

Review of the State of the Art in Semantic Scene Classification

Matthew Boutell
Christopher Brown
Department of Computer Science
University of Rochester
Rochester, NY

Jiebo Luo
Imaging Science and Technology Lab
Eastman Kodak Company
Rochester, NY

December 17, 2002

Abstract

Semantic scene classification, categorizing images into one of a set of *physical* (e.g., indoor/outdoor, orientation) or *semantic* categories (e.g., beach or party), is a relatively new field. Most of the existing techniques used primarily low-level features to classify scenes and achieved some success on constrained problems. We report on the state of the art, presenting summaries of major scene classification systems and identifying the features and inference engines they use.

Keywords: semantic scene classification, image classification, low-level features, semantic features, exemplar-based systems, model-based systems.

Contents

1	Introduction	1
1.1	Applications of Scene Classification	3
2	Core Concepts in Scene Classification	4
2.1	Features	4
2.1.1	Color	5
2.1.2	Texture	5
2.1.3	Filter output	6
2.1.4	Edges	6
2.1.5	Context Patches	6
2.1.6	Object Shape	6
2.1.7	Semantic Component and Object Classification	6
2.1.8	Other Image Understanding Systems' Output	7
2.1.9	Context and Meta-data	7
2.1.10	Statistical Measures	7
2.1.11	Spatial Layouts	7
2.2	Scene Configuration Modeling	8
2.3	Learning and Inference Engines	8
2.3.1	K -Nearest Neighbor	9
2.3.2	Learning Vector Quantization (LVQ)	9
2.3.3	Bayesian Classifier	10
2.3.4	Support Vector Machines (SVM)	11
2.3.5	Artificial Neural Networks	12
2.3.6	K -Means	12
2.3.7	Expectation Maximization (EM)	12

2.3.8	Bayesian Networks	13
2.3.9	Configuration-based Engines	13
2.4	Control Strategies	14
2.4.1	Top-Down Control	14
2.4.2	Bottom-up Control	14
2.4.3	Hybrid Control	15
2.4.4	Cost-based Control	15
3	Scene Classification Systems	16
3.1	Exemplar-based Systems	16
3.2	Exemplar-based Systems Using Low-level Features	17
3.2.1	Automatic Orientation Detection	17
3.2.2	Indoor-Outdoor Classification	17
3.2.3	Hierarchical Image Classification	18
3.2.4	Holistic Classification Using the "Spatial Envelope"	19
3.2.5	Scenes Dominated by Single Objects	20
3.3	Exemplar-based Systems Using Segmentation	21
3.3.1	Semantics-Sensitive Feature Extraction	21
3.3.2	Blobworld	22
3.4	Model-based Systems	22
3.4.1	Configuration-Based Classifiers	22
3.4.2	Composite Region Templates	23
3.5	Contextual Information	24
3.5.1	Object Priming	25
3.5.2	Focus of Attention	25
4	Discussion and Conclusions	27
4.1	Shortcomings of Low-level Feature-Based Systems Reviewed	27
4.2	Limitations of Low-level Features	28
4.3	Scene Configuration	29
4.4	Conclusion	30
5	Acknowledgments	31

Chapter 1

Introduction

“Image understanding is one of the most complex challenges of AI...” [49].

As usually defined, *image understanding* is the process of converting “pixels to predicates”: (iconic) image representations to another (symbolic) form of knowledge [2]. Image understanding is the highest (most abstract) processing level in computer vision [49]¹. Lower-level image processing techniques such as segmentation are used to create regions that can then be identified as objects. The control strategies used to order the various processing steps required can vary [3]. *Bottom-up* strategies start with the image, perform general purpose segmentation, and then attempt to recognize each object. *Top-down* strategies involve a hypothesize-verify control loop and specialized segmentation. *Hybrid* strategies attempt to take the best of both worlds. In any case, the end result desired is for the vision to support high-level reasoning about the objects and their relationships to meet a goal.

While image understanding in unconstrained environments is still very much an open problem [49, 58], much progress is currently being made in *scene classification*. Here the goal is not as ambitious; one simply wants to place an image automatically into one of a set of *physical* (e.g., indoor/outdoor, orientation) or *semantic* categories (e.g., beach or party). For instance, if a person recognizes trees at the top of a photo, grass on the bottom, and people in the middle, he may hypothesize that he is looking at a park scene, even if he cannot see every detail in the image. Or on a different level, if there are lots of sharp vertical and horizontal edges, he may be looking at an urban scene.

Often, scenes can be classified without full knowledge of every object in the image. It may be possible in some cases to use low-level information, such as (spatial distribution of) color or texture, to classify some scene types accurately. In other cases, perhaps object recognition is necessary, but not necessarily of every object in the scene. Classification seems to be an easier problem than unconstrained image understanding; early results have confirmed this for certain scene types in constrained environments [54, 58].

The description of images can vary in the level in semantic content. Wang *et al.* give four

¹As opposed to *image processing* techniques, which convert one image representation to another. For instance, using a mask to convert raw pixels to an edge image is much more concrete.

levels of semantics [61]:

1. semantic types (e.g. landscape photograph, clip art)
2. object composition (e.g. a bike and a car parked on a beach, a sunset)
3. abstract semantics (e.g. people fighting, a happy person, an objectionable photograph)
4. detailed semantics (e.g. a detailed description of a given picture)

In this review, we constrain ourselves to the first two levels, considering the bottom two levels as scene *understanding* as defined above ². Good progress has been made at the first level, which corresponds closely to image’s *physical* attributes. Examples of this broad categorization include indoor vs. outdoor [22, 44, 52, 58], image orientation [4, 60, 62], and textured vs. non-textured [61].

Some work has also been done in the second level, dealing with the content of these scenes: “Is this an image of a field or of a beach? Is it of a portrait or a picnic?” The second level of semantics seems more difficult to classify than the first. Indeed, some of the existing scene classification systems assume that the image has already been classified physically (e.g. orientation) [52] or compute it on the fly as a preprocessing step (e.g. indoor vs. outdoor) [58, 61]. While our primary interest in this review is to investigate the problem of the second level of scene classification, the first two levels form a continuum ³, so we will review research done on each.

How does one classify scenes automatically? The literature reveals two major approaches: model-based and exemplar-based. On one hand, *exemplar-based* approaches use pattern recognition techniques on vectors of low-level image features (such as color, texture, or edges) or semantic features (such as sky, faces or grass). The exemplars are thought to fall into clusters, which can then be used to classify novel test images, using an appropriate distance metric. We found that most systems use an exemplar-based approach, perhaps due to recent advances in pattern recognition techniques. On the other hand, *model-based* approaches use the expected *configuration* of a scene. A scene’s configuration is the layout (relative location and sizes) of its objects, created from expert knowledge of the scene. While it seems as though this should be very important, relatively little research has been done in this area because it is usually only possible to build a scene model for a constrained scene type and such a model is usually not generalizable to other scene types.

This review is only meant to capture the landscape of a field that is still young and still evolving. The bulk of the material came from the first author’s area paper exploring research directions for his Ph.D. thesis. Whenever possible, we tried to provide readers with links to survey papers on specific topics. Many of the works summarized in this report are deemed

²We do not necessarily interpret these levels as levels of difficulty. For instance, level 3 semantics require one to infer *emotions* of people in the image, which may be harder than describing the image (unless, of course, the level-4 description subsumes the abstractions made at level 3).

³Take, for instance, indoor/outdoor classification. While it is a physical attribute, it does convey a very basic level of semantics.

by the authors as good representatives of existing technologies. Emphasis was also placed on more recent work than earlier, and perhaps more classic, work in the field. For a comprehensive survey of image understanding including historical perspectives of the related fields, readers are referred to Rosenfeld’s survey [39].

This review is organized as follows. After motivating the problem, we proceed into a review of past and current research in Chapters 2 and 3. Chapter 2 is a detailed synthesis of core concepts used in scene classification systems and Chapter 3 is a description of many existing systems. In Chapter 4, we then discuss some of the limitations of these systems and some related open problems.

1.1 Applications of Scene Classification

Scene classification finds many applications. Because its goal is to categorize images according to physical or semantic properties, it can be very powerful. We describe three applications briefly: Content-Based Image Retrieval (CBIR), digital photofinishing, and automatic image orientation.

With digital libraries growing in size so quickly, accurate and efficient techniques for CBIR become more and more important. Many current systems allow a user to specify an image and search for images “similar” to it, where similarity is often defined only by color or texture properties. Knowing the category of a scene *a priori* helps narrow the search space dramatically [22]. For instance, knowing what constitutes a party scene allows us to consider only potential party scenes in our search and thus helps to answer the query “find photos of Mary’s birthday party”. This way, not only is the hit rate expected to be higher, the false alarm rate is also expected to be lower.

Knowledge about the scene category could find also application in digital photo-finishing [52]. When film is digitized, color balancing is applied to correct color cast and enhance the contrast of the image. Unfortunately, while a balancing algorithm, which is often based on statistical measures derived from the image, might enhance the quality of some classes of pictures, it degrades others. For instance, a photograph of a sunset should retain its brilliant colors without having the contrast adjusted. Or an image that contains skin-type colors, but is clearly not skin (to a human’s eye), should not be automatically color-balanced to look like skin.

