Learning statistical features of scene images

Wooyoung Lee

September 2014
CMU-ML-14-103

MACHINE LEARNING
DEPARTMENT

Carnegie Mellon

Learning statistical features of scene images

Wooyoung Lee

September 2014
CMU-ML-14-103

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Michael S. Lewicki
Geoffrey J. Gordon
Yaser A. Sheikh
Bruno Olshausen

*Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy.*

This research was sponsored by the Samsung Foundation, the Office of Naval Research under grant number MURI N000140710747 and by the National Science Foundation under grant number IIS1111654.

Keywords: Visual scene understanding, visual features, probabilistic models, generative models, adaptive representation, feature learning, optimization, perceptual scene layout properties, semantic scene categories, image retrieval

Abstract

Scene perception is a fundamental aspect of vision. Humans are capable of analyzing behaviorally-relevant scene properties such as spatial layouts or scene categories very quickly, even from low resolution versions of scenes. Although humans perform these tasks effortlessly, they are very challenging for machines. Developing methods that well capture the properties of the representation used by the visual system will be useful for building computational models that are more consistent with perception. While it is common to use hand-engineered features that extract information from predefined dimensions, they require careful tuning of parameters and do not generalize well to other tasks or larger datasets. This thesis is driven by the hypothesis that the perceptual representations are adapted to the statistical properties of natural visual scenes.

For developing statistical features for global-scale structures (low spatial frequency information that encompasses entire scenes), I propose to train hierarchical probabilistic models on whole scene images. I first investigate statistical clusters of scene images by training a mixture model under the assumption that each image can be decoded by sparse and independent coefficients. Each cluster discovered by the unsupervised classifier is consistent with the high-level semantic categories (such as indoor, outdoor-natural and outdoor-manmade) as well as perceptual layout properties (mean depth, openness and perspective). To address the limitation of mixture models in their assumptions of a discrete number of underlying clusters, I further investigate a continuous representation for the distributions of whole scenes. The model parameters optimized for natural visual scenes reveal a compact representation that encodes their global-scale structures. I develop a probabilistic similarity measure based on the model and demonstrate its consistency with the perceptual similarities.

Lastly, to learn the representations that better encode the manifold structures in general high-dimensional image space, I develop the image normalization process to find a set of canonical images that anchors the probabilistic distributions around the real data manifolds. The canonical images are employed as the centers of the conditional multivariate Gaussian distributions. This approach allows to learn more detailed structures of the local manifolds resulting in improved representation of the high level properties of scene images.

Acknowledgments

The past several years at Carnegie Mellon University was challenging but very fruitful time of my life. I was fortunate to be surrounded by smart, hard-working and nice people who supported and inspired me everyday.

I would like to first thank my thesis advisor Mike Lewicki for his immeasurable amount of patience and guidance throughout my path. He has always waited until I figure out the directions I want to pursue and encouraged and guided me so that I could advance towards the right directions. He has advised me remotely from Cleveland during most of my training at CMU, but he has never failed to provide me with enough resources, readings to follow, time for discussions and advice. I was fortunate to work with Geoff Gordon and learn diverse statistical and mathematical skills. I want to thank him for bearing with me when I was asking many questions to follow his thought process during the discussions. I learned how to frame research questions more interestingly and draw big pictures which can attract people from Yaser Sheikh. I was also fortunate to have Bruno Olshausen on my thesis committee and have chances to have discussions with him. I was always impressed by his sharp eyes on details and broad and deep understanding ranging from statistical models to neuroscience.

I came to CMU without much background in computer science (at least compared to my peers) but I was able to catch up very efficiently thanks to the amazingly well-designed and well-taught at Carnegie Mellon. I would like to thank Carlos Guestrin for the intro Machine Learning, Graphical Models and Optimization course with Geoff Gordon, David Kosbie for his programming course, Alyosha Efros and Lavanya Sharan for their Learning Based Methods in Vision course and Matthew Harrison for Intermediate Statistics course.

The members of the Computational Perception Laboratory at Carnegie Mellon and the Natural Perception Laboratory at Case Western Reserve University, Yan Karklin, Doru Balcan, Daniel Leeds, Eizaburo Doi and Christina Dimattina, helped me to develop and polish my research ideas. I also want to thank Aude Oliva because my research has been hugely influenced by her work and for providing the dataset she collected with Michael R. Ross for analysis. Tai-Sing Lee has contributed to my thesis by challenging me interesting questions and advice. Without the support from High Performance Computing System at Case Western Reserve University, I would not have been able to run all the simulations and optimizations which are reported in this thesis. I would like to especially thank Sanjaya Gajurel for his prompt and responsible service.

I owe a lot to Junyoung Kwak, who first introduced Carnegie Mellon to me. During the early years of grad school Seung-il Huh, Q Youn Hong, Donghun Lee, Ji

Eun Song, Soo Kyung Sul and Ryan Kelly helped me to settle down well to the life as a PhD student in Pittsburgh. I also want to thank Leman Akoglu, Kriti Puniyani and Gaurav Veda for listening to me when I was in time of need. I would like to mention that without the nagging from the senior students such as Maverick Woo and Doru Balcan, this thesis would have taken much longer to be finished. I would like to thank the best staff of the machine learning department and especially Diane Stidle for her exceptional service. I learned a lot from her professional attitude and her kindness. I was very fortunate to meet friends who were like sisters and role models to me: Sue Ann Hong, Lucia Castellanos, Alona Fyshe, Min Kyung Lee, Jueun Lee, Seyoung Kim and Sung Won Park. The discussions and conversations with the friends I met at SELECT Lab and the machine learning department, Jon Huang, Yucheng Low, Byron Boots, Joseph Gonzales, Joseph Bradely, Khalid El-Arini, Polo Chau, Hai Son Le, Ada Iosif, Yi Zhang, Mary McGlohon and Yang Xu, were always fun and inspirational.

Although they were physically very far away, my old friends, Na Young So, Elizabeth Jaeun Paik, Hana Lee, Taekyung Kim, Yujin Jeon, Ji-Min Lee, NaYoung Lee, Soo Hyun Park, Hyun Ah Lee, Kuwook Cha, Sang-il Kim and Wonyoung Sohn were psychologically close by and supportive of me. I also want to thank my master's advisor Sang Hun Lee for teaching me the joy of research when I was starting the journey with the first step.

Last but not least, I am grateful to my family, my parents, my brother and my sister, and my parents-in-law, for their unconditional love and support. I am blessed to have JinSeok Park as my {biking, hiking, movie, Korean-style karaoke, dining, cooking, workout, billiards, travel} buddy and most importantly as my loving husband. Without his smiles in good times and hugs in difficult times, I would not have been able to come this far.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Understanding the holistic properties of scene images (pictures that depict spaces rather than primarily describing objects in a scene) is a key process for scene perception. Such holistic information gives rise to perceptual spatial layout properties of scene images such as depth, openness, perspective and memorability [20, 27]. In addition, scene images that belong to the same semantic categories tend to have similar global structures [44] suggesting that the global information contributes to semantic properties of scenes.

As scene images are high-dimensional, developing a compact and efficient representation designed for encoding the global properties is challenging. Previous studies have revealed that hand-engineered features [6, 17, 32, 45] are capable of predicting the semantic properties of scene images such as perceptual properties of the spatial layouts [56], categories, memorability [26] and typicality [16]. Although these approaches have been successful, the features require careful hand-tuning of parameters depending on the tasks. This requirement limits the generality of what is learned based on such features in one dataset to others [70].

Another potential disadvantage of projecting scene images onto hand-designed feature spaces is that they do not necessarily capture all relevant scene information. For instance, although scene images have diverse local properties based on their contents (textures and objects within the scenes, etc.), the global structures of scenes are highly constrained in spatial layout and 3D structure. These constraints provide scene images with special regularities on the global scale. Hand-designed representations which do not take these regularities into account are unlikely to deal with the meaningful statistical structures of the scene images (which are potentially relevant to the perceptual properties) [49].

Several algorithms have been developed for encoding the characteristic structures of images. One approach is to build efficient representations that encode images with a small number of coefficients by imposing sparsity constraints [25, 71]. Another method is to learn a representation invariant to translations and rotations [30, 36, 55]. This method uses pooling algorithms that feed the strongest responses of local filters over a fixed range to the higher level representations. Although these methods have been successful for local textures and object recognition, they are not well suited for scene images, which have entirely different properties from textures. Deep learning algorithms are also a hot topic for learning from data [22]. The main difference between our approach and the deep learning is mainly in the fact that deep learning algorithms start from modeling the joint distribution of the data and hidden variables based on Restricted Boltzman

Machines whereas I focus more on learning the conditional distribution of data given hidden variables. Our approach allows to design prior distributions of hidden or latent variables easier than the deep learning algorithms. This flexibility can be useful especially when combined with sparse priors allowing the model parameters to efficiently represent the data [31].

This thesis addresses the challenge of developing scene descriptors that are sensitive to behaviorablly-relevant holistic properties but yet invariant to noisy variations. I propose to train probabilistic models on scene images and use the latent variables as features. This framework allows to automatically learn features that can compactly represent scenes by optimizing the model parameters to be aligned with main directions along which data is distributed. In Chapter 2, I first show that the unsupervised clusters of scene images based solely on their statistical properties are consistent with the semantic and perceptual properties of scene images [39]. Motivated by this result, I train a conditional correlational model on whole scenes to further investigate the holistic properties of scene images based on co-occurrences of local structures (Chapter 3) [40]. The model parameters reveal compact code for encoding global structures. In Chapter 4, I extend the representation to larger scene images by learning the statistical properties of multiscale scene representation. Lastly, in Chapter 5, I introduce an extended probabilistic model for arranging the conditional multivariate Gaussian models closer to the data manifolds.

Chapter 2

Unsupervised categories of scene images

2.1 Introduction

In this chapter, I develop a scene image representation that learns the compact code for representing subgroup of data based on the independent component analysis mixture model (ICAMM) [37, 38] on full scene images. ICAMM trains a mixture of ICA components on a dataset allowing us to learn specialized distributions for subpopulations of a dataset rather than training one set of basis functions to fit the entire dataset. Our approach is distinct from previous studies on scene images, first, in that I learn representations that capture scene image distributions. Second, that the representations instead of being derived for local features from patches are derived for full images.

Learning features from the data at a holistic level was not possible in the past because the number of training examples required is proportional to the dimensionality of the data which is inherently high for scene images. However, a recent study [68] shows that color scene images at the spatial resolution of 32×32 pixels contain sufficient information for identifying scene categories and to identify objects within scene images. In addition, the recent release of a large scene image dataset [75] with a sufficient number of images for training ICA on 32×32 color images has further contributed to making learning adaptive representations of scene images achievable.

2.2 ICA Mixture model

2.2.1 Derivation

I adopt unsupervised classification algorithm derived by modeling data as a mixture of classes that are each described by linear combinations of independent, non-Gaussian densities[38]. This approach learns features for specific classes of the dataset by learning exclusive sets of basis functions for the individual classes. Each data vector $\mathbf{x}$ (a concatenation of vectorized red,blue and green channels of scene images) is generated by a mixture density

$$p(\mathbf{x}|\Theta) = \sum_{k=1}^{K} p(\mathbf{x}|C_k, \theta_k)p(C_k) \tag{2.1}$$

I assume that the number of mixtures, K, and the prior probabilities of the classes, $p(C_k), (k = 1, \cdots K)$, are fixed and the prior probabilities sum up to 1. The data in each class is described by the independent component model.

$$\mathbf{x} = \mathbf{A}_k \mathbf{s}_k + \mathbf{b}_k, k = 1, \cdots, K \tag{2.2}$$

where $\mathbf{A}_k$ is a $N \times M$ matrix (N: the dimensionality of $\mathbf{x}$, M: the number of basis functions) and $\mathbf{b}_k \in \mathcal{R}^N$ is the bias vector for k^{th} class. For simplicity, I assume that $\mathbf{A}_k$ is full rank for each class ($N = M$). To constrain individual elements s_k^i ($i = 1, \cdots, N$) of the coefficient vector $\mathbf{s_k}$ to be independent and sparse, I design the distribution of $\mathbf{s_k}$ as,

$$\log p(\mathbf{s}_k) \propto -\sum_{i=1}^{N} |s_k^i| \tag{2.3}$$

In this setting, the log likelihood of the data for each class is defined as

$$\log p(\mathbf{x}|C_k, \theta_k) = \log p(\mathbf{s}_k) - \log \det |\mathbf{A}_k| \tag{2.4}$$

where $\theta_k = \{\mathbf{A}_k, \mathbf{b}_k\}$. The conditional probability of each class given the data vector $\mathbf{x}$ is

$$p(C_k|\mathbf{x}, \theta_k) = \frac{p(\mathbf{x}|C_k, \theta_k)p(C_k)}{\sum_k p(\mathbf{x}|C_k, \theta_k)p(C_k)} \tag{2.5}$$

A data vector $\mathbf{x}$ is classified into k^{th} class that has the maximum value of $p(C_k|\mathbf{x}, \theta_k), (k = 1, \cdots, K)$. This is why I also refer to this model as the ICA classifier.

