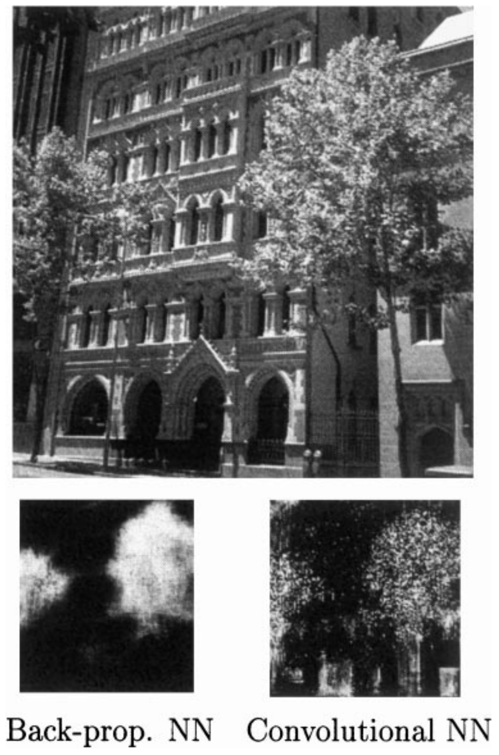

FIG. 16. A test image and the corresponding segmentation results. The image is copyright Gerhard Ortner, with whose permission it is used.

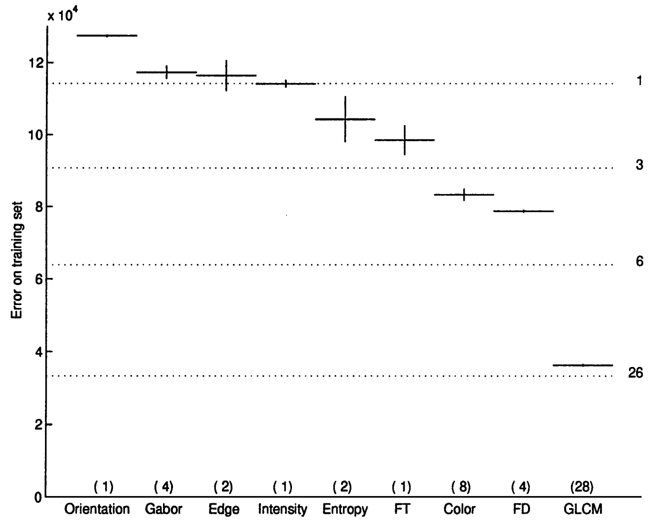

FIG. 17. Performance of each method is isolation.

lines). The performance of the different methods is roughly proportional to the number of features derived from them, giving rise to the apparent strength of the 28 GLCM-based feature set. Comparing its performance with that of random feature sets of similar size we can see that the 28 GLCM features are inferior to a random collection of 26 features. Therefore, it is obvious that this approach does not allow us to determine good feature subsets and to optimize our classification performance. Since we are interested in the best classification given all features we cannot judge the importance of features and methods by measuring their power in isolation.

5.9. *Leaving One Out*

It would therefore make more sense to consider a large feature set and to measure how much the performance of the entire set of features deteriorates as various features are left out. Since the presence or absence of a single feature is virtually unnoticeable, we measure the change of the performance if all features derived from one method are omitted (the horizontal axis of Fig. 18 shows the omitted method). Although from Fig. 18 it seems that the omission of the GLCM-based features causes the largest deterioration in the error rate, this effect is almost entirely due to the large number of features based on that method. The performance of random sets shows that the GLCM features are neither particularly useful nor particularly redundant. Another observation that can be made from this graph is that the elimination of the features of some types of measures improves on the performance of the entire set of features. While this indicates *that* the complete set is redundant it does not show *which* features to omit in order to maximize the performance.

6. CONCLUSIONS

We represent images using 51 measures from seven fundamental feature extraction methods based on color, the graylevel

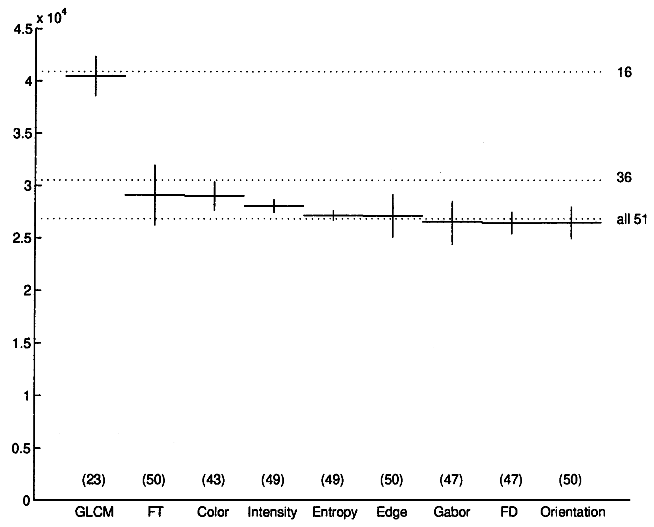

FIG. 18. Performance of features from the Leave-One-Out method.

co-occurrence matrix, Gabor filters, fractal dimension, steerable filters, the Fourier transform, and entropy.