Automatically categorizing images by their orientation (e.g., landscape vs. portrait) has many applications. Digital photographic images uploaded by consumers otherwise have to be oriented manually. It is desirable to insert the images into albums, digital or hardcopy, in their upright orientation. Knowledge of correct orientation can improve the performance of object detection algorithms. It has also been shown to help the accuracy of other image understanding tasks such as automatic main subject detection [45].

What techniques does one use to classify scenes? We now review the state of the art, both the general theoretical principles and specific systems that have been designed to implement these ideas.

Chapter 2

Core Concepts in Scene Classification

While some scene classification systems are primarily exemplar-based and others are more model-based, the literature shows that they all share a common structure. This chapter is a *synthesis* of the body of recent research in scene classification. Anyone designing a new classification system must choose appropriate features to use for the classification, a method to combine them to make a classification decision, and possibly a geometric model of the scene layout, although research in this last area seems somewhat weak. We provide the reader with an extensive list of the various features and inference engines used in the systems we reviewed.

1. *Features* are extracted from the image in order to simplify the classification and “to bridge the gap between image semantics and the pixel representation” [61], or the image and its symbolic description. Merely passing the value of each pixel to an inference engine would be prohibitively expensive and would, in general, lend no insight into the scene classification problem.¹
2. The features are used by a *learning or inference engine* to classify the image into one or more categories.
3. In addition, model-based systems also allow for the physical layout of scenes to be explicitly specified.

2.1 Features

The term *feature* can refer to any representation of the image or its components, including color, texture, and many others. Although pixel values can be considered a form of feature, albeit primitive, we refer to feature as quantities derived from pixel values in this report. We provide an exhaustive list of all features used in the systems presented in Chapter 3, as well as additional promising features.

¹In object detection, approaches feeding directly off pixel values have shown successes for rigid or nearly rigid objects such as human faces [50].

The choice of features is extremely important, because no matter which inference engine is used, a pattern recognition system can only discriminate as well as the chosen features allow [10].

2.1.1 Color

Many color spaces have been used in the literature, including standard RGB, cylindrical spaces such as HSV, and the CIE models which decouple luminance from chroma components such as $L^*u^*v^*$ and $L^*a^*b^*$ ². Many definitions of other color spaces are provided in [25, 28], which uses 10 color spaces. One of these, the Ohta color space, is particularly interesting. The axes of the Ohta space, defined as $(\frac{R+G+B}{3}, \frac{R-B}{2}, \frac{2G-R-B}{4})$, correspond to the three Principle Components of the RGB space, found by performing Principle Components Analysis (PCA) on a large selection of natural images [52]. However, there is no single color space that has been proven to be best for *all* natural scenes [3].

Color measurements can be used directly at the individual pixel level, as moments of regions (e.g. means or variances), as histograms, or as coherence vectors. Following the treatment of [59], color coherence vectors refine color histograms by dividing each bin into counts of coherent and non-coherent pixels. Coherent pixels are those that are spatially part of a large similarly-colored region (like a pixel belonging to the sky in a landscape image). The original definition of color coherence vectors can be found in [32].

2.1.2 Texture

Texture refers to objects' surface or structure properties [49]. A simple texture measure is the variance of a region's colors, corresponding to the activity in the region [60]. The MSAR (multi-resolution simultaneous auto regression) model predicts pixels based on non-causal³ neighborhoods [52] and has been shown to perform well on static textures (e.g. the Brodatz album) [52]. Fractal dimension is correlated with the coarseness of an object [49].

Wavelet decompositions [23] have been used as texture features as well. The computationally efficient indoor/outdoor classifier described in [44] uses wavelets due to their low computational cost and reasonable performance on natural images.

In general, there is no multi-purpose texture representation that is fastest and works best in all circumstances. One must take into consideration the setting in which the texture feature will be used before deciding which one is best. For a good comparative study of major texture features for texture classification, see [35].

²Definitions of these standard color spaces can be found in any standard computer vision text [2].

³A pixel's *non-causal* neighborhood includes pixels after it in a raster-scan ordering.

2.1.3 Filter output

Some systems use the output of various spatial filters. Examples include Gabor or Gabor-like filters [41], Fourier Transforms (both global and windowed) [29, 52, 54, 55], Discrete Cosine Transform (DCT) coefficients (related to JPEG compression) [58], and spatio-temporal filters [34].

2.1.4 Edges

Edge or gradient information can be used, for example, to distinguish images of man-made structures from natural scenes [58]. Edge direction histograms quantize the direction of edge pixels, such as those detected by a Canny detector, into bins. Edge direction coherence vectors, like ones used for color, store the number of coherent and non-coherent pixels within each direction. Intuitively, this discriminates between structured and arbitrary edges.

2.1.5 Context Patches

As used in [43], a *context patch* is a dominant edge in an object, together with its neighboring edges. The rationale behind using these patches is that they are somewhat invariant to translation, scale and 2D rotation of the object, since the patch is normalized with respect to scale and orientation). The patches can be used by an object recognizer as keys into a lookup table of patches found in the object database.

2.1.6 Object Shape

The Blobworld system [7] computes the area, eccentricity, and orientation of each region thought to correspond to an object. Note that this feature is more accurately described as *segmented region* shape, since object detection is not used, only segmentation. In systems that perform object recognition [43], more accurate measures could be calculated. However, it is unclear whether shape features would be salient features of scene categories in most cases. Certainly, if object recognition is needed to determine these features, the object's classification (e.g., *wall clock*) provides more useful evidence than just its shape (e.g., *circular*).

2.1.7 Semantic Component and Object Classification

If any component of the image can be classified reliably, it can be used as a cue to scene types. Examples of semantic categories that can be classified accurately include sky, grass, water, faces, and skin [22, 45]. In addition, object recognition systems have improved to the degree that it is conceivable that object knowledge may be used within scene classification systems. While a bottom-up strategy in which object recognition is used to label every region

may still be impossible, recognition of some objects could be used to verify scene category hypotheses posed using features more easily derived.

2.1.8 Other Image Understanding Systems' Output

It is conceivable that one could use output of other image understanding systems as a clue of certain scene types. For example, if the main subject [45] is found to occupy a large, rounded region in the center of the image, it may provide evidence that the scene is a person's portrait.

2.1.9 Context and Meta-data

Context is derived from images, either from the statistics of a single image (such as scale, focus, and pose [55]), or from multiple images (such as adjacent images on a roll of film). Meta-data comes from outside of the image, such as digital cameras (which can save the orientation with which the camera was held and whether or not flash was used), online images (which contain captions in the form of HTML "ALT" tags [20]), or video systems (which use closed captions or speech recognition to classify segments of video [14]).

2.1.10 Statistical Measures

If the image features have high dimensionality, projecting into a lower dimensional subspace using linear (e.g. PCA or Linear Discriminant Analysis [10]) or nonlinear (e.g. Locally Linear Embedding (LLE) [40]) methods may be helpful to avoid the curse of dimensionality [10]. K -means has also been used to cluster features to reduce dimensionality [60].

2.1.11 Spatial Layouts

While some features may be considered relevant for an entire image, many of the features we have described take on very different values at different *positions* within a given image, making the spatial layout of feature extraction very important. We discuss the following spatial location options:

1. None (global features). One example is a global color histogram, used by many early CBIR systems (e.g. QBIC [11]) to describe images. Many of the systems described later in this review use these systems as a baseline. Other examples include global edge direction histograms and global Fourier Transforms [29] An obvious limitation of strictly global feature extraction is that individual components' location, shape, and texture are discarded. However, we note that depending on the classification at hand, this may be desirable.
2. Blocks. A simple method of encoding spatial information is to partition an image into a grid and compute features for each block of pixels. Clearly, there is not a one-to-one

mapping between pixel blocks and actual objects, but this information can still be helpful. For example, blue blocks at the bottom of outdoor scenes provide evidence for water, while the same blue colors at the top may signify the presence of sky.

Examples here include spatial color moments [58, 60, 62] and spatial (local) edge direction histograms [62]. Block color means are closely related to low-resolution representations of the image [19, 36]. The regular partitioning is also related to windowed filter-based techniques, such as the Windowed Fourier Transform (WFT) used in [29].

3. Segmented regions. When images are segmented, features can be computed on each region. Again, no segmentation is perfect, so even regions don't map directly to objects. However, more meaningful statistics, such as shape and location, may be calculated on regions. Examples of systems using segmentation include [7], [45], and [61].

2.2 Scene Configuration Modeling

Some researchers have attempted to provide a model for the scenes they attempt to categorize. While global features ignore object or component location altogether, and block features encode it implicitly, scene models detail explicitly the configuration, or relative location, of objects that are the salient for the scenes in which they appear. For a survey on general knowledge-based image understanding systems, see [9].

One moderately successful system using scene configuration was designed by Lipson [18]. She crafted scene models by hand for mountains, fields, and waterfalls. For instance, she defines a field image as one in which a large bluish region occurs over a large greener region. She points out that the strength of her system lies in the flexibility of the template, in terms of both luminance and position. More details are given in Section 3.4.1.

Another system attempting to model each scene was designed by Smith and Li [48]. They segment the image and determine the relative ordering of the segments' colors along vertical scan lines. However, they do not design their models by hand, but learn them from a set of training data.