$\mathbf{A}_k$ is adapted using the gradient ascent method,

$$\begin{aligned} \Delta A_k &\propto \frac{\delta \log p(\mathbf{x}|\theta_k)}{\delta A_k} \\ &= p(C_k|\mathbf{x}, \theta_k)\frac{\delta \log p(\mathbf{x}|C_k, \theta_k)}{\delta A_k} \end{aligned} \tag{2.6}$$

Each iteration, I sample a batch of data vectors and average ΔA_k over the multiple data vectors. After updating $\mathbf{A}_k$ in each iteration with Eqn 2.6, and then calculating $p(C_k|\mathbf{x}_t, \theta_{k=1:K})$ with the updated $\mathbf{A}_k$, the bias vector $\mathbf{b_k}$ is updated as

$$\mathbf{b_k} = \frac{\sum_t \mathbf{x}_t p(C_k|\mathbf{x}_t, \theta_k)}{\sum_t p(C_k|\mathbf{x}_t, \theta_k)} \tag{2.7}$$

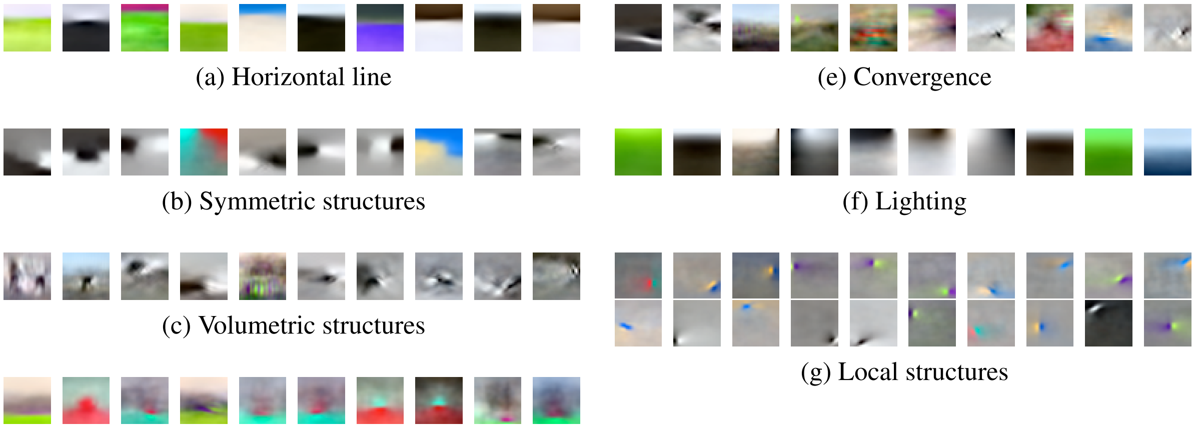
2.2.2 Model training

I first downsampled the scene images in the dataset to 32×32 color images. As most of the images have longer width (height) than height (width), I sampled a few squares along the longer axis with the square window to increase the number of samples in the dataset and then downsampled the samples. Before the training, I centered the dataset by subtracing the mean of the dataset from each image and then whitened the dataset[25]. I initialized $\mathbf{A}_{k=1:K}$ to the identity matrix and $\mathbf{b}_{k=1:K}$ to random. Each iteration, I sampled batches of images and calculated the gradient by Eqn.2.6 and updated $\mathbf{A}_k$. I terminated the training procedure when the likelihood of the model does not increase significantly.

2.3 Properties of the basis functions

2.3.1 The emergence of global structures

Increasing the number of classes in ICAMM allows us to learn specialized distributions for subpopulations of scene images. Thus, the basis functions that emerge as I increase the number of classes unveil features for specific representations of the dataset. When I train ICAMM with $K = 1$ on scene images (which is identical to standard ICA except for the bias term), most of the basis functions have local structures which are similar to the basis functions trained on local natural textures [72], as shown in Figure 2.1g. As I increase the number of classes, K, from one to four, the structures of the basis functions become more diverse. While some basis functions maintain local structures, basis functions with global structures also arise. Figures 2.1a–2.1f show sample basis functions that encode the following structures of scene images: horizontal lines, symmetric structures, volumetric structures, volumetric structures on the ground, convergence, and light gradients. Note that the global features learned by the model begin to take on forms that resemble physical properties of the scenes. As increasing the number of classes further is computationally more expensive and the evolution of the basis function properties as I increase the number of classes suggest that hierarchical representations would be able to capture the subcategories better than the single layered representation with larger number of classes, I limit the number of classes to four in this paper.


(a) Horizontal line

(e) Convergence

(b) Symmetric structures

(f) Lighting

(c) Volumetric structures

(g) Local structures

(d) Volumetric structures on the ground

Figure 2.1: (a)–(f) Basis functions with global structures emerge when I train ICAMM with $K = 4$. (g) Basis functions with local structures are dominant when I train ICAMM with $K = 1$.

2.3.2 Types of basis functions

With regard to color, 98% of the basis functions can be organized into one of four types; gray channels (Gray, Figure 2.2b), yellow and blue channels (YB, Figure 2.2c), red and cyan channels (RC, Figure 2.2d) and purple and green channels (PG, Figure 2.2e). The percentages of Gray and YB are quite similar in the four sets of basis functions; 44.8%–45.4% for Gray and 30.7%–31.0%

for YB. However, RC appears only in the basis functions of Classes 1 and 2 (23.7% and 24.2%) and PG appears only in the basis functions of Classes 3 and 4 (23.4% and 17.4%) (Figure 2.2a). This result implies that natural clusters of images arise depending on their color components. I will discuss how the color components of each class are related to the super-ordinate scene categories discussed in section 2.4.1.

In terms of spatial configurations (Figure 2.2f), the basis functions can be categorized by the spatial scales (global vs. local) and the dominant orientations of the structures (horizontal, vertical, oblique and complex when more than one orientation of the structures are dominant) (Figure 2.2g–2.2n). Figure 2.2f shows the fractions of the spatial configuration types (spatial scale× dominant orientation) in each class. The basis functions corresponding to Class 4 contain the highest percentages of global basis functions (81.2%) followed by Class 3 (61.0%), Class 2 (40.4%) and finally Class 1 (31.0%). In section 2.4.2, I will further discuss how the percentages of the global and local basis functions are related to the spatial layout properties of scene images.

2.4 Unsupervised scene classification

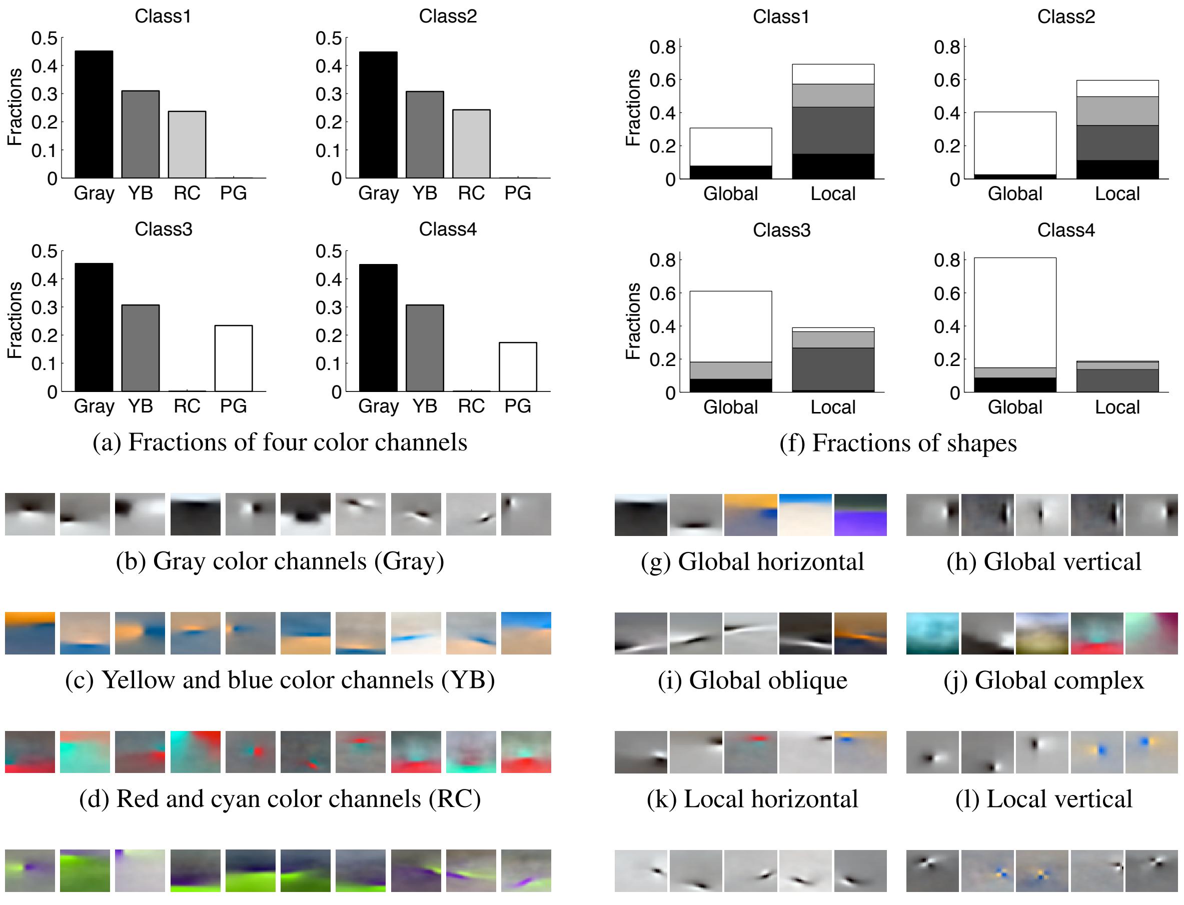
Given a scene image $\mathbf{x}$, ICAMM computes $p(C_k|\mathbf{x}, \theta_{1:K})$ classifying $\mathbf{x}$ into the k^{th} class with the maximum value of $p(C_k|\mathbf{x}, \theta_k)$. This unsupervised scene classification reveals natural categories of scene images based solely on their statistics and not on their semantic labels. In this section, I analyze the super-ordinate scene category and perceptual scene layout rating distributions among scene images categorized into different classes by ICAMM. I use the model parameters that I estimated when training the model on the SUN database.

2.4.1 Super-ordinate scene categories

The majority of the indoor scenes from the SUN database are categorized into Classes 1 and 2 while most of the outdoor-natural scenes are categorized into Classes 3 and 4. Outdoor-manmade scenes, which have both artificial and natural components, are evenly distributed among the Classes 1–4 (Figure 2.3a). Note that the dominance of indoor scenes in Classes 1 and 2 and outdoor-natural scenes in Classes 3 and 4 is consistent with the distribution of color channels among the basis functions that correspond to the four classes (Section 2.3.2). This result suggests that while the gray channels and the yellow and blue channels encode common components between indoor and natural scenes, the red and cyan channels and the purple and green channels are specialized to encode artificial components that are prevalent in indoor scenes and natural components that are dominant in natural scenes, respectively.

2.4.2 Perceptual properties of spatial layout

Ross and Oliva [56] collected perceptual ratings of mean depth, openness and perspective on a continuous 1 to 6 scale for 7,138 images. The analysis of the relation between the perceptual scene layout properties and the spatial configurations of the basis functions suggest that both the global and local structures learned from the scene images are necessary for encoding the spatial



(a) Fractions of four color channels

(f) Fractions of shapes

(b) Gray color channels (Gray)

(g) Global horizontal

(h) Global vertical

(c) Yellow and blue color channels (YB)

(i) Global oblique

(j) Global complex

(d) Red and cyan color channels (RC)

(k) Local horizontal

(l) Local vertical

(e) Purple and green color channels (PB)

(m) Local oblique

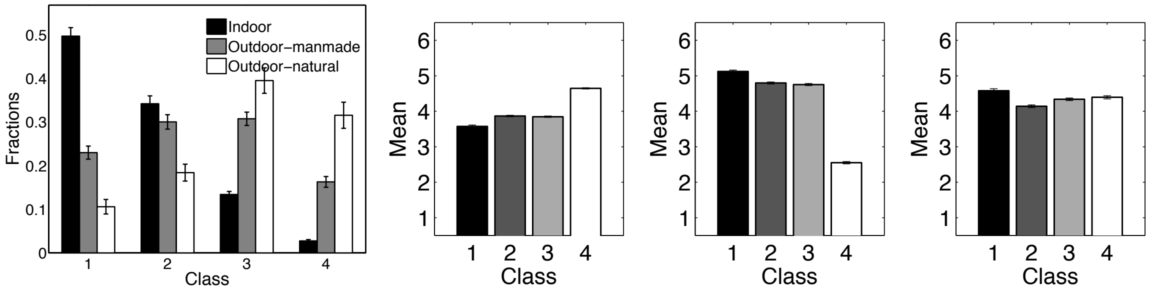
(n) Local complex

Figure 2.2: (a) Gray and YB channels appear in all of the sets of the basis functions while RC channels are found only in Class 1 and 2 and PG in Class 3 and 4. (b)–(e) Sample basis functions of each color channels. (f) The fractions of local basis functions decrease from Class 1 to 4 while those of the global basis functions increase. Black, dark gray, light gray and white each correspond to horizontal, vertical, oblique and complex orientations. (g)–(j) Sample basis functions of global shapes. (k)–(n) Sample basis functions of local shapes.

layout properties of the scenes. The average of mean depth ratings and openness ratings increase and decrease, respectively, from Classes 1 through 4; two-sample t-tests show significant results (p<0.01) except for that between the openness ratings of Classes 2 and 3.

2.5 Conclusion

Training ICAMM on color scene images reveals the basis functions that have diverse structures directly learned from scene images in order to compactly represent them. The distribution patterns of color channels and spatial configurations of basis functions in different classes imply that the natural categories of scene images based on their inherent statistical properties are derived from the content (natural and artificial components) and the spatial structures of scene images.



(a) Super-ordinate categories (b) Depth (c) Openness (d) Perspective

Figure 2.3: (a) Distributions of indoor, manmade-outdoor and outdoor-natural scenes in Classes 1–4. All pairwise t-tests are significant ($p<0.01$) except for those between the fractions of the indoor and manmade-outdoor scenes in Class 2, the fractions of manmade-outdoor scenes in Classes 2 and 3 and the fractions of outdoor-natural scenes in Classes 3 and 4. (b)–(d) The average and the standard error of spatial layout property ratings of scene images classified into Classes 1–4.