We present a method that extracts good subsets of features from the set of 51 features. The resulting subset is almost as powerful as the entire set but reduces the time for the feature extraction phase by 75%. The fact that representatives from all but the steerable filter method are present in this subset lets us conclude that the mixture of methods and features provides for better classification than the use of features derived from just one method (as was shown in Fig. 17). The 13-feature solution we have found outperforms all the methods in isolation (even the combined set of 28 GLCM-based features).

6.1. *Linear Dependence*

Covariance- (or correlation-) matrix-based feature relevance analyses fail even if only a small number of features are significantly linearly correlated. Any attempt to eliminate the most linearly correlated features is likely to discard many important features and curtail the power of the remaining feature set. From our experiments we found that feature sets whose covariance matrices are sufficiently independent to allow their inverses to be obtained only perform as well as random feature sets of the same size. Therefore, we conclude that the contribution of linearly dependent features within nonlinear classifiers can be significant, as was shown in Section 4.

6.2. *Assumptions of Linear and Quadratic Classifiers*

Many of the available classifiers also make implicit assumptions that the different classes/groups have similar covariance matrices or are normally distributed. If this assumption is violated many of the derived results become invalid. Section 5.7 showed that classical linear methods are inferior to nonlinear methods, such as a back-propagation neural network.

6.3. *Random Feature Sets*

For k features we have 2^k possible subsets of features and transitivity cannot be used to build optimal sets of size $N + 1$ from optimal sets of size N (likewise we cannot eliminate the least important feature to obtain sets of size $N - 1$ from sets of size N). However, we have shown how to find good feature sets by using a greedy strategy on good random collections of features (see Section 4.2.2).

The best 13-feature set, found using a greedy strategy, spans all but one feature extraction method and outperforms all the methods in isolation (even the combined set of 28 GLCM-based features). Generally, random collections of features from different methods out perform equally sized feature sets of one type.

6.4. *Back-Propagation Neural Networks*

The use of a back-propagation neural network offers a simple solution to the laborious task of finding a good combination of the available features, thus solving our senso fusion problem. We have shown that feature sets like the one presented have sufficient expressive power to allow good generalization from only a few training images. An analytical approach, on the other hand, is difficult to conduct since the interactions even between modest numbers of dependent features is complex.

Natural scenes and objects often have signature colors and textures, which is why we expect the presented approach to be most useful in this domain. Humanmade structures and objects often do not exhibit the same degree of color and texture constancy between instances and our approach hence appears less well suited to them.

ACKNOWLEDGMENTS

We thank Guy Smith and Ian Burns from the University of Queensland for their texture analysis package MEASTEX.

REFERENCES

1. T. D. Alter and D. W. Jacobs, Error propagation in full 3D—from 2D object recognition, in *Proceedings of Computer Vision and Pattern Recognition 1994*, pp. 892–899.
2. S. T. Bow, *Pattern Recognition and Image Preprocessing*, Dekker, New York, 1992.
3. S. Belongie, C. Carson, H. Greenspan, and J. Malik, Color- and texture-based image segmentation using EM and its application to content-based image retrieval, in *IEEE Workshop on Content Based Access of Image and Video Databases, Bombay, India, 1998*. [In conjunction with ICCV'98]
4. J. F. Canny, *Finding Edges and Lines in Images*, Masters thesis, MIT Press, Cambridge, MA, 1983.
5. R. W. Conners and C. A. Harlow, A theoretical comparison of texture algorithms, *IEEE Trans. Pattern Anal. Mach. Intelligence* **2**, 1980, 204–222.
6. B. B. Chaudhuri, N. Sarkar, and P. Kundu, Improved fractal geometry based texture segmentation technique, *IEE Proc.* **140**, 1993, 233–241.
7. G. Cybenko, Approximation by superposition of sigmoidal function, in *Mathematics of Control, Signals, and Systems*, Chap. 2, pp. 303–314, 1989.