We discuss the matching algorithms used by systems such as this in Section 2.3.9.

2.3 Learning and Inference Engines

In general, pattern recognition systems are designed to perform classification of entities represented by feature vectors (see a good review in [15]). In the computer vision domain, relevant features (as discussed in the previous section) are extracted from each of a set of training images. To classify a novel test image, the system extracts the same features from the test image and compares them to those in the training set [10]. This exemplar-based approach is used by most of the current systems, as we discuss in Chapter 3.

The classifier used to perform the comparison may vary greatly in sophistication. In this

report, we discuss a few major classifiers, specifically those used in the systems presented in Chapter 3.

2.3.1 K -Nearest Neighbor

One of the simplest classifiers is the 1-nearest neighbor (1-NN) classifier. The system calculates the Euclidean distance from the test feature vector to each feature vector in the training set; the test image is classified with the same label as that of the closest training image [10].

This technique can be generalized easily using a voting technique; the k nearest neighbors in the training set are found and the test vector is given the same label as the majority of its neighbors. In binary classifiers, choosing k to be odd (many systems use $k = 5$ or 7) eliminates the need to break ties.

2.3.2 Learning Vector Quantization (LVQ)

While the simplest method of classifying a test vector is to compare it to its nearest neighbors, finding which neighbors are closest can be computationally expensive, especially for large training sets (Figure 2.1). What can be done is to find a representative set of vectors for each class, called a *codebook*. There may be m codebook vectors for n training samples, $m \ll n$. The test vector is then classified with the same label as the nearest codebook vector (Figure 2.2). This makes sense particularly if the data is clustered and a codebook is placed within each cluster.

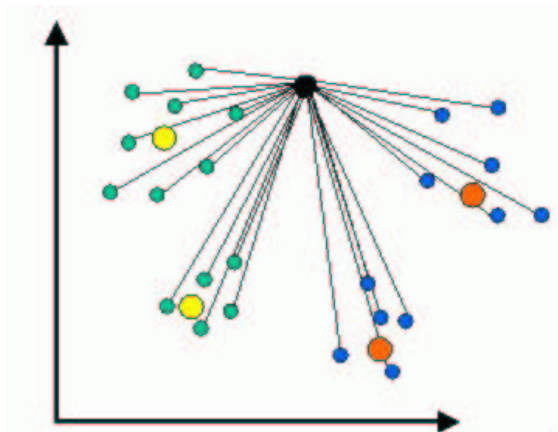


Figure 2.1: Nearest neighbor computation. Calculating distances to each training vector can be expensive.

One issue with LVQ is choosing the size of the codebook. Some choose it using analytic means such as the Minimum Description Length (MDL) principle⁴ [60], while others choose

⁴The MDL principle relates to Occam's razor [1], in this case penalizing a large codebook as well as a bad fit of the codebook to the data [24].

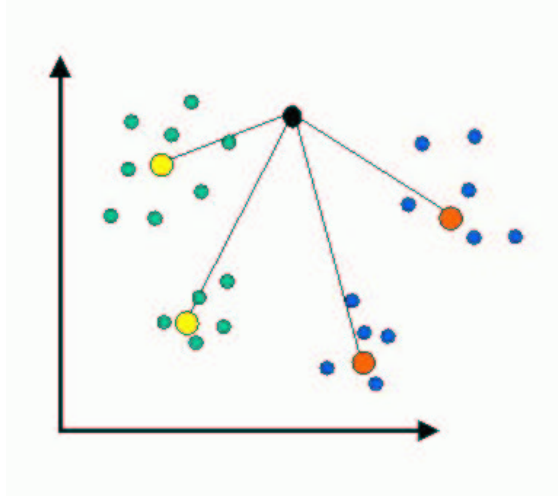


Figure 2.2: Example of Vector Quantization. Comparing the test vector with only the codebook vectors and not each of the training vectors is much less computationally demanding if $m \ll n$.

it empirically. More details about LVQ can be found in Kohonen’s original papers [16, 17].

2.3.3 Bayesian Classifier

A Bayesian classifier, in theory, will give an optimal classification rule within a specified model. However, because the exact distribution of features is unknown, several assumptions must be made to approximate a classification. We present the framework and assumptions used in [58, 60].

First, the underlying probability density function generating a set of data can be approximated using a mixture of Gaussians. If LVQ has been used to quantize the training set, a Gaussian can be centered on each codebook vector.

The class-conditional density function for a feature vector y given class ω is approximated by:

$$f_Y(\mathbf{y}|\omega) \propto \sum_{j=1}^q m_j \cdot e^{(-0.5 \cdot \|\mathbf{y} - \mathbf{v}_j\|^2)}$$

where v_j represents codebook vector j , q is the codebook size, and each weight m_j is the proportion of training vectors closest to codebook vector j , calculated while the codebook is created.

These density functions are the likelihoods for each class, which are used in conjunction with the priors to calculate the posteriors. According to Bayes’ Rule,

$$p(\omega|\mathbf{y}) \propto \mathbf{f}_Y(\mathbf{y}|\omega_i)\mathbf{p}(\omega_i),$$

where $p(\omega_i)$ is the prior for class ω_i ⁵. The posteriors are calculated for each of the classes and

⁵The denominator in Bayes’ rule is just a normalizing scaling factor; we can ignore it since we are only

the class that corresponds to the maximum posterior is chosen. This is called the *Maximum A Posteriori (MAP)* principle.

Without the weighting factor m in the Gaussians, this approach would mirror a k -NN approach. The m_i factor is a weighting factor, giving a larger weight to those codebooks which are in the center of a dense cluster (Figure 2.3). The weights can be calculated while the codebook is being constructed. This better approximates the true underlying density function. Further details of the MAP classifier can be found in [58].

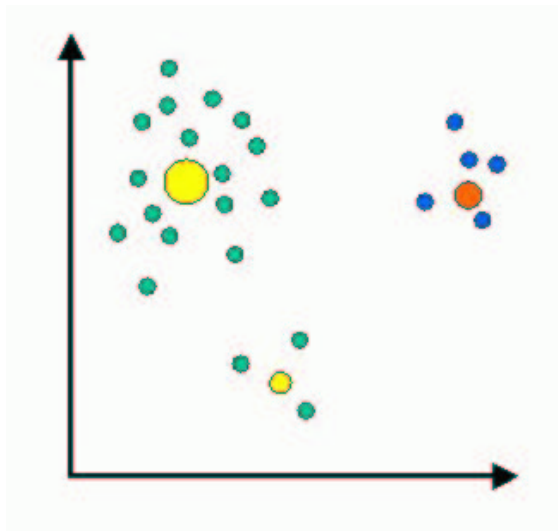


Figure 2.3: Codebooks within a larger cluster are given more weight in the Bayesian classifier.

2.3.4 Support Vector Machines (SVM)

Support vector machines are based on minimizing the number of classification errors by placing optimal borders between the classes. Unlike the previous classifiers explained above, one does not need to specify the number of codebooks or the number of nearest neighbors to use; the number of support vectors is calculated automatically by the SVM.

SVMs map data points into higher dimensions where the points are more likely to be separable. This mapping occurs via a kernel function, which can be linear, polynomial of various degrees, or Gaussian (Radial Basis Function, or RBF). The kernel specifies what form the border between classes will take.

SVMs are by nature dichotomizers, so they only work for distinguishing between two classes of data. However, one can use multiple SVMs for n -class problems. There are two techniques discussed in the literature [42].

The first technique is a sort of “winner-take-all” approach, where n classifiers are trained, each of which can distinguish one class from all of the others. A test vector is then classified

calculating the *argmax*.

by each of the classifiers and is assigned the class corresponding to that of the machine producing the highest real-valued output.

The second technique uses pairwise classification. Here, $\frac{n(n-1)}{2}$ SVMs are trained, each of which can distinguish between one pair of classes. A test vector is then classified by each of the classifiers and a vote (using the signum output of the SVMs) is taken. The vector is classified with the class receiving the most votes, with the real-valued output being used to break ties. While using more SVMs, each takes less time to train, so there is not a large performance penalty.

2.3.5 Artificial Neural Networks

Neural networks are generalizations of content addressable memories, which use hidden units to gain more power ⁶. In the most common network, the *feed-forward* network, information flows forward in the network from the inputs, through the hidden layers, and into the outputs, which give the classification. While the inputs to each layer are linear combinations of the previous layer, continuous activation functions allow the network to be trained using backpropagation techniques, which are equivalent to hill-climbing. More details can be found in [1].

2.3.6 *K*-Means

The *k*-means algorithm uses a two-step process for unsupervised clustering of data into *k* clusters [24]. First, it initializes *k* cluster centers, often randomly. It then classifies each data point with the same class as its nearest neighbor cluster center. The cluster centers are then updated to the mean of the exemplars classified with its class. The exemplars are then reclassified, and the process is iterated until a convergence criterion is met. One drawback of *k*-means is that *k* must be known in advance; if not, various values of *k* can be used and the best can be selected. One could choose the best value of *k* by measuring the fit of the clusters to the data or by using the Minimum Description Length principle to select it, namely by always choosing a smaller value of *k* when two values fit the data equally well.