Our results suggest that investigating natural categories of scene images might be beneficial for organizing large scene image databases compactly in contrast to current scene categories defined based on linguistic labels associated with scene images. In addition, the rich structures that preserve the statistical properties of the scene images can be useful for complex scene understanding models than scene categorizations which encode and predict high level semantics (the meaning and functions) of scenes.

Chapter 3

Learning global properties of scenes using conditional correlational structure

3.1 Introduction

For learning regularities of scene images, one interesting objective would be to encode the co-occurrences of local structures on global scales. For instance, horizontal lines, which are prevalently observed structures in scene images, are composed of horizontal structures over space around similar vertical locations. A model which can encode such prevalent global structures based on the co-occurrences of local structures would be able to represent global regularities of scene images. To learn a representation which is more adequate for the purpose of learning the global structures of the scene images, I train a hierarchical probabilistic model (which will be referred to hereafter as the distribution coding model) that infers the correlational structures of the distributions from which specific types of scenes are drawn [29]. The distribution coding model compactly represents the space of covariance matrices that best capture correlational structure of the scene mages. Since the model encodes a scene image based on its distribution but not its pixel values, it is invariant to image variability that is not aligned with the statistical regularities of scene images.

3.2 Model training

3.2.1 Model description

To learn the global structures captured by the correlational relationships over space, I trained the distribution coding model [29] on whole scene images. The distribution coding model assumes that data, $\mathbf{x}$, e.g., vectorized scene images in our setting, follows a conditional multivariate gaussian distribution,

$$\mathbf{x}|\mathbf{y} \sim \mathcal{N}(0, \mathbf{C}(\mathbf{y})) \tag{3.1}$$

The zero mean assumption is valid because averaging a sufficient number of scene images shows that the pixel values of the mean scene image have almost uniform values. To satisfy the positive definiteness constraint on covariance matrices, the model formulates the logarithm of the covariance matrices as a function of the latent variable $\mathbf{y}$ as below,

$$\log(\mathbf{C}(\mathbf{y})) = \sum_j y_j \mathbf{A}_j = \sum_j y_j \sum_k w_{j,k} \mathbf{b}_k \mathbf{b}_k^T \tag{3.2}$$

where y_j corresponds to the j th element of $\mathbf{y}$. With this formulation, the distribution coding model is capable of defining a continuum of covariance matrices that are defined by the continuous latent variables $\mathbf{y}$. Note that the model encodes $\mathbf{x}$ in terms of its distribution unlike other scene descriptors. This approach makes the representation robust to noise which is not relevant to the regularities present in the scene images.

Since $\mathbf{A}_j$ is symmetric, the distribution coding model formulates it as the weighted sum of the outer products of vectors $\mathbf{b}_k$s whose dimensionality is identical to that of the data. Each $\mathbf{b}_k$ corresponds to a direction along which the covariance matrices can vary. Rather than learning separate sets of $\mathbf{b}_k, (k = 1, \cdots, K)$ for each $\mathbf{A}_j$, the model lets them share the common dictionary of $\mathbf{b}_k$s and incorporate coefficients $w_{j,k}$ to reduce the dimensionality of the parameters; $\mathbf{A}_j$ with a high value of $w_{j,k}$ strongly encodes the correlational structures present in $\mathbf{b}_k$. On the other hand, a low value of $w_{j,k}$ corresponds to a suppressed variability along $\mathbf{b}_k$. I constrain $\mathbf{b}_k$ and $\mathbf{w}_j = \{w_{j,1}, \cdots, w_{j,K}\}$ on the unit norm ball to prevent degenerate solutions [1].

To enforce the model parameters to learn a compact representation of covariance matrices, the model uses a laplacian prior on $\mathbf{y}$,

$$\log p(\mathbf{y}) \propto -\sum_j |y_j| \tag{3.3}$$

For each sample $\mathbf{x}$, the model infers $\mathbf{y}$ that maximizes its likelihood (Eq. 3.4). Since the original likelihood function is intractable, the integral is approximated by the volume under the maximum joint probability of $\mathbf{x}$ and $\hat{\mathbf{y}}$, $p(\mathbf{x}, \hat{\mathbf{y}})$, where $\hat{\mathbf{y}} =$ arg max $p(\mathbf{x}, \mathbf{y})$:

$$p(\mathbf{x}|\theta) = \int_{-\infty}^{\infty} p(\mathbf{x}|\mathbf{y})p(\mathbf{y})\mathrm{d}\mathbf{y} = p(\mathbf{x}|\hat{\mathbf{y}})p(\hat{\mathbf{y}}) \tag{3.4}$$

To avoid computing the eigen decomposition necessary for the matrix exponential, I use Taylor series approximation for computing the likelihood and the gradients.

3.2.2 Learning and inference

To deal with the large size of the dataset required for estimating the high dimensional parameters, I trained the model with the minibatch training method [33]. In each iteration, I randomly sample a subset of the training data (a minibatch) on which I run the inference and learning steps until convergence. I move on to the next minibatch starting from the current estimate of the model parameters, but discard the gradients and the approximate second-order information obtained from the previous minibatch. One benefit of this approach is that when new training samples are introduced, one does not need to train the model parameters from scratch but rather only update the model parameters on the new samples [76]. The size of a minibatch ranged from 650 to 900 in the experiments reported in this paper.

The model parameters $\Theta = \{\mathbf{b}_k, \mathbf{w}_j\}$ were initialized randomly and optimized using the maximum likelihood method. I first infer latent variables for each data point in the minibatch by maximizing Eq. 3.4 with respect to $\mathbf{y}$ holding $\mathbf{b}_k$'s and $\mathbf{w}_j$'s fix. Then, with the latent variables fixed, I learn the model parameters: I first maximize the likelihood of the data with respect to $\mathbf{b}_k$s with $\mathbf{w}_j$s fixed and then update $\mathbf{w}_j$ with $\mathbf{b}_k$s fixed. To ensure the model parameters have converged with respect to each minibatch, I iterate the process three times per minibatch. In most of the cases, after three iterations of inference and learning steps, the values of the objective function (negative of log likelihood) saturates.

Once the training process is completed, I can use the model parameters $\mathbf{b}_k$'s and $\mathbf{w}_j$'s to infer the latent variables for new scene images. I do so by using the same procedure that I used in the inference step in the training process. Latent variables are initialized to small random values drawn from the Laplacian distribution and updated to maximize the likelihood of a new sample.

The number of $\mathbf{b}_k$'s and the number of $\mathbf{w}_j$'s, K and J, are fixed beforehand (here, K = 450 and J = 25 for all layers). Note that manipulating the two parameters for DCM is not comparable to tuning the parameters for hand-engineered features such as GIST and HOG, but rather more analogous to choosing the number of principal components in PCA. As these parameters increase, the model accounts for the noisier part of the distribution and thus the sensitivity of the model tends to saturate after high enough values.

3.2.3 Fixed norm constraint and line search

Without constraints, the norms of $\mathbf{w}_j$ and $\mathbf{b}_k$ may increase to infinity. Even if they do not, the range of $\mathbf{b}_k$ norms can be considerably wide which results in ambiguous interpretation of the impact of $w_{j,k}$; if $||\mathbf{b}_1|| \ll ||\mathbf{b}_2||$, then, even if their associated weights $w_{1,j}$ and $w_{2,j}$ are the same, this does not imply that the covariance unit A_j is stretched by the same amount along the two directions. Similarly, if $||w_1|| \ll ||w_2||$, then $y_1 = y_2$ does not imply that the covariance matrix $C(\mathbf{y})$ (Eq. 5.2) is stretched along A_1 and A_2 by the same degree.

To prevent degenerate solutions, I constrained the L_2-norm of $\mathbf{b}_k$ and $\mathbf{w}_j$ $k = 1, \cdots, K$,$j = 1, \cdots, J$ to have fixed norms. In each iteration t, given the current parameter estimate x_t, a step size α and a descent direction γ_t, I project $x_t + \alpha\gamma_t$ onto the sphere that satisfies the fixed norm constraint by the retraction function

$$R_{x_t}(\alpha\gamma_t) = c\frac{x_t + \alpha\gamma_t}{||x_t + \alpha\gamma_t||_2} \tag{3.5}$$

where c is a pre-defined sphere radius [1].

I evaluate the objective function (negative of the likelihood in Eq. 3.4) on $R_{x_t}(\alpha\gamma_t)$ and select a stepsize α for which the decrease in the objective function value $f(x_t) - f(R_{x_t}(\alpha\gamma_t))$ satisfies the line search criterion.

3.2.4 Optimization methods

The original implementation of DCM [29] employed the stochastic gradient descent (SGD) method for learning and inference. While SGD is easy to implement, the method requires careful tuning of the learning parameters such as step sizes. The manual tuning of parameters is especially difficult when there are more than one parameter to optimize (hence more than one step size) and the objective function is of higher order. To stabilize and facilitate the training procedure, I tested the conjugate gradient (CG) method [19] and the limited memory BFGS (L-BFGS) method [42]. CG has been developed for efficient search of the solutions by enforcing the search directions to be orthogonal to each other. L-BFGS approximates the second-order information by storing the history of the first-order information for a fixed number of iterations. Therefore it converges faster with greater stability than SGD. The two methods employ line search as a subroutine, and thus there is no need to hand-tune step sizes.

For efficient line search, I estimate initial step lengths by assessing the objective function values at preset step lengths and picking the one that returns the minimum objective function value. Adaptively setting initial step sizes helps especially because different minibatches result in different optimization landscapes and also since I were approximating the gradients and the objective function values to avoid computing matrix exponentials.

Table 3.1 shows the average elapsed time per minibatch and the average decrease in the objective function per minibatch. I fixed the step size for SGD beforehand. CG and L-BFGS take longer to converge on each minibatch but also result in larger decrease in the objective function. Although SGD is faster to run on each minibatch, it converges to different local minima from CG or L-BFGS. Therefore running SGD for longer time does not guarantee that it would reach as good solutions as CG or L-BFGS. The results I report in this paper are obtained by the L-BFGS method which is a good compromise between the computation time and the decrease in the objective function among the three techniques tested. Also, the model parameters obtained by the L-BFGS method are robust; the optimization is insensitive to starting points.

	Condition	Optimization		
		SGD	CG	LBFGS
$\mathbf{b}_k$	Elapsed time (/minibatch)	0.04	115.85	28.32
	Number of updates	1.00	8.89	5.92
	Elapsed time (/update)	0.04	7.89	4.82
	Δ Objective function	3.10	10.01	10.89
$\mathbf{w}_j$	Elapsed time (/minibatch)	0.03	38.05	42.52
	Number of updates	1.00	4.24	4.00
	Elapsed time (/update)	0.03	7.58	12.07
	Δ Objective function	0.19	8.02	7.88

Table 3.1: Elapsed time in seconds. Elapsed time per minibatch varies between optimization techniques since the number of update per minibatch and also the elapsed time per update vary. The results were obtained on a GPGPU Tesla M2070 GPU and the dimensionality of $\mathbf{b}_k$ and $\mathbf{w}_j$ were 186 and 450, respectively, and the number of units K and J were fixed to 450 and 25.

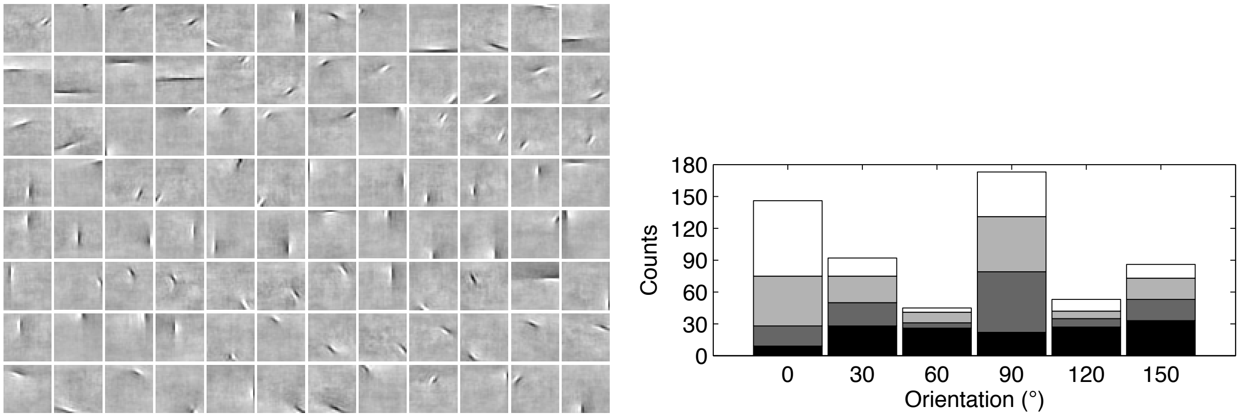
3.2.5 Training data and preprocessing

I trained the distribution coding model on 130,519 scene images (from 397 scene categories) in the SUN database [75]. The dataset is hierarchically organized and covers wide varieties of scene images with diverse structures. Due to the technical constraints such as the number of training examples required for avoiding overfitting and the computational cost, I downsampled the original scene images to 32×32 grayscale images suitable for performance of object detection and scene categorization tasks by human subjects [68]. Because the dataset has enough number of scene images compared to the dimensionality of the model parameters, it is unlikely that the results are overfitted to the training data. This is demonstrated when I apply the model parameters trained on the SUN database to other scene image datasets [32, 56] and scene images downloaded from the web, as the latent variables have similar properties.

3.3 Model representation

3.3.1 Model parameters

As discussed in Section 3.2.1, $\mathbf{b}_k$ encodes a common direction along which the covariance units $\mathbf{A}_j$ can vary. When trained on the 32×32 scene images, $\mathbf{b}_k$s show gabor-like structures as shown in Figure 4.2. Note that the formulation of the model did not constrain $\mathbf{b}_k$s to have localized structures; rather, the structures emerged while fitting the parameters to the scene image statistics. If I generate sample images using a multivariate Gaussian distribution with the covariance matrix $\exp(\mathbf{b}_k \mathbf{b}_k^T)$, the pixels located at the same positions as the elements of $\mathbf{b}_k$ which have the same signs will be correlated in the generated samples. On the other hand, if two elements of $\mathbf{b}_k$ have opposite signs, then the pixel values found at the same location with theses elements in the generated samples will be anti-correlated.