8. S. Fahlman, Faster-learning variations on back-propagation: An empirical study, in *Proceedings of 1998 Connectionist Models Summer School*, Morgan Kaufmann, San Mateo, CA.
9. R. A. Fisher, The use of multiple measurements in taxonomic problems, *Ann. Eugen.* **7** Part 2, pp. 179–188, 1936.
10. M. A. Fischler, Robotic vision: Sketching natural scenes, in *Proceedings of 1996 APRA IU Workshop*.
11. I. Fogel and D. Sagi, Gabor filters as texture discriminator, *J. Bio. Cybernet.* **61**, 1989, 103–113.
12. W. T. Freeman and E. H. Adelson, The design and use of steerable filters, *IEEE Trans. Pattern Anal. Mach. Intelligence* **13**, 1991, pp. 891–906.
13. D. Gabor, Theory of communication, *Proc. Inst. Electr. Eng.* **93**(26), 1946, 429–441.
14. R. Gonzalez, *Digital Image Processing*, Addison–Wesley, Reading, MA, 1986.
15. D. J. Hand, *Construction and Assessment of Classification Rules*, Wiley, New York, 1997.
16. R. M. Haralick, K. Shanmugam, and I. Dinstein, Textural features for image classification, *IEEE Trans. Systems Man Cybernet.* **3**(6), 1973, 610–621.
17. M. S. Lew, K. Lempinen, and N. Huijsmans, Webcrawling using sketches, in *Proceedings of Visual97*, 1997.
18. D. W. Jacobs, *Recognizing 3-D Objects Using 2-D Images*, Ph.D. thesis, MIT, Cambridge, MA, 1992.
19. R. Jain, R. Kasturi, and B. Schunck, *Machine Vision*, pp. 278–284, McGraw–Hill, New York, 1995.
20. Y. Bengio, Y. LeCun, and D. Henderson, Globally trained handwritten Word recognizer using spatial representation, convolutional neural networks and hidden Markov models, in *Neural Networks*, 1994.
21. J. M. Keller and S. Chen, Texture description and segmentation through fractal geometry, *Comput. Vision Graphics Image Process.* **45**, 1989, 150–166.
22. B. S. Manjunath and W. Y. Ma, Texture features for browsing and retrieval of image data, *IEEE Trans. Pattern Anal. Mech. Intelligence* **18**(8), 1996, 837–859.
23. B. F. J. Manly, *Multivariate Statistical Methods*, Chapman & Hall, London/New York, 1994.
24. S. Marks and O. J. Dunn, Discriminant functions when covariance matrices are unequal, *J. Am. Stat. Assoc.* **69**, 1974.
25. G. Smith and I. Burns, *MeasTex*, available at http://www.cssip.elec.uq.edu.au/˜guy/meastex/meastex.html.
26. J. R. Miller, J. R. Freemantle, M. J. Belanger, C. D. Elvidge, and M. G. Boyer, Potential for determination of leaf chlorophyll content using AVIRIS, in *Proceedings of the Second Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) Workshop, Pasadena, 1990*, pp. 72–77.
27. Y. Awaya, J. R. Miller, and J. R. Freemantle, Background effects on reflectance and derivatives in an open-canopy forest using airborne imaging spectrometer data, in *Proceedings of the XVII Congress of ISPRS, Washington, D.C., 1992*, pp. 836–843.
28. A. P. Pentland, Fractal-based description of natural scenes, *IEEE Trans. Pattern Anal. Mach. Intelligence* **6**, 1984, 661–672.
29. M. A. Turk and A. P. Pentland, Face recognition using eigenfaces, in *Proceedings of Computer Vision and Pattern Recognition, 1991*, pp. 586–591.
30. S. Peleg, J. Naor, R. Hartley, and D. Avnir, Multiple resolution texture analysis and classification, *IEEE Trans. Pattern Anal. Mach. Intelligence* **6**, 1984, 518–523.
31. P. Perona, Deformable kernels for early vision, *IEEE Trans. Pattern Anal. Mach. Intelligence.* **17**, 1995, pp. 488– 499.
32. B. Vijayakumar, D. J. Kriegman, and J. Ponce, Invariant-based recognition of complex curved 3D objects from image contours, in *International Conference on Computer Vision 1995*, pp. 508–514.
33. M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker, Query by image and video content: The QBIC system, *IEEE Comput.* **28**(9), 1995, pp. 23–32.
34. A. C. Rencher, *Methods of Multivariate Analysis*, Wiley, New York, 1996.
35. R. H. Riffenburgh and C. W. Clunies-Ross, Linear discriminant analysis, *Pacific Sci.* **14**, 1960, 251–256.
36. D. F. Specht, Generation of polynomial discriminant functions for pattern recognition, *IEEE Trans. Electron. Comput.* **EC-16**(3), 1967, pp. 308–319.
37. M. Szummer and R. W. Picard, Indoor–outdoor image classificatoin, in *IEEE Workshop on Content Based Access of Image and Video Databases, Bombay, India, 1998*. [In conjunction with ICCV'98; available at http://www-white.media.mit.edu/people/szummer/profile.html]
38. A. Vailaya, A. Jain, and H. J. Zhang, On image classification: City images vs. landscapes, in *Workshop on Content Based Access of Image and Video Libraries, June, 1998.*
39. J. S. Weszka, C. R. Dyer, and A. Rosenfeld, A comparative study of texture measures for terrain classification, *IEEE Trans. Systems Man Cybernet.* **6**, 1976, 269–285.