2.3.7 Expectation Maximization (EM)

The EM algorithm can be viewed as a probabilistic, or “soft” version of *k*-means. It also uses two steps, but may assign each exemplar to more than one class, with weighted membership probabilities.

One can think of EM as estimating the value of hidden parameters [24]. For instance, if one thought that the data is best modeled by multivariate Gaussians, EM could be used to estimate the mean vectors and covariance matrices of each cluster. The two steps can be described as follows:

⁶They can also be considered function approximators.

- *Estimate*: if we knew the hidden parameters, we could compute expected values (i.e. cluster membership probabilities).
- *Maximize*: Given the expected values, we can then re-compute the values of the hidden parameters that maximize the likelihood of the data.

Both the k -means and EM algorithms are hill climbing methods, only guaranteed to find local maxima.

2.3.8 Bayesian Networks

Bayesian (or *belief*) networks are used to model causal relationships when the evidence is probabilistic in nature [8]. The network is a directed acyclic graph which encodes conditional independence between variables. Because of these independences, the joint probability distribution of all of the variables in the system can be specified in a greatly simplified way. This allows for fast inference when the graph is singly-connected.

Each node represents evidence, while the arcs represent causalities. Specifically, the network consists of four parts, as follows [45]: *Priors* are the initial beliefs about the root nodes in the network. Each node has a *conditional probability matrix* associated with it, representing the causality between the node and its parents. These can be assigned by an expert or learned from data. *Evidence* is the input presented to the network. *Posteriors* are the output of the network.

Bayesian networks have been used in many computer vision applications including control of selective perception [38], main subject detection [45], and indoor/outdoor classification [22, 30]. An advantage of Bayesian networks is that they are able to fuse different *types* of sensory data (e.g. low-level and semantic features) in a well-founded manner. For further details of Bayesian networks, we direct the reader to Pearl [33].

2.3.9 Configuration-based Engines

Systems that explicitly encode the configuration of objects in the scene may need to use a custom matching process to perform the classification. Lipson's system described in [18] uses such a process. The hand-crafted model is a template of scenes in a given class, and contains relative x- and y-coordinates, relative R-, G-, B-, and luminance values, and relative sizes for two regions in the image. As the model is compared to the image, the model can be deformed by moving the patch around so that it best matches the image. Her current approach is if one configuration of the model matches the image, then it is a match, but this criterion may be extended to include the degree of deformation and multiple matches depending on how well the model is expected to match the scene.

2.4 Control Strategies

Another issue in computer vision in general, and scene categorization in particular, is the overall control of the system. What drives the system? What is done first? How are hypotheses formed and verified? While many of the systems using strictly low-level features that we will describe can operate using single classifiers, it is possible that a more comprehensive approach, incorporating more evidence, is necessary for robust scene classification.

We now summarize Battle *et al.*'s review of control strategies given in [3]. While the focus of their work is on recognizing objects in outdoor scenes, and how control affects region segmentation, labeling, and feature extraction, the concepts also apply to scene categorization. (Haralick and Shapiro [13] also give a fairly comprehensive review, which we do not cover specifically here.) We describe the three types of hierarchical control: top-down, bottom-up, and hybrid.

2.4.1 Top-Down Control

Top-down control can be best described as “hypothesize-and-test”. Once a hypothesis is generated, segmentation routines specialized to the hypothesized object can be used. The features to be extracted can also be tailored to the object hypothesized and can vary greatly. The regions created by the segmentation routine are then checked against what is expected by the model. If they do not match well enough, another hypothesis is generated, and the process is repeated.

Since it is completely goal oriented, the top-down strategy is the most efficient strategy of the three. It is limited by its inability to handle unexpected regions, but can handle variations, exceptions, and special cases that are known *a priori*. In general, it is used in Special Purpose Vision Systems (SPVS) ⁷ in constrained environments, and thus is more successful at a more humble goal. It was used often in the 1990s. Marti's system of describing outdoor scenes [25] is a good example of a system using top-down control.

2.4.2 Bottom-up Control

Bottom-up control takes the opposite approach, assuming no knowledge during the low-level image processing stages. For instance, segmentation and feature extraction are general purpose and non-semantic. Once each region is assigned a feature vector, each region is labeled using an object model database and an inference engine.

Systems using the bottom-up strategy are much better at handling unmodeled regions than those using the top-down strategy. This strategy has been used since the 1970's, but there has also been a recent surge in use because of recent advances in AI techniques.

⁷Compared to biological vision, *all* existing computer vision systems are special purpose. Therefore, we interpret the authors' use of SPVS [3] to mean systems that are *highly* constrained, like assembly-line inspection problems under fixed lighting and orientation.

2.4.3 Hybrid Control

Hybrid approaches seek to get the best of the other two approaches [3]. The most accepted hybrid approach, according to Battle *et al.*, is to use general purpose segmentation to obtain regions, but then use specialized feature extraction. This top-down extraction can be used to correct the errors that often occur in segmentation, but is more robust than a pure, top-down approach. Wang [61] uses an approach like this, extracting features depending on segmentation-based hypotheses.

Applied to the realm of scene classification, a hybrid approach conceivably could be devised in the following fashion: first a bottom-up process is employed to generate some pre-cursors to a pre-determined database of scene models; the most likely scene models can then be used to perform “hypothesize-and-test”, including perhaps triggering searches for additional cues either to look for additional confirmation or to refine the level of scene classification. A scheme in a similar vein was described in early work by Ohta on knowledge-based interpretation of outdoor natural scenes [28].

2.4.4 Cost-based Control

Rimey and Brown [38] use Bayesian networks to control selective perception. The control strategy used to recognize a scene was determined by weighing cost (defined as real-world execution time) of vision routines vs. their expected benefits.

Mirmehdi *et al.* [27] use feedback between low-level and high-level routines to determine the control strategy. Their strategy is designed to minimize the search time, focusing high-level processing power only on regions of the scene where the target object may be located.

Chapter 3

Scene Classification Systems

In this chapter, we review many systems that perform image and scene classification. We found that most systems in the literature were exemplar-based, using a training set and pattern recognition techniques. However, a few systems were model-based, using the configuration of the regions in the scene to infer the scene’s category. We review examples of each type in detail.

3.1 Exemplar-based Systems

Exemplar-based systems use the pattern recognition techniques and classifiers described in Section 2.3; due to recent advances in machine learning and pattern recognition techniques, these systems are currently very popular.

As stated in Section 2.1, a pattern recognition system can only discriminate as well as the chosen features allow. Moreover, studies show that various features can help to identify some scene types much better than others. For example, Vailaya and Jain show in [59] that on one hand, for classifying outdoor scenes into city and landscape scenes, edge features gave approximately 20% higher performance than color features (93% vs. 75%). On the other hand, when discriminating forest from mountain scenes, color features gave 8% better performance (91% vs. 83%) than edge features¹. In short, the “best” features to use depend on the classification task.

The above study compared only low-level features. The choice of features, however, is more general. While low-level features that do not require the image to be segmented are thus simpler, there are perhaps advantages to segmenting the image first and searching for specific objects. For instance, accurate sky, grass, and face detection have been found to improve performance on certain classification problems [22].

¹The relative performance is what is important. The absolute performance numbers given are for a constrained version of the problem (see Section 4.1).

3.2 Exemplar-based Systems Using Low-level Features

Low-level features, such as colors, textures, and edges, have the advantage of simplicity. Global or local features are calculated for each image without first segmenting the image. In this section, we review a number of such systems.

3.2.1 Automatic Orientation Detection

Detecting the orientation of an image automatically is also an important image classification problem. Assuming that the images are not tilted arbitrarily, there are only four classes: north-, south-, east-, and west-oriented images. These correspond to how photographs would be aligned for scanning or how a camera is held while taking a picture. In [60], Vailaya *et al.* use spatial color moments and features, finding the mean and variance of each band in $L^*u^*v^*$ color space for each block of a 10×10 grid.

They use a Bayesian-based classifier (using a mixture of Gaussians) to obtain posterior confidences that could be used for a reject option. They quantize the training data using LVQ to improve the efficiency of the system. They report accuracy of approximately 89% on an independent test set, assuming equal priors on the orientation.

However, follow-up studies using different data sets call into question the generalizability of this method. Recent work by Wang and Zhang [62] (who incidentally, co-authored [60]) reports 69% performance on the same problem using an LVQ classifier, increased to 74% using an SVM classifier. They then further increase performance to 78% by adding an edge direction histogram SVM classifier and combining the results from the two. Boutell and Luo report similar numbers (70% LVQ, 74% SVM) [4].

The reasons for the discrepancies are not entirely clear, but Vailaya cannot supply the exact split of the database into training and test sets [57], so the experiment cannot be replicated. Yang *et al.*, in personal correspondence [63] stated that in their experiments in a follow-up study [64] using SVMs, they were given the training and test sets already preprocessed, so they never saw the original images or knew how they were split. Furthermore, while the numbers reported in [4, 62] do not use rejection, it is not clear whether the numbers in the original paper [60] do.