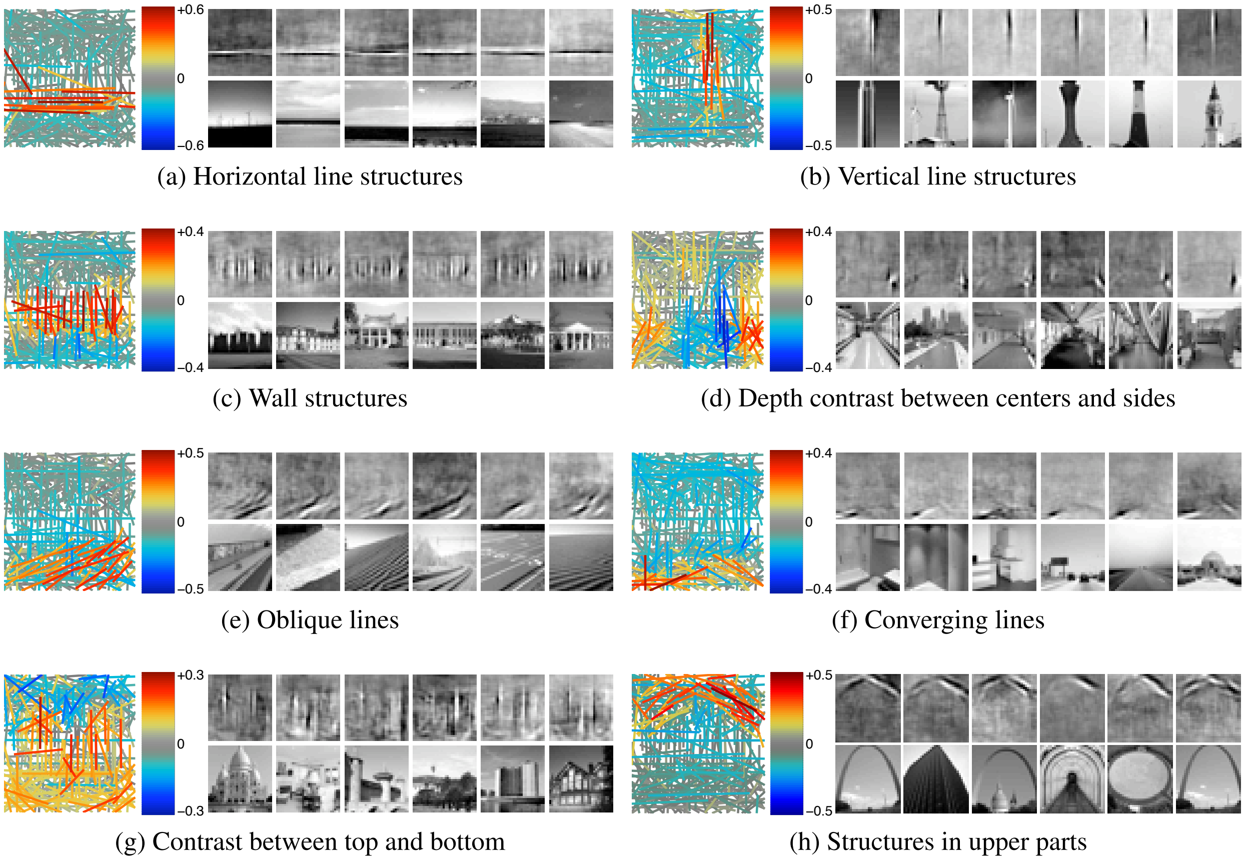


(a) A subset of $\mathbf{b}_k$s

(b) Stacked histogram of orientations and scales of $\mathbf{b}_k$s

Figure 3.1: (a) 96 out of 576 randomly selected are shown. To visualize $\mathbf{b}_k$s which are vectors, I rearrange their elements into 32×32 matrix form. (b) The stacked histogram describing the orientation and scale of $\mathbf{b}_k$s. 0° corresponds to the horizontal orientation, 90° to the vertical orientation. The $\mathbf{b}_k$s are sorted from the most localized to the most global.The black, dark gray, light gray and white parts of the bar graph correspond respectively to the group of the top 25% localized structures, the groups of top 25–50% and 50–75% localized $\mathbf{b}_k$s and the group of the most global $\mathbf{b}_k$s.

When I categorize $\mathbf{b}_k$s based on their orientation and scale, the horizontal and vertical orientations are dominant in light of the external physical structures. In terms of scale, horizontal units, compared to other orientations, have a greater portion of the most global scales (Figure 3.1b). The non-isotropic distribution of scale and orientation of $\mathbf{b}_k$s, the common directions along which the covariance units $\mathbf{A}_j$s can vary, suggests the density component model invests more resources for prevalent visual structures in scene images. This contrasts with most hand-designed visual features in that they tend to allocate uniform bits of information for all orientations and scales.



(a) Horizontal line structures

(b) Vertical line structures

(c) Wall structures

(d) Depth contrast between centers and sides

(e) Oblique lines

(f) Converging lines

(g) Contrast between top and bottom

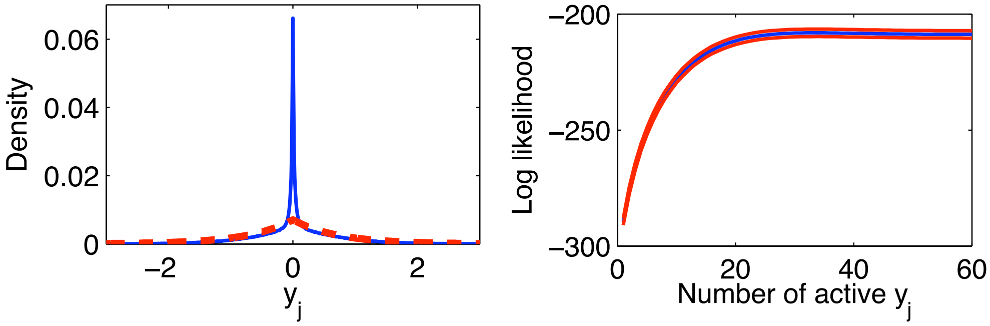
(h) Structures in upper parts

Figure 3.2: (a)–(h) Representative $\mathbf{A}_j$ on the left with corresponding color bars. The red corresponds to positive values of $w_{j,k}$ while blue represents the negative values. On the right, top rows show images generated from multivariate Gaussian distributions with $\exp(y_j \mathbf{A}_j)$ as covariance matrices ($y_j > 0$). The bottom rows show scene images from the SUN database which have the highest values of $\hat{y}_j$.

While $\mathbf{b}_k$s showed localized properties, I find that $\mathbf{w}_j$s encode global information by incorporating the localized correlational structures encoded in the $\mathbf{b}_k$s over space. To visualize each $\mathbf{w}_j$, I first assign a bar to each $\mathbf{b}_k$ which has the same location and orientation with that $\mathbf{b}_k$ in the image space. I then assign each bar a color value corresponding to the value of $w_{j,k}$. I show eight out of sixty $\mathbf{w}_j$s, equivalent to the $\mathbf{A}_j$ (Eq.5.2) in Figure 3.2; these $\mathbf{w}_j$ reveal horizontal and vertical line structures (Fig. 4.4a–4.4b), wall structures (Fig. 4.6a), depth contrasts between centers and sides (Fig. 4.6a), oblique lines (Fig. 3.2e), converging lines (Fig. 3.2f), contrasts between top and bottom (Fig. 3.2g) and structures in upper part of images (Fig. 3.2h). I demonstrate the global correlational structures encoded by $\mathbf{w}_j$ by generating random samples from a multivariate Gaussian distribution whose covariance matrices is $\exp(y_j \mathbf{A}_j)$ $(y_j > 0)$. The generated samples show visually similar structures as the corresponding covariance matrices. In addition, scene images which have the highest values of y_j among the SUN database contain visual structures that resemble the visualization of correlational structures encoded in $\mathbf{A}_j$.

3.3.2 Latent variables

Due to the sparsity constraint on the latent variables (Eq.5.3), the distribution of latent variables $\hat{y}$ peaks around zero (Figure 3.3a). Even though there exist 60 covariance units ($\mathbf{A}_j$), only approximately 20 units are necessary for capturing the correlational structures of a scene image (Figure 3.3b); when I order the elements of the latent variable $\hat{y}$ of a scene image $\mathbf{x}$ according to their magnitudes, and maintain the values of the most active elements, while setting others to zero to compute the likelihood of $\mathbf{x}$, the log likelihood is saturated when I use 20 most active units. Note that this number corresponds to only less than 2% of the original dimensionality of 32×32 grayscale images.



(a) Distribution of $\hat{y}_j$

(b) Log likelihood

Figure 3.3: (a) Distribution of values of $\hat{y}_j$ for scene images in the SUN database (blue solid line) and the constraint I imposed on y_j (Eq.5.3, red dashed line). (b) The log likelihood computed using the most active $\hat{y}_j$s. The x-axis corresponds to the number of most active units used (60 indicates using the original $\hat{y}$), while the y-axis corresponds to the log likelihood of the data computed using the most active $\hat{y}_j$s. The blue line corresponds to the mean over the SUN database and the red lines are the error bars.

When I visualize the covariance matrices determined by the latent variables, they are visually similar to the salient visual features of the corresponding scene images (Figure 3.4). For each sample scene image, I order its latent variables $\hat{y} = \{\hat{y}_1, \cdots, \hat{y}_J\}$ based on their magnitudes. I show the logarithms of the cumulative covariance matrices, $\sum_{i=1}^{k} \hat{y}_{I(i)} \mathbf{A}_{I(i)}$, in the first rows; I corresponds to the order of $\hat{y}_j$s based on the absolute values in the descending order. The positive and negative components of $\hat{y}_{I(k)} \mathbf{A}_{I(k)}$ are separately displayed in the second and the third rows separately for visual clarity. The second column corresponds to $k = 1$ and the right-most column corresponds to $k = 6$. Consistent with the sparse distribution of $\hat{y}$, the first few elements of the $\hat{y}_j$ encode the salient global structures of scene images.

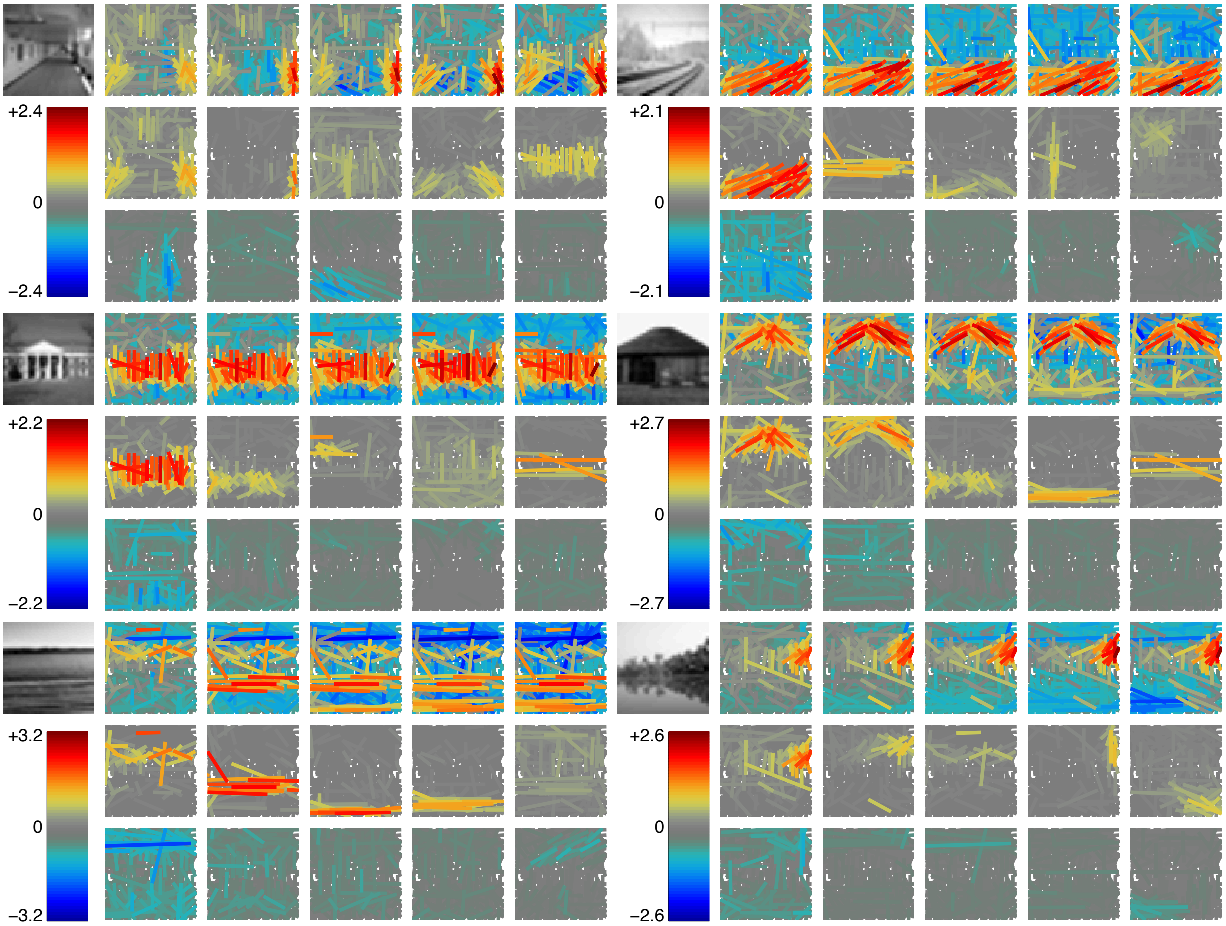


Figure 3.4: For each target image $\mathbf{x}$, I infer its latent variable $\hat{y}$ (Section 3.2.2) and order the $\hat{y}_j$s according to their absolute values. The first rows show the cumulative sum of the logarithm of the covariance matrix using the k most active $\hat{y}_j$s. The second and the third rows show the positive and negative parts of $\hat{y}_{I(k)}\mathbf{A}_{I(k)}$, respectively. I refers to the order of $\hat{y}_j$s based on their magnitudes. This figure is best viewed in color.