3.2.2 Indoor-Outdoor Classification

Szummer and Picard [52] obtained very good results on a specific problem: classifying images as indoor or outdoor. The problem is straightforward, as it is a binary decision, with a very small amount of ambiguity. Breaking the image into sub-blocks and using Ohta-space color histograms and MSAR texture features on each sub-block, they obtain 90.3% performance. k -NN classifiers are used to vote, classifying each of 16 sub-blocks using both color and texture, for a total of 32 votes. The image is then classified using a majority voting scheme. Performance is reported on a database of 1343 consumer photographs using the leave-one-out method. The caveat to this performance is that there are “near-duplicate” images within

this database, artificially boosting the performance by over 5% according to a Kodak study [44], which also reported slightly over 90% performance on a subset of the same database after pruning the “near-duplicate” images.

Paek and Chang [30] also obtained similar results using a much different approach. They calculated HSV color histograms and edge direction histograms, representing color and texture, for each block in the image. They classify each image using a k -NN classifier, with histogram intersection as a distance measure. They trained classifiers to recognize images as indoor/outdoor, sky/no sky, and vegetation/no vegetation (as secondary cues for the indoor outdoor problem). They then feed the classification results of each into a belief network. Using a set of 1708 consumer photographs (which they claim to be the same set of images used in [52]), they obtain 83.1% accuracy with a single indoor/outdoor classifier and 86.3% when they use the belief network. They claim that if they included “near-duplicate” images (such as would be typical of consumer images) in their testing, their accuracy would be approximately 90%, matching Szummer and Picard ².

3.2.3 Hierarchical Image Classification

In [59, 58, 56], Vailaya *et al.* show that within a constrained domain, low-level features can successfully discriminate between *many* scene types arranged hierarchically. Performance and the features used are reported for the following scenes: vacation images into indoor vs. outdoor (88.7% using spatial color moments), outdoor images into city vs. landscape (93.8% using edge direction coherence vectors), landscape images into sunset vs. forest/mountain (94.3% using color coherence vectors and edge direction coherence vectors), and finally the last images into forest vs. mountain (93.5% using color histograms and edge direction coherence) ³. This hierarchy of categories was chosen based on a study in which users were asked to categorize a set of 171 vacation images. Testing and training were performed on a database consisting of 6931 images.

These results were the highest taken from an extensive study evaluating the saliency of various feature sets, as discussed briefly above, such as color histograms, color coherence vectors, spatial color moments, edge direction histograms, edge direction coherence vectors, and DCT coefficients, using k -NN classifiers to evaluate each feature or combination of features.

As with the automatic orientation detection research, they later switched to a Bayesian-based classifier (using a mixture of Gaussians) to obtain posterior confidences that could be used for a reject option. They quantized the training data using LVQ to improve the efficiency of the system.

One limitation inherent in hierarchical classifiers is the cascading of errors. While the performance at each level of the hierarchy is excellent, it is an easier problem. For instance, in the city vs. landscape problem, all of the images are known to fall into one of the two categories,

²thus implying that Szummer and Picard’s results were slightly inflated.

³These accuracies are for their test sets only, and thus are slightly lower than the numbers they report in their abstract, which include performance on the training set.

allowing a forced classification. To classify an unknown vacation image as a specific type is more difficult. For example, we desire to classify an unknown vacation image as a forest image. We first hope to classify it as an outdoor image (with the possibility of wrongly calling it an indoor image). We then classify it as a landscape (but possibly calling it a city scene) and repeat for the landscape classifiers. By multiplying the above performance numbers, we see that theoretically a forest image can only be classified as such with 77% accuracy (if the classification choices at each level are treated as probabilities and each choice must be correct for a correct final classification). While this number may vary in practice, it serves to illustrate the problem.

3.2.4 Holistic Classification Using the "Spatial Envelope"

Research at MIT on scene categorization also uses low-level features. Oliva, Torralba, and Sinha [29, 54, 55] cite biological vision as their motivation: humans can recognize scenes and animals seem to, both seemingly without recognizing all of the individual objects, thus implying a *holistic* approach, as opposed to a bottom-up, parts-based one [54].

These researchers represent scenes using the output of a windowed Discrete Fourier Transform (DFT). (They also use the global DFT in [29], but report better accuracy with the localized version.) The image is sampled at a low resolution to make it more robust in the presence of changes in location and orientation of individual small objects. Since the dimensionality is still high, the principle components are extracted using PCA.

This non-intuitive representation is projected onto what the authors term the "spatial envelope" of an image [29] using linear regression. The spatial envelope was based on a study where people were asked to categorize a set of 81 images into groups *not* based on objects in the scene or semantic scene types (beaches, parks, etc.) but on global structure. It consists of five qualities: "naturalness" (vs. man-made), "openness" (presence of a horizon line), "roughness" (fractal complexity), "expansion" (perspective in man-made scenes), and "ruggedness" (deviation from the horizon in natural scenes). Each spatial envelope feature corresponds to a dimension in this space.

These projections are then used to classify novel images as natural or man-made (93.5%), obtaining similar results for the other dimensions.

Natural images are then projected along the openness and ruggedness dimensions. A k -NN classifier then uses labeled scenes to categorize them further into coasts, country scenes, forests, and mountains, with 89% accuracy. Similarly, urban scenes are projected into openness/expansion space and classified as highways, streets, tall buildings, and close-up scenes with 88% accuracy.

The authors acknowledge that the categories chosen correspond nicely to the features used; for instance, the horizon line used to discriminate "openness" is very obvious in the spectral space. In many ways, it seems as though, while the mathematics is much more involved, that the system is using edge information to distinguish at least some of the categories. It makes sense that its performance should be similar to Vailaya's. One advantage of their approach is that once the projections are performed, the classifiers can be very simple.

Torralba and Sinha’s system [54] uses similar features to recognize indoor scenes. Scenes are represented using outputs of Gabor filters tuned to different spatial frequencies and orientations (very much like the output of a windowed Fourier transform). Scenes are classified using “visual scene landmarks”, features that are unique within the training set (salient and specific to a given room). Specifically, ambiguous features do not help to identify an image; conditional entropy of each image is calculated and generic views are recognized as such by their high measures of entropy. A Bayesian classifier with Gaussian distributions centered on each of the training images is used; if the posterior is found to be below a threshold, the image is considered generic and is thus not classified. Otherwise, the image contains a landmark, and so is classified with the same label as the test image containing that landmark.

The system is trained on 1500 images taken from video footage of fifteen rooms. Its performance is tested by processing another similar video and classifying each image. In this setting, the threshold can be set high enough to accept the 40% of the frames about which the system is most confident. These key frames can then be used to classify adjacent frames, labeling rooms correctly with 95% accuracy. However, the authors state that the actual performance depends on the size of the training set, the number of features calculated per image, and the threshold used ⁴.

Torralba and Sinha conclude that the holistic approach serves as a plausible biological model. They question its extensibility to novel environments and to other low-level feature sets. A recent study at the request of Kodak using consumer photographs⁵, which do not necessarily fall nicely into well-defined scene categories, yielded mixed results when images are labeled in terms of “openness”, “naturalness”, “depth”, “expansion”, and “busyness”.

The authors also discuss the possibility of holistic features being used synergistically with an object detection system, with the holistic features serving to prime the object detector (discussed in Section 3.5). This is an interesting direction for future work.

3.2.5 Scenes Dominated by Single Objects

Schmid [41] developed a system that extracts Gabor-like features for each pixel, and classifies each one into one of a group of generic descriptors (using a k -means algorithm). They are then clustered spatially using Bayes Rule. Finally, each cluster’s “significance” is computed, depending on its preponderance in positive vs. negative training images.

This is a CBIR system not used specifically for classification but for retrieval. The accuracy is not good, given the size of the database (only 660 images). One strength of the system is that it not only retrieves images, but localizes the objects in the images that match those in the query image. It does seem good for objects that are highly textured and deformable (e.g. zebras).

⁴The 95% figure corresponds to 80 features, 100 images/room and 40% of frames labeled.

⁵from the JBJL database.

3.3 Exemplar-based Systems Using Segmentation

In the systems so far, low-level features are computed for blocks or the entire image, purposely *not* attempting to detect objects. We now turn our attention to systems that do attempt to find objects, usually in a bottom-up fashion, by segmenting the image. While perfect segmentation is nearly as difficult as the general problem of image understanding, relying on object knowledge, these systems try to compensate for imperfect segmentations. One advantage to these approaches is that the number of segments in an image is much smaller than the number of pixels, and tends to be smaller than the number of blocks. Other advantages are that segmented regions correspond better to objects than blocks do and that they allow more types of features to be extracted [45].

3.3.1 Semantics-Sensitive Feature Extraction

The SIMPLIcity” system (Semantics-sensitive Integrated Matching for Picture Libraries) was developed by Wang *et al.* [61]. Recognizing that the features extracted determine how well one bridges the gap between semantics and pixel representation, Wang *et al.* extract features on the fly based on the results of a few initial, easily-determined classifications.

For example, they segment the image at a low resolution and use model-based approaches to determine if an image is a photograph or a graph (containing text, clip-art, figures) and whether it is entirely filled by a texture or not. These classifications are quick and accurate. Other types of classifiers in this category, including “indoor/outdoor”, “city/landscape”, or “with people/without people” have been studied elsewhere.

Then based on the classification found, features salient for that class are extracted from each region. This principle is not much different than that of Vailaya’s hierarchical system explained in Section 3.2.

Once the features are extracted, the regions from the query image are matched with regions in the image database using a technique the authors call “Integrated Region Matching”. The fuzzy nature of this method, allowing a region in one image to match with several in another image, compensates for potentially poor segmentation.