I can also analyze the covariance matrices that best describes corresponding scene images by spectral analysis. The spectral analysis reveals the directions along which the covariance matrices are expanded or contracted. In Figure 3.5, I visualize the eigenvectors of the covariance matrices. Note that the eigenvectors corresponding to the positive values of eigenvalues have similar global structures to the scene images. Also, the structures encoded in the eigenvectors corresponding to the negative values of the eigenvalues are absent in the corresponding scene images. Consistent with the previous analysis, this results suggest that the directions along which the covariance matrices are extracted encode the global structures of scene images.

Lastly, I show randomly generated samples drawn from multivariate Gaussian distributions with covariance matrices parameterized by latent variables corresponding to target scenes, respectively (Figure 3.6). It is interesting to note that the generated samples only preserve global structures corresponding to low frequency information. Note that the generated samples, however, do not preserve the edges present in the original images and suggests that the covariance structures in images do not necessarily preserve contours.

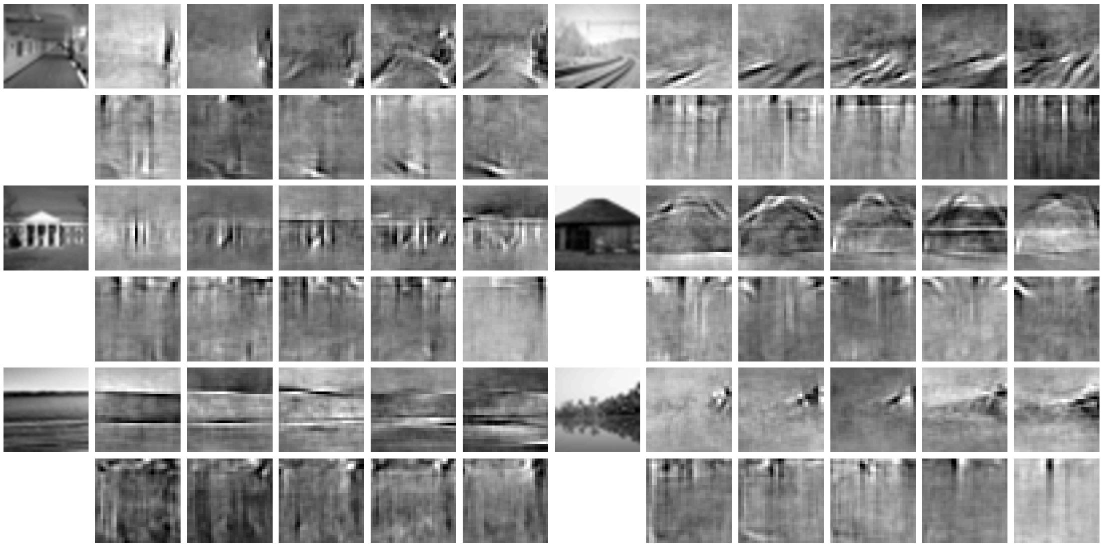


Figure 3.5: For each target image $\mathbf{x}$, I show eigenvectors corresponding to the positive values (upper row) and the negative values (lower row) of the eigenvalues.

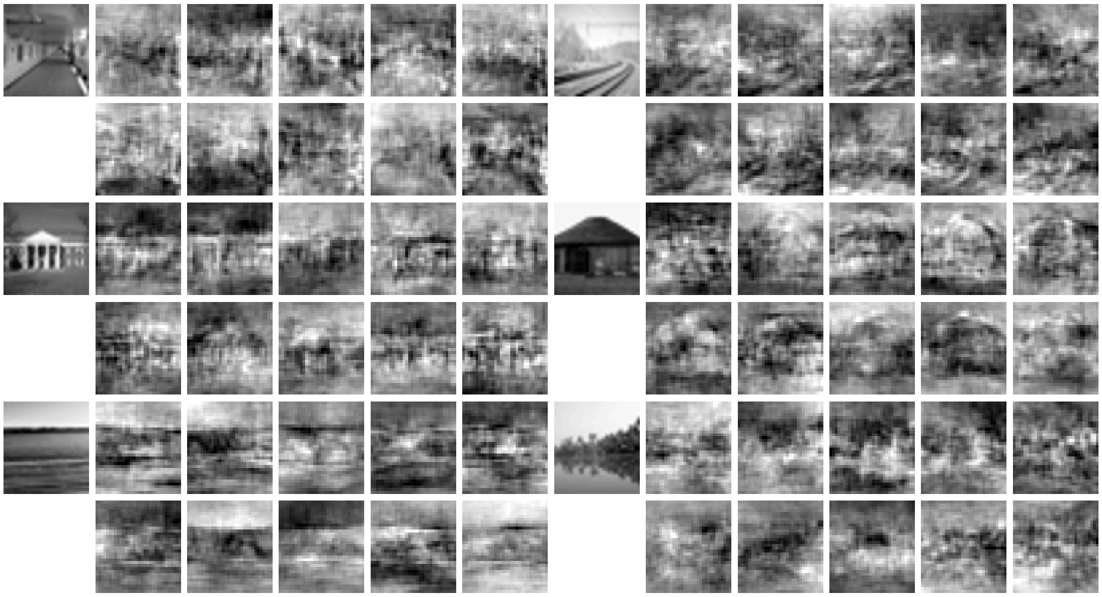


Figure 3.6: Generated random samples from multivariate Gaussian distributions with the covariance matrices parameterized by latent variables corresponding to original images.

3.4 Similarity measure based on the distribution coding model

In the previous section, I showed that the latent variables $\hat{y}$ capture the correlational information which is consistent with the visual structures of scene images and that this representation is efficient in that it requires only a small number of variables to encode the salient properties of scene images. In this section, I discuss how I can utilize the correlational structures encoded in the latent variables as a scene similarity measure and show image retrieval results based on it.

Once I train the distribution coding model and infer the latent variables for scene images, I can develop a metric for measuring the scene similarities in terms of correlational structures using the joint probability of a target scene image $\mathbf{x}_t$ and a latent variable $\hat{y}_c$ of a candidate scene image $\mathbf{x}_c$,

$$p(\mathbf{x}_t, \hat{y}_c) \propto p(\mathbf{x}_t|\hat{y}_c)p(\hat{y}_c) \tag{3.6}$$

The metric consists of two terms; the first term indicates the level of similarity between a target scene image and a candidate scene image in terms of correlational structures. If two data points, $\mathbf{x}_t$ (a *target* point) and $\mathbf{x}_c$ (a *candidate* point), have similar correlational structures, then $\mathbf{x}_t$ will be highly likely under the multivariate Gaussian distribution with the covariance matrix determined by the latent variable for $\mathbf{x}_c$; thus the conditional probability of $\mathbf{x}_t$ given $\hat{y}_c$, $p(\mathbf{x}_t|\hat{y}_c)$ (Eq. 3.1), will be high. Consider the two dimensional example illustrated in Figure 3.7. In the figure, the ovals represent the distributions (characterized by the conditional covariance matrix) that best explains the corresponding data points under the model.

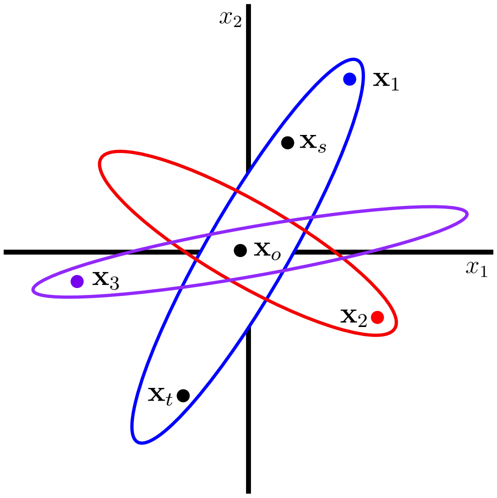


Figure 3.7: Schematic representation of conditional normal distributions $p(\mathbf{x}_t|\hat{y})$. The blue and red ovals represent the conditional covariance matrix $\mathbf{C}(\hat{y}_1)$ (Eq.5.3) and $\mathbf{C}(\hat{y}_2)$ each, where $\hat{y}_i$ indicates the latent variable optimized for $\mathbf{x}_i$. The red oval represents the conditional normal distribution that captures anti-correlated x_1 and x_2. Under the conditional normal distribution represented by this covariance matrix, $\mathbf{x}_t$ will have low likelihood. The purple oval optimized for $\mathbf{x}_3$ encodes positive correlations between x_1 and x_2, but to a different degree from the blue oval. Thus, $\mathbf{x}_t$ will have low conditional probability under the distribution optimized for $\mathbf{x}_3$.

The two data points $\mathbf{x}_t$ and $\mathbf{x}_s$ show similar correlational structures to $\mathbf{x}_1$. Thus, $\mathbf{x}_t$ and $\mathbf{x}_s$ are well captured by the covariance matrix defined by $\hat{y}_1$, which is the latent variable for $\mathbf{x}_1$. On the other hand, covariance matrices which are optimized for data points with different correlational structures from those of $\mathbf{x}_t$ and $\mathbf{x}_s$ (for instance, $\mathbf{x}_2$ and $\mathbf{x}_3$ in Figure 3.7) return low conditional probability values of $\mathbf{x}_t$ and $\mathbf{x}_s$.

In image space, each axis would correspond to individual pixel values of images. Note that the representation achieves invariance to the pixel values of images as illustrated in Figure 3.7) in that the model considers $\mathbf{x}_t$ to be more similar to $\mathbf{x}_1$ than $\mathbf{x}_2$ and $\mathbf{x}_3$ even though the two are closer to $\mathbf{x}_t$ in terms of the Euclidean distances based on pixel values. The analogy extends to the high dimensional space. The reason I do not use $p(\mathbf{x}|\hat{y}_t)$ to find similar data points to a target image $\mathbf{x}_t$ is that data points near the origin (for instance, $\mathbf{x}_o$ in Figure 3.7) will be well captured by any multivariate Gaussian distributions regardless of their covariance information. The second term in the metric, $p(\hat{y}_c)$, favors the correlational structures that can be described by sparse latent variables; in the case that two candidates return the same value of the conditional probabilities of the target image given their latent variables, the metric prefers the one that results in sparser representation, as it returns higher prior probability values (Eq. 5.3). I demonstrate the usage of the joint probability described above using the image retrieval task; for a target image $\mathbf{x}_t$, I retrieve candidate scene images from a large scene image pool (I used 108,754 images in the SUN database as the pool), whose latent variable returns the highest joint probability value with $\mathbf{x}_t$, or equivalently the lowest value of $-p(\mathbf{x}_t, \hat{y})$. I call this the probabilistic correlational distance (PCD) hereafter. In Figure 3.8, I show the five most similar candidate scene images from 108,754 images retrieved with PCD, GIST, HOG, PHOG and spatial pyramid of SIFT. For GIST and HOG, I tried three different spatial scales (1×1, 2×2 and 4×4) and show the qualitatively best results. For all other representations than the distribution coding model, I used the Euclidean distances as similarity measures. Even though the model representation requires a small number of units to represent a scene image, the image retrieval results are qualitatively satisfactory. The distribution coding model achieves the efficiency by projecting scene images based on their characteristic features rather than representing scene images with fixed number of scales and orientations. In addition, it takes approximately 0.1 seconds to retrieve the similar images to targets using PCD which is fast enough for real-time image retrieval.

3.5 Quantitative evaluation of scene similarity measures

To investigate whether the global correlational information encoded by the distribution coding model is consistent with the perceptual similarities between scene images, I conducted an experiment in which subjects were asked to select candidate scene images that were most similar to a target scene image in terms of spatial layout. In each trial, a target image from one of 397 semantic categories of the SUN database [75] was presented together with 25 randomly chosen *candidate* scene images. Subjects were allowed to select more than one candidate images if they were equally similar to the target images. I call the selected candidate images *similar* images. In the trials when none of the candidate images were perceptually similar to the target images or when the target images mainly consisted of objects and it was thus difficult to get a sense of spatial layout of the scene, subjects could skip the trial. Subjects were specifically instructed

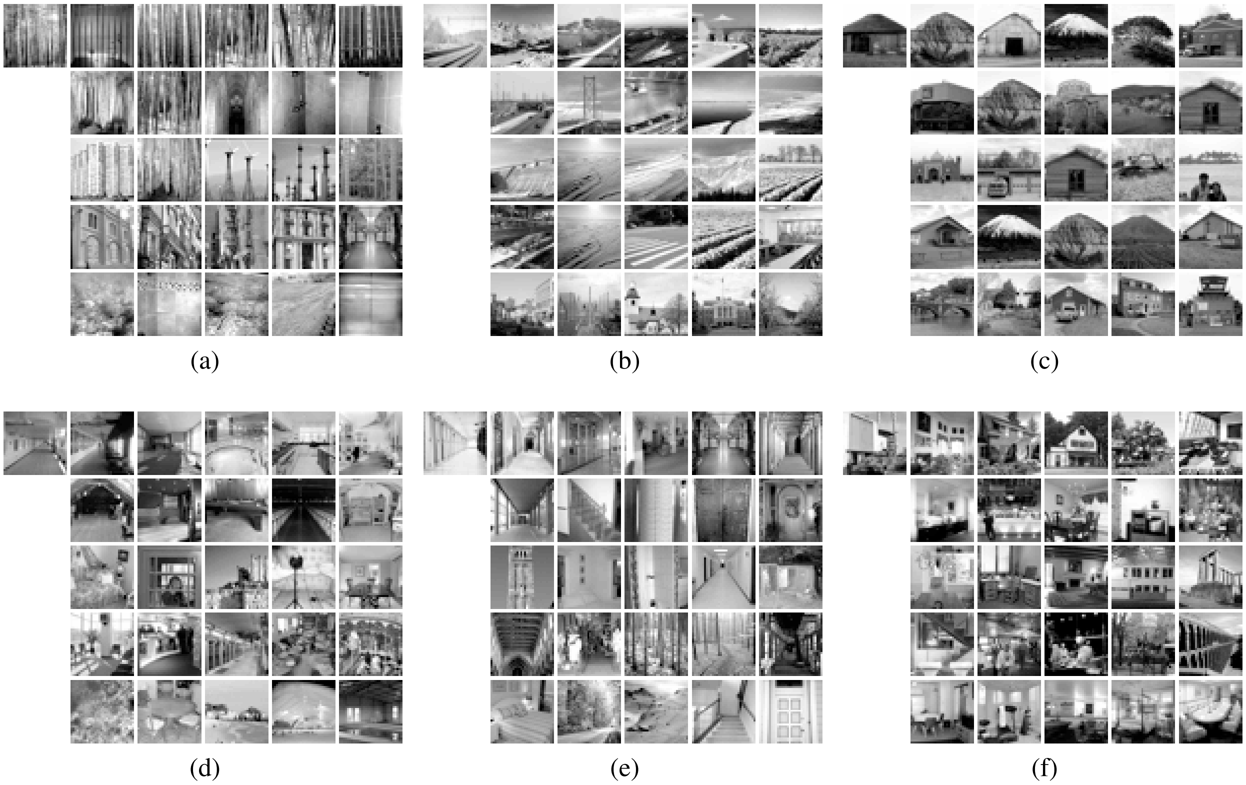


Figure 3.8: (a)–(h) Scene image retrieval results. The top left portion shows the target scene images. The retrieved images are ordered so that the left-most columns shows the most similar and the right-most columns show the 5 th similar candidate scene images to the targets. From the top to the bottom rows correspond to PCD, GIST(4×4), HOG(2×2), PHOG (3 levels), spatial pyramid of SIFT (3 levels). For (a)–(f) the target images are from the SUN database while the target images shown in (g)–(h) are not.

to focus on the shape and spatial layout of the scenes and to ignore non-spatial attributes such as color or types of objects in the scenes. Candidate images were chosen only from the same semantic categories as the target images, in order to control the difficulty of the tasks. Without such constraints, candidate images from different scene categories are too dissimilar to make meaningful judgements. In addition, using candidate images from the same category prevents subjects from depending on any semantic information to perform the task. Five subjects (one female; with normal or corrected to normal vision; 22-33 years old) participated in the experiment. I collected 2597 trials and the subjects selected 1.39 candidate images per trial on average (the number of candidate images selected per trial ranged from 0 to 13). Out of 2597 trials, subjects selected more than one similar images in 834 trials and selected zero similar images in 825 trials.

I evaluate the performances of various representations based on two criteria. The first one is the percentage of trials in which the similar images coincided with the closest candidate image to the target in a feature representation, the *closest* image. If a feature representation is consistent with the perceptual properties of images, the closest image will be perceived to be similar to the targets. The other criterion is the mean rank of the similar images when all the candidates in a trial are sorted in the ascending order in terms of the distances to the target in each representation. I assume that similar images will be more likely to have shorter distances to the targets than others and thus will have lower mean ranks of similar images if the distance is consistent with perception.

Feature	Resolution	Mean Ranks			
		Total	In	O-N	O-M
DCM	32×32	**7.01**	**5.91**	**7.16**	**6.93**
GIST(4×4)	128×128	10.7	9.67	10.4	11.0
ICA	32×32	11.7	10.7	11.1	12.2
HOG(2×2)	128×128	10.9	9.70	10.9	10.9
PHOG(L=3)	128×128	11.0	11.0	10.9	11.0
SIFT(L=3)	128×128	9.78	9.35	9.88	9.72

Table 3.2: Performance evaluation of various representations for the perceptual experiment on scene layout similarities. I show detailed performances for subcategories of scene images. In, O-N and O-M correspond to Indoor, Outdoor natural and outdoor manmade scenes, respectively. The percentage refers to the fraction of trials in which candidate images retrieved based on each representation is reported to be similar to the targets. Mean Ranks refer to the average rank of the selected candidate scene images based on the similarity between the candidates and the corresponding targets based on each representation

I use PCD introduced in the previous section as the scene similarity measure for the distribution coding model. For other representations, the Euclidean distances between the features extracted from images were adopted as the similarity measures. As I trained the distribution coding model and ICA on images of 32×32 resolutions, I downsampled the original images to 32×32 pixels and then extracted the corresponding features. For all other state-of-the-art representations, I extracted the features from images of 128×128 resolutions. As reported in Table 3.2, PCD shows the most consistencies with the perceptual experiment in terms of both criteria. Note that the percentage criterion only takes into account the closest images whereas the mean ranks criteria considers all the similar images within a trial.

3.6 Conclusion

We trained the distribution coding model to learn the correlational information on the whole scene images. The model parameters show global correlational structures reflecting the regularities found in the scene images. Adaptive representation to the characteristic statistics allows encoding of the data with a small number of latent variables. In addition, the experiment for perceptual scene image similarities suggest that the model representation is a good scene image descriptor with significantly greater consistency with perceptual properties of the global structures in scene images. The probabilistic correlational distance can be used for image retrieval systems. Also the latent variable encoding the covariance information is significantly more predictive of perceptual spatial layouts (depth and openness) of scene images.

Our approach can be extended to larger size images for encoding more detailed local information by first learning the correlational structures on local patches and integrating the local information over space. Also, the probabilistic distance measure introduced in this paper can be utilized not only for whole image retrieval but also for finding local interest matching points between images. As the model represents images or patches based on their adaptive representation rather than fixed number of scales and orientations, it could find match points more accurately especially in natural scenes in which points and lines are not defined by as high contrasts as indoor or manmade scenes. Extending the model training to images describing mainly of objects can also be useful for understanding object invariances under diverse viewing angles or nonrigid objects. Lastly, the analysis can be applied to face recognition system.

Chapter 4

Adaptive scene representation based on multiscale correlational patterns

4.1 Introduction

Automatic scene recognition is essential for many aspects of vision, including visual memory, contextual cues for visual recognition, spatial memory for location, or navigation. Recently, the vision community has invested tremendous effort towards solving it, by providing plentiful and accurately labeled image datasets for intelligent systems to be trained on [10, 75] . However, the growth in the size of scene databases challenges the development of visual features that are suited for perceptual scene discrimination and at the same time can tolerate irrelevant variations. For such large-scale datasets, visual features and algorithms that worked well for smaller datasets are not as successful [11, 41]. This is because usually the large-scale databases contain much finer scene categories, which may be more difficult to discriminate from each other especially when differences between categories are subtle compared to irrelevant variations among scene images belonging to the same category. Moreover, with finer scene categories often comes the issue of having fewer positive examples for training. These are all serious challenges to the development of optimal features for reliable scene classification.

Accurate registration of global structure is a key property of a scene descriptor that is sensitive enough to discriminate densely sampled scene images but is robust to irrelevant variations. Global structures of scene images give rise to perceptual spatial layout properties of scene images such as depth [60], openness and perspective [20]. In addition, scene images that belong to the same category tend to have similar global structures [44] suggesting that the global information contributes to semantic properties of scenes. Also, contextual information driven from the global structures facilitates object detection [46].

Global structures of scene images have been typically extracted from large images in a fine-to-coarse fashion. Original images are first divided into small, regularly spaced patches. From these, fine scale information is extracted and used for constructing the coarse scale information. To that end, the most popular approach so far [6, 9, 45, 74] has been to concatenate local features extracted from predefined locations. Therefore, global structure encoded in these representations is heavily dependent on fine scale information; the underlying assumption here is that coarse

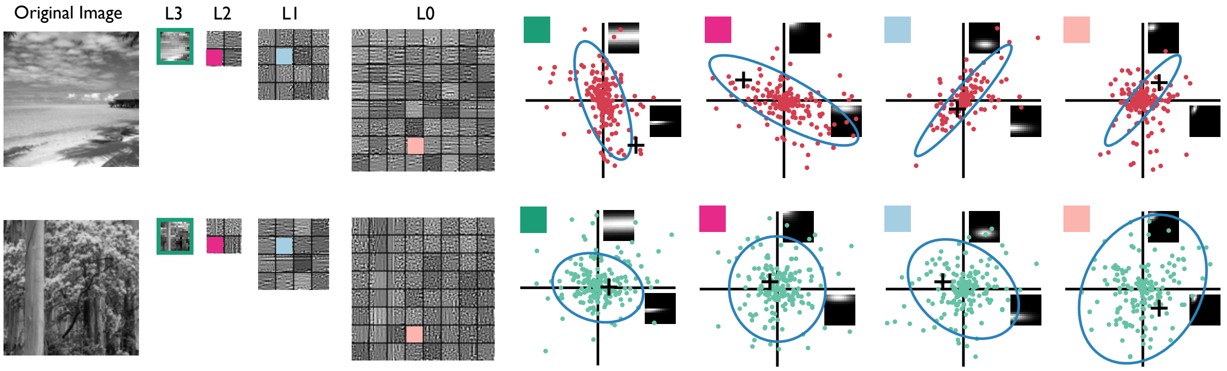


Figure 4.1: LP and the patterns of their responses to linear filters. Left: I show two exemplar scene images with distinct global properties; the upper row shows an open scene with relatively far mean depth and not much gradient in depth whereas the one in the bottom describes a closed scene with relatively uniform depth. The resolutions of L_0, L_1, L_2 and L_3 correspond to $128 \times 128, 64 \times 64, 32 \times 32$ and 16×16, respectively. L_3 is displayed in twice the size of its original resolution for visualization purpose. Right: The plots show the filter responses of LP extracted from 200 scene images with high openness ratings (top row) [56] and another 200 scene images with low openness (bottom row). The corresponding linear filters are shown on each axis. The black cross symbols in each panel corresponds to the sample image shown on the left. The color code on the top left of each panel indicates the corresponding locations in the pyramid. The filter responses are centered around zero and will be hard to be distinguished when the responses are overlapped. However, two different populations of scene images with distinct global structures reveal distinct correlational patterns.

scale representation is well approximated by local structure. A potential disadvantage of encoding global structure by concatenating local structures is that the representation is highly sensitive to the choice of parameters in terms of scales and orientations [56].

Another technique for encoding global structure based on local structure is computing the mean or preserving the maximum responses of local responses over restricted regions [7, 32, 61]. These techniques have been actively developed for mid-level representations and the features derived from them have been shown to be responsive to object parts [77]. However, when there are competing structures, only the most dominating one among all local features over the restricted region will be encoded and passed on to the coarse level representation. This limits the effectiveness of such approaches for encoding the global structures, since sensitive representation of global structures often requires registering attenuated long-range patterns.

Motivated by the need for more sophisticated encoding of global structures in scene images, I propose a multiscale scene descriptor. To achieve this, I first decompose original images into separate scales and extract features separately for each scale [4]. This approach allows to encode the generic coarse scale information rather than building it based on fine scale information. I use the Laplacian pyramids (LP) [8] as a means to disentangle original images into separate layers of localized spatial frequencies. Since LP representation is overcomplete [64], extracting lower dimensional features from the over-complete representation is necessary to take advantage of the separation among different scales. As it is not well understood what type of features are useful for LP, I propose to learn adaptive representations [49, 53, 71] of LP rather than to use image descriptors hand-tailored to the properties of original images.

Due to the high computational costs and the limited number of samples for training, adaptive representations are typically trained on relatively small size patches rather than on entire images [28]. Similarly to the previously mentioned approaches, this method is limited by the assumption that the long-range global properties of images are contained in the regularly spaced patches; as a result, such representation is sensitive to the alignment of the local patches. Adaptive image representations on top of LP can handle this issue naturally. Highest level sub-bands which contain the most coarse scale information are low-dimensional (in Fig. 4.1, L_3 is of 16×16 pixels) and therefore I can train adaptive representations on them without further dividing them into smaller size patches. For lower-level sub-bands, I still need to divide them into grids to train the adaptive representations. However, as the lower-level sub-bands contain high spatial frequency information, the potential information loss by dividing into grids is less drastic.

In this chapter, I automatically learn an appropriate feature representation for LP by directly modeling their probability density. This is based on the observation that scene images of distinct perceptual properties demonstrate different correlational patterns of linear filter responses for separate layers of the pyramids (Figure 4.1). The difference in correlational patterns of responses to pairs of linear filters accumulates as I consider more filters and regions in the pyramids to distinguish scene images with distinct properties. These patterns motivated training the density coding model (DCM) [29] discussed in Chapter 3 on LP since DCM essentially learns the dictionary of filters that best encodes the population of data in terms of their correlational structures. Directly modeling the statistical distributions of data leads to a representation that achieves invariance to nonessential variations by characterizing each sample with respect to the major directions along which the population of samples is aligned.