Scene classification results are reported for a random sampling of 10 categories, corresponding to Corel CDs. Precision within the first 100 images (which equals recall in this special case of 100 images per class) retrieved averages 40% for most categories, as compared to color histograms, which averaged 15-30%⁶. While these figures may seem low, this is a promising result for CBIR, considering their database contains 200,000 images.

⁶Recall = (number of true positives)/(number of true positives + number of misses). Precision = (number of true positives)/(number of true positives + number of false positives). These two values together can be interpreted as a point on an ROC curve. The two are related inversely: by changing system parameters, one can often increase one value at the expense of the other.

3.3.2 Blobworld

The “Blobworld” system [6, 7], developed by Carson *et al.* at Berkeley, was created primarily for CBIR. In an attempt to fill the gap between objects and low-level features, it represents each image as a set of regions containing coherent color and texture. Each pixel’s HSV color and texture is extracted, and the EM algorithm is used to obtain a rough segmented image, with the number of regions obtained guided by the MDL principle. The segmentation is cleaned up using voting techniques and connected components analysis. Features are then extracted for each region: color, texture, area, orientation, and eccentricity. The location of each object is not used.

Its strength as a CBIR system is its ability to visualize each of the query objects, providing the user with intuition and control over the query.

In follow-up work [5], Carson *et al.* use the “blob” representation of objects for scene classification. Each blob’s features are quantized down to a lower dimensionality (13 colors and 6 textures, ignoring shape descriptors) before being increased again by adding spatial information (coordinates within a 3×3 grid imposed upon the image). The system is trained by having each blob cast a vote into the appropriate bin (or bins if the object spans two blocks in the grid); the counts are then used as probabilities. The system then estimates $P(C_i|blob)$ using Bayes Rule and uses the MAP principle to classify novel test images.

In their experiments, the researchers obtained better results than global color histograms for most of the classes they tested. Classes tested and accuracies include [5]: airshows (80%), bald eagles (48%), polar bears (54%), brown/black bears (19%), elephants (36%), tigers (54%), cheetahs (36%), mountains (63%), fields (48%), night scenes (79%), deserts (48%), and sunsets (89%). Each class had roughly 60-70 images. In general, it performed much better on the landscape images than the animal images, possibly due to the salient color layouts of many landscape scenes (e.g. blue on top, brown or green on bottom). Note that it performed best on sunset detection. However, we have reservations about these performance numbers due to the meager size of the image set used: while they state that they have a collection of 28000 images, they only use 1080 of them (12 classes with approximately 90 images each). Furthermore, they train on 2/3 of the images, leaving only 30 images per class for testing.

3.4 Model-based Systems

Model-based systems rely heavily upon the configuration of the scene components. In this section, we review two such systems.

3.4.1 Configuration-Based Classifiers

Grimson, Lipson, and Sinha at MIT use an approach they call “configural recognition” [18, 19], using relative spatial and color relationships between pixels in low resolution images

to match the images with class models.

The features extracted in this system are very simple. The image is smoothed and subsampled at a low resolution (ranging from 8×8 to 32×32). Each pixel represents the average color of a block in the original image. For each pixel, only its luminance, RGB values, and position are extracted.

The hand-crafted scene models are also extremely simple. A template for a snowy mountain image, for instance, is a blue region over a white region over a dark region. The relative (not absolute) values of the color positions are used in the matching process, in an attempt to achieve illumination invariance, while using relative positions mimics the performance of a deformable template.

Classification is binary for each classifier. On a test set containing 700 professional images (the Corel Fields, Sunsets and Sunrises, Glaciers and Mountains, Coasts, California Coasts, Waterfalls, and Lakes and Rivers CDs), the authors report recall using four classifiers: fields (80%), snowy mountains (75%), snowy mountains with lakes (67%), and waterfalls (33%). Unfortunately, exact precision numbers cannot be calculated from the results given.

The authors state that each class model captured only a narrow band of images within the class and that multiple models were needed to span a class.

In a follow-up study by Ratan and Grimson [36], they also used the same model, but learned the model parameters from exemplars. They reported similar results to the hand-crafted models used by Lipson.

3.4.2 Composite Region Templates

Smith and Li at IBM developed a system for classifying images using “composite region templates” [48]. This system uses vertical scans of a segmented image to impose a relative ordering upon objects in the image. If different semantic classes have different order signatures, then the ordering can be used to classify images.

The image is segmented using color back-projection, closely related to the method of Swain and Ballard [51]. The basic idea, as described in [47], is to back-project the quotient of a query histogram and the image histogram onto the image, giving the most likely location(s) of a spatially-localized color histogram within the image. They perform this for each quantized color (from 166 color histogram bins) during preprocessing, allowing for quicker color indexing. They then use morphological operations to obtain the color regions.

The image is then divided into five columns and each column is scanned from top to bottom. The authors justify the use of vertical orderings with the observation that flipping an image horizontally often does not affect the semantic content of the image, while flipping it vertically does [48]. The order in which the regions appear in the column is saved into a *region string* $s_0 s_1 \dots s_n$, where the s_i correspond to the color labels.

The *composite region template* is then a relative ordering of L symbols, $T = t_0 t_1 \dots t_{L-1}$, where T_i precedes T_{i+j} in a symbol string for $j > 0$. The function $I(S, T)$ counts the number of occurrences of a template within a symbol string. For example, given a string

$S = s_5s_4s_5s_7s_1s_7s_3$ and the template $T = s_5s_7$, $I(S, T) = 4$, since s_5 occurs before s_7 in four different ways in the string. If the string represents blue over brown, there would be four times that it occurred in the column.

Finally, a *CRT descriptor matrix* is defined as a matrix M where m_{ij} is the count $I(S, s_i s_j)$ of each 2-dimensional CRT, $T = s_i s_j$ in S .

In summary, the frequencies of the CRTs (which colors appear above which other colors in the columns (taken from the region strings)) are summed over all columns in the image and saved into the CRT descriptor matrix, where M_{ij} gives the frequency of color i appearing above color j . This seems to be a limited, low resolution version of a color co-occurrence matrix.

Then, the matrices of every image in a semantic class (say sunsets) are summed; the set of matrices spanning all classes forms a CRT library. These matrices can then be used to classify images. Each entry in the matrix gives the likelihood that the CRT is found in that class. The class best explaining the image is determined using the MAP principle. Specifically, for each CRT relationship T_i in a test image, $P(C_k|T_i) = \frac{P(T_i|C_k)}{P(T_i)}$ (we assume they use equal priors on each class). The classification is then given as $argmax_k \sum_i P(C_k|T_i)$.

A system was trained on 91 images and evaluated on 266 images. The classes used consist of beach, buildings, crabs, divers, faces, horses, nature, silhouettes, sunsets, and tigers. The overall accuracy reported was 70.7%, slightly better than a color histogram benchmark of 67.3%. However, for certain classes (sunsets, buildings, and beach scenes), it performed by over 10% better. For certain classes, it is possible that the configuration signature is strong enough to make the scenes easily recognizable by such a system. However, the results of this study may not be reliable. Again, very few images were included and in addition, some choices of the scene classes were questionable, perhaps for maximum separability from others.

3.5 Contextual Information

Context information was discussed in Section 2.1 as a feature useful for scene classification. While the literature does not contain context-based classifiers specifically, we now review Torralba and Sinha’s model, which uses context to aid in object recognition [53, 55]. A model like this could be extended to use in scene classification.

Many current object recognition techniques match a model or exemplar to each neighborhood in the image (e.g., template matching) to find where there is a good match. However, this can be computationally expensive, as it requires an exhaustive multi-scale search across the image.

It is been shown ([31] and four other references in [53]) that humans make use of context to facilitate object recognition. There are at least two roles of context: (1) to aid when the object has very little intrinsic structure, as when it appears at a very small scale, and (2) to be used as a preprocessing step, where knowledge of context can cut down on the search, by giving cues to focus and scale, for instance. This is only useful if the context is less expensive

to compute than the exhaustive multi-scale search.

Torralba and Sinha introduce the concept of “statistical context priming” to aid object recognition. They consider the whole image as an approximation of context, under the assumption the object does not occupy a large portion of the image. They define context priming as the conditional PDF: $P_c(p, \sigma, x, o_n | v_c)$, where p = pose, σ = scale, x = location, o_n signals the presence or absence of an object, and v_c is the set of image measurements. They further break down this PDF, using Bayes rule successively, to $P_c(p, \sigma, x, o_n | v_c) = P_p(p | \sigma, x, o_n, v_c) P_s(\sigma | x, o_n, v_c) P_f(x | o_n, v_c) P_o(o_n | v_c)$.

In order, the right-hand side of the equation refers to conditional density functions for:

- Pose (the most likely poses or points of view of the object)
- Scale (the most likely sizes and distances of the object)
- Focus (the most likely location of the object)
- Object priming (the most likely objects given contextual information)

It is thought that if these PDFs were known, they could be used to prime the object detector, by providing priors. They approximate the PDFs by a mixture of Gaussians learned from a set of training data, using EM on the same raw features as those used in the “spatial envelope” [29] (see also Section 3.2.4).