4.2 Related work

4.2.1 Laplacian pyramid

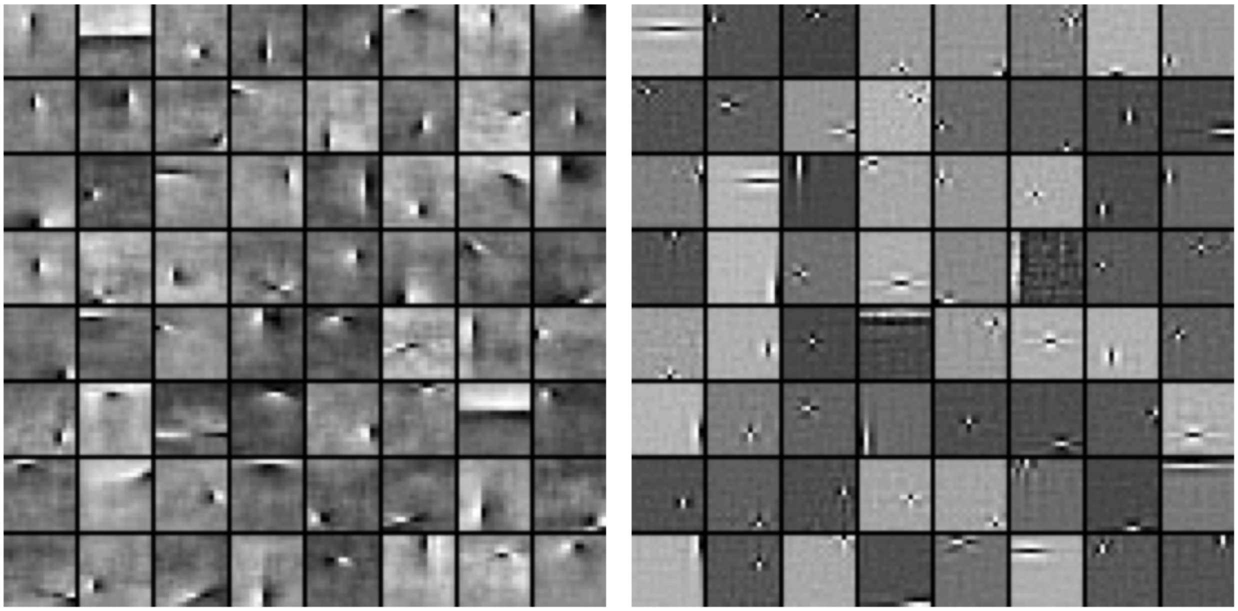
To decompose scene images into distinct scale sub-bands, I employ the Laplacian pyramid (LP) [8]. It first approximates a signal with its low-pass filtered and down sampled version. The higher resolution version is predicted by upsampling and filtering the low-pass filtered signal. The difference between the predicted version and the original version is stored and the process repeats. Figure 4.1 illustrates LP of examplar images; here, I demonstrate 4-layers of LP extracted from 128×128 grayscale images. The coarse band (L_3 in Figure 4.1) encodes the global information in the original image while the finer bands (L_0–L_2) contain increasingly localized information. The benefit of this pyramidal representation is that features extracted from each level bring new information; the information in the higher level bands is not attainable from the lower level bands.

4.3 Training data and preprocessing

I trained DCM on a large scale scene database, the SUN database [75]. I first converted the images into grayscale and then resized them so that the shorter dimension (either height or width) of each image corresponds to 128 pixels. To increase the number of training examples, I randomly sampled up to 10 subsamples of 128×128 pixels along the longer axis for each image in the dataset. I extracted 4-layer LP (L_0, L_1, L_2 and L_3 are 128×128, 64×64, 32×32 and 16×16 pixels each) using "9-7" filters [12] and then divided each sub-band into equally spaced patches so that each patch is of size 16×16 (L_0, L_1, L_2 and L_3 results in 1×1, 2×2, 4×4 and 16×16 grids. I separately PCA-whitened 16×16 patches from each layer to maintain 99% of variance and avoid wasting resources modeling noisier parts.

4.4 Model parameters

As discussed in Chapter 3, $\mathbf{b}_k$ encodes a common direction along which the covariance units $\mathbf{A}_j$'s can vary. When trained on different layers of LP of scene images, $\mathbf{b}_k$'s reveal localized structures both in space and spatial frequencies (Fig. 4.2). Note that the model formulation did not constrain $\mathbf{b}_k$'s to have localized structures; rather, these automatically emerged while fitting the parameters to the scene image statistics.



(a) $\mathbf{b}_k$ trained on $L3$

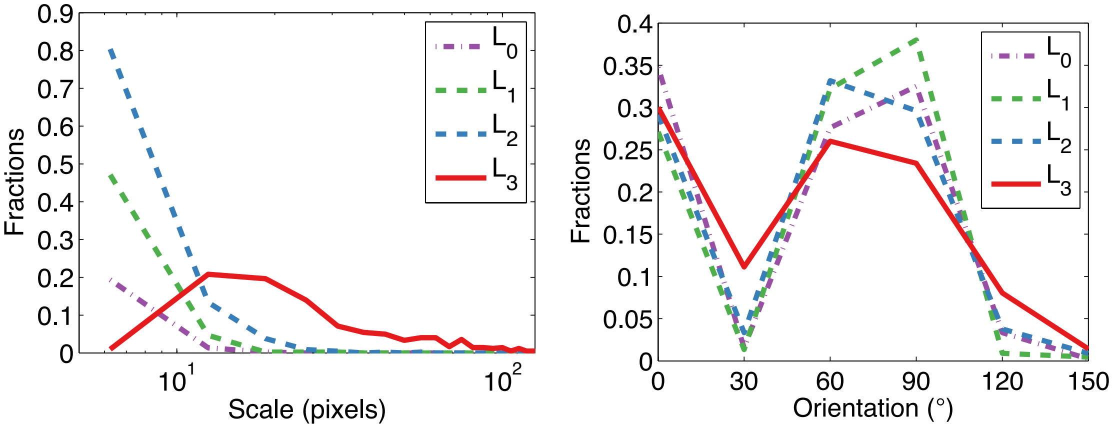
(b) $\mathbf{b}_k$ trained on $L0$

Figure 4.2: Random subsets of $\mathbf{b}_k$ optimized for each layer. I only show results trained on L_3 and L_0; $\mathbf{b}_k$s trained on L_1 and L_2 have qualitatively similar structures to L_0.

To estimate the distribution of scales and orientations of $\mathbf{b}_k$s, I fit them with Gabor filters[1].

[1] I used the routine developed in [13]. `http://www.snl.salk.edu/~edoi/resource.html`

The scale of $\mathbf{b}_k$s are defined as $\sqrt{\sigma_x^2 + \sigma_y^2}$. The $\mathbf{b}_k$s trained on L_0–L_2 show similar distribution of scales and orientations (Fig. 4.3); the $\mathbf{b}_k$s with small receptive fields and the horizontal and vertical orientations dominate. On the other hand, $\mathbf{b}_k$s fitted to the low-pass filtered images, L_3, have higher fractions of units with larger receptive fields. Also, L_3 show more evenly distributed orientations. The non-isotropic distribution of scale and orientation of $\mathbf{b}_k$s suggests that DCM allocates more resources to prevalent visual structures in scene images. This contrasts with most hand-designed visual features which tend to allocate uniform bits of information to all orientations and scales.

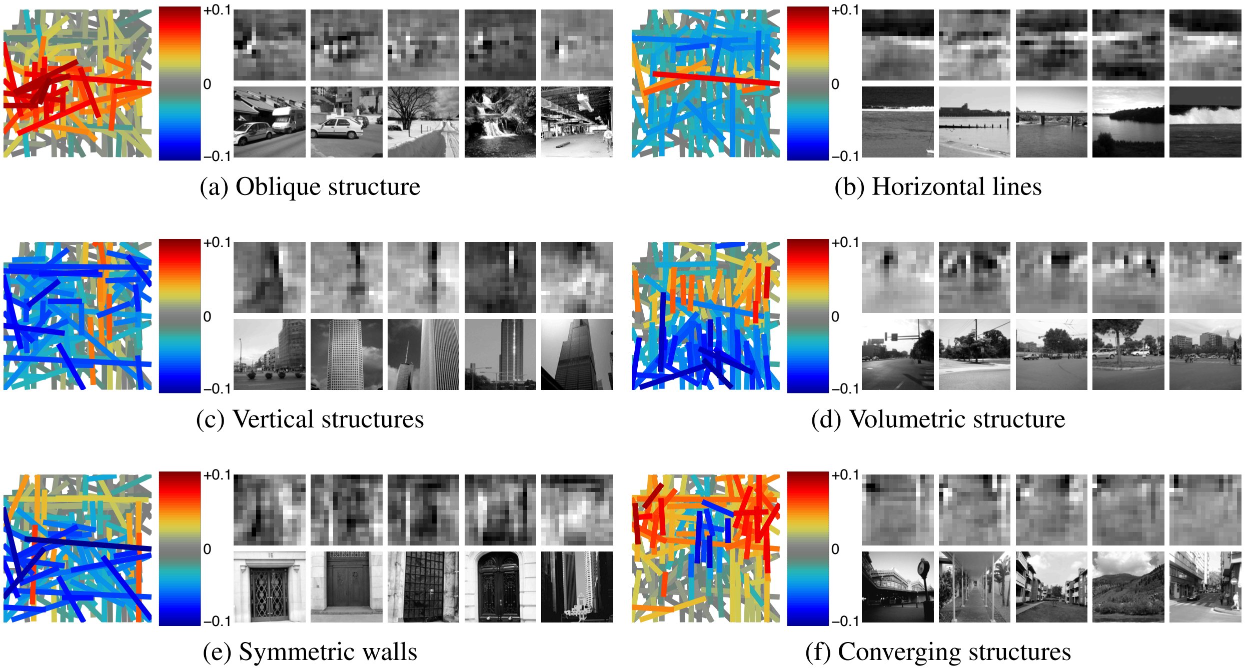


(a) Scale

(b) Orientation

Figure 4.3: Distributions of scales and orientations of $\mathbf{b}_k$s trained on separate layers. L_3 reveals the basis functions with larger receptive fields and more even distributions in terms of orientations compared to other layers. $\mathbf{b}_k$s trained on L_0–L_3 show similar distributions to each other.

To visualize $\mathbf{w}_j$s, I first assign a bar to each $\mathbf{b}_k$ which has the same location and orientation with that $\mathbf{b}_k$ in the image space based on the Gabor filters fitted to it. I then assign a color value corresponding to $w_{j,k}$ to each bar. In Figure 4.4, I show a subset of $\mathbf{w}_j$ trained on L_3. I demonstrate the global correlational structures encoded by $\mathbf{w}_j$ by generating random samples from a multivariate Gaussian distribution centered around zero whose covariance matrices is $\exp(y_j \mathbf{A}_j)$ $(y_j > 0)$. The generated samples show qualitatively similar visual structures with the corresponding covariance matrices. In addition, scene images which have the highest values of $\hat{y}_j$ among the samples in [56] contain visual structures that resemble the correlational structures encoded in $\mathbf{A}_j$. In contrast to L_3, $\mathbf{w}_j$ trained on L_0–L_2 reveal high frequency structures (Fig. 4.5).



(a) Oblique structure

(b) Horizontal lines

(c) Vertical structures

(d) Volumetric structure

(e) Symmetric walls

(f) Converging structures

Figure 4.4: Representative $\mathbf{w}_j$ trained on L_3 with corresponding color bars. Red corresponds to positive values of $w_{j,k}$ while blue represents the negative values. On the right, top rows show images generated from multivariate Gaussian distributions with $\exp(y_j \mathbf{A}_j)$ as covariance matrices ($y_j > 0$, Eq. 5.2). The bottom rows show scene images from [56] which have the highest values of $\hat{y}_j$. Note that the images in the top rows are of size 16×16 and the bottom rows 128×128. Best viewed in color and in magnification.

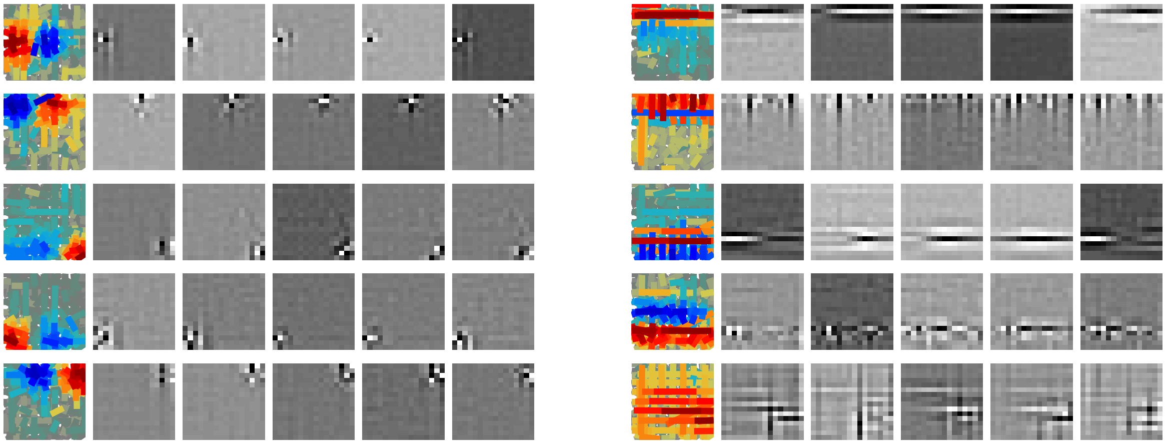


Figure 4.5: Representative $\mathbf{w}_j$ trained on L_2 and the random samples generated using the corresponding units. Some units have localized high spatial frequency structures (left) whereas others show high spatial frequency structures over larger receptive field (right). L_0–L_2 show qualitatively similar structures. Best viewed in color and in magnification.

4.5 Experimental Results

For new scene images, I first convert them into 128×128 gray scale images, extract LP and infer the latent variables using the model parameters as discussed in Section 4.4. I employ the latent variables for each image to predict semantic categories (Section 4.5.1) and perceptual spatial layout ratings (Section 4.5.2).