We focus on two of the PDFs, object priming and focus of attention:

3.5.1 Object Priming

Object priming ($P_o(o_n | v_c)$) gives the probability of seeing various objects, given the context, capturing the intuition that “cars should not be found in indoor scenes”. The authors report results on four categories of objects (people, furniture, vehicles, and trees) in [53], claiming 95% precision when it is forced to decide on at least 50% of the images⁷, using two Gaussians and 32 contextual features. It bases its probabilities on the *possible* presence of an object, not the actual presence, returning a high probability if it is consistent that the object would appear. For instance, it may return a high probability for a person in a kitchen, even if no person appears in the image.

3.5.2 Focus of Attention

Focus of attention ($P_f(x | o_n, v_c)$) is modeled after biological vision, which seems to concentrate its resources on specific regions in the visual field when searching for specific objects depending on context [37, 55]. The authors report results on the search for human heads, defining the focus as the vertical portion of the image most likely to contain a head⁸. They

⁷This threshold can be changed in the system.

⁸They also considered horizontal position, but found a nearly uniform distribution in their training set.

report that when the portion to be searched is 33% of the image, that 87% of the heads in the test set were located in these portions.

Chapter 4

Discussion and Conclusions

As we have seen, many of the existing scene classification and image retrieval systems developed use low-level features in order to *infer* the semantics of the images. For some scene types, they have been successful. However, due to the gap between image semantics and pixel representations, we believe there is a limit to the accuracy obtainable by a system that relies *solely* on low-level features. In this section, we outline the main shortcomings in current low-level feature-based research. We then discuss the need for more work in scene configuration modeling. We conclude by discussing perhaps the most important work yet to be done, that of incorporating scene *context* into the classification process.

4.1 Shortcomings of Low-level Feature-Based Systems Reviewed

We have two main concerns with the performance reported for the systems we have reviewed that use low-level features. The first is their generalizability. Many of the results reported are on a constrained set of images, the Corel stock photo library. While an excellent resource in that it contains many images spanning a large range of semantic categories, the photos are taken by professionals, and as such, there is less variation than found in consumer photos because professionals give greater care to color and composition [44]. Because of the variation encountered in consumer images (e.g., non-centered main subjects, background clutter, inaccurate exposure), classifying them is a more difficult problem. Furthermore, unlike Corel stock photos, which were nicely sorted into categories with deliberate intention, many of the consumer pictures do not fall into well-defined categories. Can the current approaches generalize well to real-world, unconstrained photographs by consumers?

In addition, often researchers show the promise of a system by testing it on a set of images that are even further constrained. As an example, the categories used by Vailaya in [58] and [59] are chosen specifically to be those that seem to separate nicely due to their scene content. For instance, he states in [58] that

“we thus restricted classification of landscape images into three classes that could

be more unambiguously distinguished, namely, sunset, forest, and mountain classes. Sunset scenes can be characterized by saturated colors (red, orange, or yellow), forest scenes have predominantly green color distribution due to the presence of dense trees and foliage, and mountain scenes can be characterized by long distance shots of mountains.”

It is understandable that the researcher would take this approach, and his research is indeed a valuable contribution, since he does appear to solve the problem in a constrained domain. However, there is clearly still room for improvement, and more importantly, there is clearly *demand for improvement for practical applications*.

As another example, Torralba and Sinha reported impressive results when labeling rooms within a single building, an extremely limited domain [54]. They themselves question the generalizability of their system and state that extending it is a direction for future research.

Our second main concern is that in some cases, the high numbers reported may convey the (misleading) impression that the scene classification problem is solved, while we know that there is indeed much work remaining to be done. These numbers were frequently obtained from, in fact, constrained training and testing data. In some of the research, we were unable to duplicate the results of the original work. While we know the images that were used in the entire set, we do not know how they were split into training and test sets. Clearly, the performance of an exemplar-based system depends heavily on the exemplars used, but it is rarely published which images were used to train the system and which were used to test its performance. We know that splitting the dataset in different ways can cause much different performance (see experiments in [4], for instance).

Furthermore, different researchers used different metrics in reporting their experiments. For example, the accuracy reported in the abstract of [58] was for the entire database, *including* the training set. This non-standard practice inflates the numbers by an average of 1-2%. While the authors state the individual training and test results later in the paper, the numbers are misleading at first glance.

In summary, while accurate performance on scene classification has been reported by a number of researchers, some results correspond to constrained scene types (classifying rooms within a building) or to using a set of professional photos and splitting the training and test sets in such a way that the experiment cannot be duplicated.

4.2 Limitations of Low-level Features

We have mentioned that there is room for improvement with systems that use low-level features. We question whether more accurate or more generalizable results can even be obtained using these features. One interesting direction would be to conduct a study of the cues humans use to classify images. A search of the psychological literature shows that very little research has been done in that area, at least in the case of image orientation. Recently, the research community began to recognize the importance of understanding how humans perform image understanding type of tasks [26, 21].

Many researchers stated that they chose to use low-level features for one of two reasons:

1. they were trying to model human vision, which they believed used a holistic approach without object recognition [29], or
2. high-level features (such as object or component recognizers) were either too expensive to compute or their results were too unreliable to use for categorization purposes. As stated earlier, object recognition in unconstrained environments is still an open problem.

However, we believe that it is now possible, and in fact, necessary, to incorporate higher-level, semantic features to perform the categorization, for the following reasons:

1. Low-level features such as color and textures are not always directly related to semantic interpretation and therefore unable to bridge the “semantic gap”, which is “the lack of coincidence between the information that one can extract from the visual data and the interpretation that the same data have for a user in a given situation” [46].
2. Computational power has increased, so that we are able to detect objects and components in a shorter time than was previously possible. As an example, view-based recognition of potentially occluded objects using a database of 30,000 views can be done in 1-2 seconds on a 1.5 GHz processor [12]. We believe that this trend will continue.
3. Component detectors exist now for a number of semantic objects, such as sky, grass, skin, and faces (in consumer photos). More importantly, they are already or becoming sufficiently reliable.

One interesting direction for future research proposed in [54] is that of using a system based on holistic features synergistically with an object recognition system. Knowledge gained about the scene type can be used to prime the recognizer and objects recognized can be used to improve the scene class hypothesis and confidence.

4.3 Scene Configuration

There appears to be much work remaining to be done in scene modeling and configuration. We were able to find very little research on this approach, perhaps because of the amount of work required; Lipson found that a combination of a number of narrow detectors was needed to cover most of the members of a class, and each was hand-coded [19]. Early work done to learn the models automatically has been promising [36].

One drawback of Lipson’s system is that it must meet all of a category’s configural constraints to be placed in that category; the rules are hard-coded, even if learned. While they do provide flexibility by using relative feature values rather than absolute ones, their performance does not degrade gracefully.

Smith's system allows for more flexibility because the likelihoods he calculates for each configuration, given the class, are calculated from training data. In this way, the likelihoods provide an estimate of the saliency of the configuration. However, his system only looks at configurations along the vertical dimension.

Finally, Lipson states that the technique is not designed to classify scenes that depend on object recognition; however, this not just a limitation of her system, but of the current state of the art, since to our knowledge, no systems have incorporated object detection into semantic scene classification.

4.4 Conclusion

Scene classification is still a relatively young field of research, only gaining popularity and some successes in the late-1990's. We discussed a number of existing systems and synthesized from the research many potential options both for features to be extracted, inference engines that could be used to combine them, and control strategies. We are currently applying these ideas to develop a sunset detector, which as a prototype has already shown good success. Our future work includes extending this system to classify multiple semantic scene types.

Chapter 5

Acknowledgments

Boutell's and Brown's contribution to this research was supported in part by a generous grant from Eastman Kodak Company under its external technology program. Some of the preliminary results were based on work performed at Kodak Research Labs. Their work was supported furthermore by the NSF under grant number EIA-0080124 and by the Department of Education (GAANN) grant number P200A000306. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of Eastman Kodak Company or these agencies.

We would also like to thank Amit Singhal, Bob Gray, Zhao-hui Sun, and Navid Serrano, all of Eastman Kodak Company, for their support and advice throughout the course of this research.

Bibliography

- [1] Dana H Ballard. *An Introduction to Natural Computation*. MIT Press, Cambridge, MA, 1997.
- [2] Dana H Ballard and Christopher M Brown. *Computer Vision*. Prentice-Hall, Englewood Cliffs, NJ, 1982.
- [3] J. Batlle, A Casals, J Freixenet, and J Mart. A review on strategies for recognizing natural objects in colour images of outdoor scenes. *Image and Vision Computing*, 18(6-7):515–530, May 2000.
- [4] Matthew Boutell and Jiebo Luo. Single frame orientation using low-level features. Technical Report 758, University of Rochester, Rochester, NY, September 2001.
- [5] C. Carson, S. Belongie, H Greenspan, and J. Malik. Recognition of images in large databases using a learning framework. Technical Report 97-939, U.C. Berkeley, 1997.
- [6] C. Carson, S. Belongie, H Greenspan, and J. Malik. Region-based image querying. In *Proc. IEEE Workshop on Content-Based Access of Image and Video Libraries*, pages 42–49, 1997.
- [7] C. Carson, S. Thomas, M. Belongie, J.M. Hellerstein, and J. Malik. Blobworld: a system for region-based image indexing and retrieval. In *Third Intl. Conf. on Visual Information Systems*. Springer-Verlag, June 1999.
- [8] Eugene Charniak. Bayesian networks without tears. *AI Magazine*, 12(4):50–63, April 1991.
- [9] D. Crevier and R. Lepage. Knowledge-based image understanding systems: A survey. *Computer Vision and Image Understanding*, 67(2):161–185, August 1997.
- [10] R. Duda, R. Hart, and D. Stork. *Pattern Classification*. John Wiley and Sons, Inc., New York, 2 edition, 2000.
- [11] Myron Flickner, Harpreet S. Sawhney, and et al. Query by image and video content: The qbic system. *IEEE Computer*, 28(9):23–32, 1995.
- [12] Isaac Green. Informal discussion regarding the University of Rochester’s object recognition system, April 2002.

- [13] R. M. Haralick and L. G. Shapiro. *Computer and Robot Vision*, volume 2, chapter 19. Addison-Wesley, Inc., Reading, MA, 1993.
- [14] A. Hauptmann and M. Smith. Text, speech, and vision for video segmentation: The informedia project. In *AAAI Symposium on Computational Models for Integrating Language and Vision*, Fall 1995.
- [15] A.K. Jain, R.P.W. Duin, and J. Mao. Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):4–37, January 2000.
- [16] T. Kohonen. Improved versions of learning vector quantization. In *Proc. International Joint Conference on Neural Networks*, pages 545–550, San Diego, June 1990.
- [17] T. Kohonen, J. Kangas, J. Laaksonen, and K. Torkkola. Lvq_pak: A program package for the correct application of learning vector quantization algorithms. In *Proc. International Joint Conference on Neural Networks*, volume 1, pages 725–730, Baltimore, June 1992.
- [18] P. Lipson. *Context and Configuration-Based Scene Classification*. PhD thesis, MIT, Cambridge, MA, 1996.
- [19] P. Lipson, E. Grimson, and P. Sinha. Configuration based scene classification and image indexing, 1997.
- [20] Y. Lu, C. Hu, X. Zhu, H. J. Zhang, and Q. Yang. A unified framework for semantics and feature based relevance feedback in image retrieval systems. In *ACM Multimedia Conference*, Los Angeles, October 2000.
- [21] J. Luo, A. Singhal, and R. Gray. A human factor study of image orientation determination. In *SPIE International Symposium on Electronic Imaging, 2003*.
- [22] J. Luo and A. Savakis. Indoor vs. outdoor classification of consumer photographs using low-level and semantic features. In *IEEE International Conference on Image Processing*, Thessaloniki, Greece, October 2001.
- [23] Stephane Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–692, July 1989.
- [24] C. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA, 1999.
- [25] Joan Martí, Jordi Freixenet, Joan Batlle, and Alicia Casals. A new approach to outdoor scene description based on learning and top-down segmentation. *Image and Vision Computing*, 19(14):1041–1055, December 2001.
- [26] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *IEEE International Conference on Computer Vision, 2001*.

- [27] M. Mirmehdi, P.L. Palmer, J. Kittler, and H. Dabis. Complex feedback strategies for hypothesis generation and verification. In *Proceedings of the 7th British Machine Vision Conference*, pages 123–132. BMVA Press, September 1996.
- [28] Y. Ohta. *Knowledge-based Interpretation of Outdoor Natural Color Scene*. Pitman Advanced Publishing Program, London, 1985.
- [29] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. In *International Journal of Computer Vision*, volume 42, pages 145–175, 2001.
- [30] S. Paek and S.-F. Chang. A knowledge engineering approach for image classification based on probabilistic reasoning systems. In *IEEE International Conference on Multimedia and Expo. (ICME-2000)*, New York City, NY, Jul 30-Aug 2 2000.
- [31] S. E. Palmer. The effects of contextual scenes on the identification of objects. *Memory and Cognition*, 3:519–526, 1975.
- [32] G. Pass, R. Zabih, and J. Miller. Comparing images using color coherence vectors. In *Proceedings of the 4th ACM International Conference on Multimedia*, pages 65–73, Boston, Massachusetts, November 1996.
- [33] J. Pearl, editor. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann Publishers, 1988.
- [34] Rajesh Rao and Dana Ballard. Efficient encoding of natural time varying images produces oriented space-time receptive fields. Technical Report 97.4, University of Rochester, Rochester, NY, August 1997.
- [35] T. Randen and J.H. Husoy. Filtering for texture classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(4):291–310, April 1999.
- [36] A. Ratan and W.E.L. Grimson. Training templates for scene classification using a few examples. In *Proceedings of IEEE Content Based Access of Image and Video Libraries*, San Juan, 1997.
- [37] R. D. Rimey and C. M. Brown. Control of selective perception using Bayes nets and decision theory. *International Journal of Computer Vision, Special Issue on Active Vision*, 1994.
- [38] Raymond D. Rimey. *Control of Selective Perception using Bayes Nets and Decision Theory*. PhD thesis, Computer Science Dept., U. Rochester, Rochester, NY, December 1993.
- [39] A. Rosenfeld. From image analysis to computer vision: An annotated bibliography, 1955-1979. *Computer Vision and Image Understanding*, 84:298–324, 2001.
- [40] S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, December 22 2000.

- [41] Cordelia Schmid. Constructing models for content-based image retrieval. In *Computer Vision and Pattern Recognition*, Kauai, Hawaii USA, December 2001.
- [42] B. Scholkopf, C. Burges, and A. Smola. *Advances in Kernel Methods: Support Vector Learning*. MIT Press, Cambridge, MA, 1999.
- [43] Andrea Selinger and Randal C. Nelson. A perceptual grouping hierarchy for appearance-based 3d object recognition. *Computer Vision and Image Understanding*, 76(1):83–92, October 1999.
- [44] Navid Serrano, Andreas Savakis, and Jiebo Luo. A computationally efficient approach to indoor/outdoor scene classification. In *ICPR*, (to appear).
- [45] A. Singhal. *Bayesian Evidence Combination for Region Labeling*. PhD thesis, University of Rochester, Rochester, NY, 2001.
- [46] A.W.M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1349–1380, December 2000.
- [47] J. R. Smith and S.-F. Chang. Tools and techniques for color image retrieval. In *Symposium on Electronic Imaging: Science and Technology - Storage and Retrieval for Image and Video Databases IV*, volume 2670, San Jose, CA, February 1996.
- [48] J. R. Smith and C.-S. Li. Image classification and querying using composite region templates. *Journal of Computer Vision and Pattern Recognition*, 75(1/2):165 – 174, July/August 1999.
- [49] Milan Sonka, Vaclav Hlavac, and Roger Boyle. *Image Processing, Analysis, and Machine Vision*. Brooks/Cole Publishing, Pacific Grove, CA, 2 edition, 1999.
- [50] K.K. Sung and T. Poggio. Example-based learning for view-based human face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1):39–51, January 1998.
- [51] M. J. Swain and D. H. Ballard. Color indexing. *International Journal of Computer Vision*, 7(1), 1991.
- [52] Martin Szummer and Rosalind W. Picard. Indoor-outdoor image classification. In *IEEE International Workshop on Content-based Access of Image and Video Databases*, Bombay, India, 1998.
- [53] A. Torralba and P. Sinha. Contextual priming for object detection. Technical Report AI Memo 2001-020, CBCL Memo 205, MIT, September 2001.
- [54] A. Torralba and P. Sinha. Recognizing indoor scenes. Technical Report AI Memo 2001-015, CBCL Memo 202, MIT, July 2001.

- [55] A. Torralba and P. Sinha. Statistical context priming for object detection. In *Proceedings of the International Conference on Computer Vision*, pages 763–770, Vancouver, Canada, 2001.
- [56] A. Vailaya. *Semantic Classification in Image Databases*. PhD thesis, Michigan State University, East Lansing, MI, 2000.
- [57] A. Vailaya. Personal correspondence via email, June and July 2001.
- [58] A. Vailaya, M. Figueiredo, A. Jain, and H.J. Zhang. Content-based hierarchical classification of vacation images. In *Proc. IEEE Multimedia Systems '99 (International Conference on Multimedia Computing and Systems)*, Florence, Italy, June 1999.
- [59] A. Vailaya, A. K. Jain, and H.-J. Zhang. On image classification: City images vs. landscapes. *Pattern Recognition*, 31:1921–1936, December 1998.
- [60] A. Vailaya, H.J. Zhang, and A. Jain. Automatic image orientation detection. In *Proc. IEEE International Conference on Image Processing*, Kobe, Japan, October 1999.
- [61] J. Wang, J. Li, and G Wiederhold. SIMPLIcity: Semantics-sensitive Integrated Matching for Picture Libraries. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(9):947–963, 2001.
- [62] Yongmei Wang and Hongjiang Zhang. Content-based image orientation detection with support vector machines. In *IEEE Workshop on Content-Based Access of Image and Video Libraries (CBAIVL2001)*, Kauai, Hawaii USA, December 14 2001.
- [63] C. Yang. Personal correspondence via email, July 2001.
- [64] C. Yang, F-I Liu, A Vailaya, H.J. Zhang, and A. Jain. Automatic image orientation detection. Michigan State University CSE Department Poster Workshop, April 2000.