4.5.1 Scene categorization

In this section, I examine the latent variables of DCM as the visual features for encoding the scene categories. I trained support vector machine classifier with radial basis function (RBF) kernels. The σ values of RBF kernels were selected by cross validation. For individual categories in a dataset, I train a 1-vs.-all classifier on a train set; for each scene category, I set the examples from the scene category as positive and the rest as negative. As the baseline, I train SVM classifiers using spatial pyramid (SP) [32], HOG [9], PHOG [6] and GIST [44].

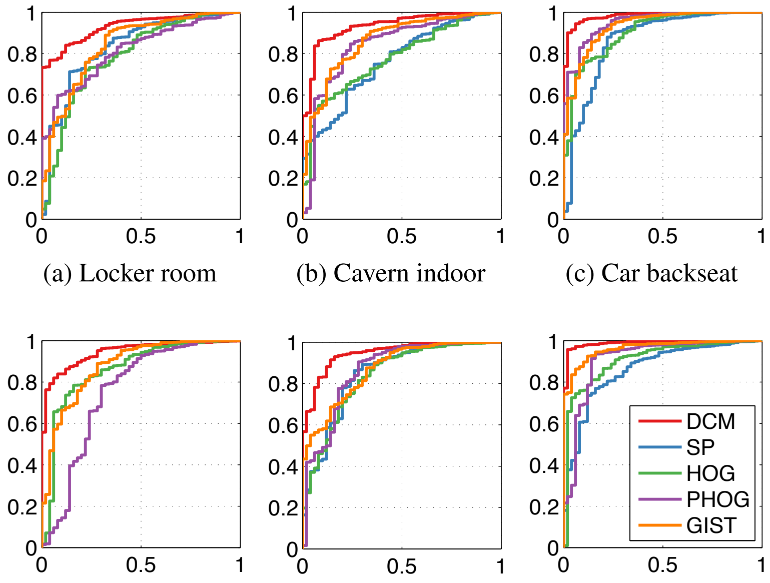
Table 4.1 shows the AUCs of 1-vs.-all classifiers for scene categorization. For each dataset, I report the average over all categories in the dataset ("Total") and also the averages over categories that belong to indoor (In), outdoor natural (O-N) and outdoor manmade (O-M) scenes separately. For *LabelMe* dataset (60–410 color images of 256×256 pixels from 8 categories) [44], the performance of DCM is on par with baseline features (PHOG). For a larger dataset, *15scene* dataset [17, 32], DCM shows significantly better performance over other baseline representations overall. DCM is especially useful for 1-vs.-all classifiers for indoor scene categories ("bedroom", "kitchen", "living room", "office" and "store").

For *SUN* database [75], I performed the scene categorization task at two levels of the hierarchy; the highest level in which scene images are categorized into indoor, outdoor natural and outdoor man-made scenes and the leaf level with 397 scene categories. DCM significantly outperforms other baseline representations although by narrow margin. This is because the performance of each feature varies depending on the property of the target categories. Figure 4.6 shows a few examples of leaf-level categories in *SUN* database for which the multiscale DCM representation yields superior performance to others.

The performance gap between different methods is widened for finer-scale scene categorization. For instance, for *LabelMe* and *15scene*, the gap between the best and the worst performance on each category is 0.04 (ranging from 0.01 to 0.09) and 0.06 (ranging from 0.01 to 0.13) on average whereas the leaf level scene categorization of *SUN* database has the average gap is 0.11 (ranging from 0.01 to 0.28). Also, the average range of each feature for *LabelMe* is 0.10, whereas for the leaf-level categorization for *SUN* it is 0.40. This implies that for more challenging scene categorization, which discriminates scene categories at finer-scales and inherently has smaller number of positive examples, it is difficult to develop an optimal feature that works well for all categories. The multiscale DCM provides a principled way to mitigate this challenge by encoding scene structures over multiple scales and learning the optimal representation based on data statistics.

Condition		DCM	SP	HOG	PHOG	GIST
LabelMe	O-N	0.95	0.93	0.93	0.95	0.93
	O-M	0.97	0.95	0.93	0.97	0.96
	Total	0.96	0.94	0.93	0.96	0.94
15Scene	In	**0.93**	0.87	0.86	0.88	0.91
	O-N	0.96	0.94	0.93	0.95	0.95
	O-M	**0.93**	0.90	0.88	0.92	0.91
	Total	**0.94**	0.90	0.89	0.92	0.92
SUN	In	**0.91**	0.84	0.84	0.86	0.87
	O-N	**0.95**	0.88	0.87	0.93	0.93
	O-M	**0.83**	0.76	0.77	0.80	0.80
	Total	**0.89**	0.83	0.83	0.87	0.87
SUN Leaf	In	**0.85**	0.78	0.79	0.81	0.84
	O-N	**0.90**	0.83	0.84	0.87	0.89
	O-M	**0.85**	0.78	0.80	0.83	0.84
	Total	**0.86**	0.79	0.80	0.83	0.85

Table 4.1: Scene classification performance (AUC) of 1-vs-all nonlinear SVM classifiers applied to Indoor (In), Outdoor Natural (O-N), and Outdoor Man-made (O-M) scenes. Bold-faced figures indicate statistical significance (paired t-test; $p < 0.05$).



(a) Locker room (b) Cavern indoor (c) Car backseat

(d) Wind farm (e) Elevator (f) Sauna

Figure 4.6: ROC curves of 1-vs.-all classifiers trained on SUN database.

4.5.2 Perceptual spatial layout

In this section, I evaluate how well our proposed scene descriptor predicts the perceptual spatial layouts of scene images using the human ratings collected in [56]. I use multilinear regression with stepwise method [14] for predicting the perceptual ratings of depth, openness and perspective collected on a 1-to-6 continuous scale. Mean depth [44] refers to depth in a global sense related to the physical size of a scene (1= close to the camera, 6= far). Openness [20] of a scene refers to the quantity and location of boundaries in a scene (1= large portion of unobstructed sky and dominant horizontal lines; 6= closed scenes with limited spatial extent). Perspective [48] of a scene describes the amount of depth expansion within the scene (1= strong convergence between the parallel lines; 6= scenes with surfaces at fairly uniform distances from the camera).

Table 4.2 shows the mean error of predicting the perceptual ratings. The dataset consists of urban scenes and natural scenes. I train and test on each category separately and also using both categories ("Total" condition in Table 4.2). DCM outperforms other baseline features for predicting the perceptual ratings using both urban and natural scenes. DCM is especially accurate at predicting openness ratings. For perspective ratings, GIST and PHOG results in accurate predictions using only natural and urban scenes respectively, but for more general task of predicting perspective ratings irrespective of contents, DCM shows better performance. Overall, DCM outperforms in predicting perceptual ratings in a scenario where the train and the test sets consist of scene images with more diversity in terms of contents ("Total" condition).

Condition		DCM	SP	HOG	PHOG	GIST
Openness	Natural	**0.80**	1.08	1.08	0.99	1.02
	Urban	**0.70**	1.04	1.03	0.90	0.94
	Total	**0.77**	1.06	1.09	0.99	1.00
Depth	Natural	0.69	0.74	0.75	0.72	0.69
	Urban	0.60	0.70	0.66	0.63	0.62
	Total	**0.66**	0.72	0.72	0.69	0.68
Perspective	Natural	1.21	1.28	1.20	1.20	**1.17**
	Urban	1.18	1.36	1.24	**1.15**	1.20
	Total	**1.19**	1.33	1.30	1.22	1.23

Table 4.2: Error of stepwise regression functions for predicting the perceptual spatial layout ratings. Bold faced fonts indicate statistical significance (paired t-tests; $p < 0.01$).

4.6 Conclusion

I proposed an adaptive representation which encodes the global structure of scene images by separating the input into different scales and extracting correlational structures separately from each scales with DCM. To facilitate and stabilize the training procedure, I optimized the parameters by imposing constraints on their magnitudes ($L2$-norm), leading to a manifold algorithm. The model affords intuitive visualization of statistical properties of scene images and this can be applicable for developing future scene descriptors. I demonstrate that our approach is useful for characterizing scene images in large scale databases which requires more sensitive encoding of global properties informative of high-level properties such as semantic and perceptual properties. Future research will develop more compact representation of the global properties of scene images by exploiting the conditional distribution of fine scale structures given the corresponding coarse scale structures.

Chapter 5

Learning image manifold structures with canonical images and correlational structure

5.1 Introduction

The space of scene images are inherently high dimensional and therefore it is difficult to study their manifold structures. Previous studies have applied manifold learning techniques to images and demonstrated the effect of lighting conditions and view points by projecting them onto lower dimensional (2D) spaces [57, 59, 67, 73]. However, the datasets adopted in those studies consist of very carefully curated images such as an object or a face viewed from many different angles and under different lighting conditions or hand-written figures of a number collected from many people. While these studies certainly shed light on the high-dimensional shapes of image manifolds, the results cannot be easily generalized since it is not practical to collect such well curated datasets for several objets or scene types. To develop a more general framework for studying high-dimensional image manifold structures, one should develop a more general model which depends on less conservative assumptions. In this chapter, I introduce a way to to do so by assuming the high-dimensional manifolds as mixtures centered around a set of canonical points and learn the covariances between samples and the canonical points to which they are close by. To learn the detailed structures of the local manifolds, I learn the covariance matrices specialized for each data point around its closest canonical point.

Subdividing a dataset into smaller groups and finding representatives of such small groups has been actively studied in many domains such as images, biological and geological data[15, 18, 58, 58]. However, the previous methods often fail to extend to larger dataset with high-dimensional data points such as images resulting in few clusters that are too general to encode the detailed structures of the local manifolds. I develop an algorithm which discovers clusters of images specialized enough to keep the local structures of the manifolds. I adopt the euclidean distances between the data points as a criterion for assigning a sample to a cluster. When the dataset is not sampled densely enough, however, the distance may fail to capture the true similarities or dissimilarities between samples. To improve the accuracy of the euclidean distances, I introduce a procedure called *image normalization* in which I iteratively apply physically plausible transformations to a sample until the distance between the transformed samples and a canonical image converges. The types of physically plausible transformations can vary according to the nature of datasets; in this work, I considered a sample image as a 2-dimensional plane and projected the plane onto other planes by varying the scale, translation and rotations (Figure 5.1). These transformations approximately mimic the effects of different camera angles, positions and the focal lengths. I then measure the minimum distances between the set of transformed samples based on a real sample and a canonical point. This way, I am allowing images describing a same scene but taken with small range of camera parameter perturbations to be taken into account when judging a sample's similarity to others. This procedure allows the system to be invariant to the camera transformations, which is more general form of invariance to local translations covered in [3, 34]. Also, identical forms of transformations were applied to all the data points without any need to approximating the transformations for each sample [63]. Inferring the camera parameters or spatial layouts from a single picture has been well studied [24]. Some algorithms depend on finding reliable vanishing points [66] and others are specialized for indoor scenes with walls, floors and ceilings [21]. My approach was developed because the previous studies do not work well for general cases when a certain reference structure cannot be reliably detected and the algorithms are often computationally expensive to be applied to a large-scale dataset.

Once the clusters and the canonical points are discovered and the samples are concentrated around them, I train the density component model discussed in Chapter 3 centered around the canonical points using the tightened data points. Learning the latent models for multivariate Gaussian distributions has been mostly approached with the assumption that the mean component is a linear sum of a set of basis functions [54, 65]. The basis functions trained for the means are inherently off from the actual data manifolds by definition, and it is difficult to read any structures from them. My approach allows to anchor the datasets along the canonical images close to the actual data points and reveals interesting patterns inherent in data.

5.2 Image normalization

In this section, I discuss the model I used for applying the physically plausible transformations to input image.

5.2.1 Background

When a plane is projected onto another plane, a point $\mathbf{a} = (a, b, c)$ is projected onto $\mathbf{a}' = (a', b', c')$. The relationship between the two is represented by a non-singular 3×3 matrix:

$$\begin{pmatrix} a' \\ b' \\ c' \end{pmatrix} = \mathbf{H} \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix}$$

In other words, the projection of $\mathbf{a} = (a, b, c)$ onto another point would result in 2D $\boldsymbol{\alpha} = (\alpha, \beta)$

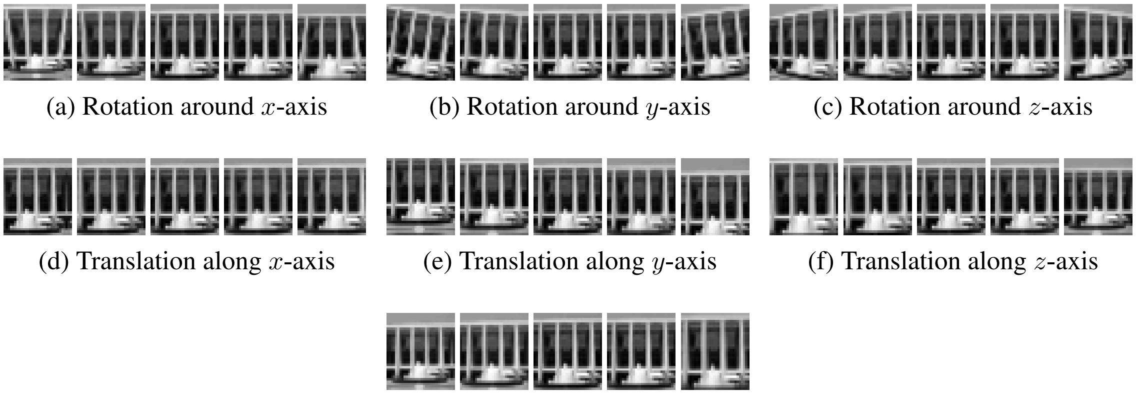
$$\alpha = \frac{a'}{c'} = \frac{h_{11}a + h_{12}b + h_{13}c}{h_{31}a + h_{32}b + h_{33}c} \qquad \beta = \frac{b'}{c'} = \frac{h_{21}a + h_{22}b + h_{23}c}{h_{31}a + h_{32}b + h_{33}c}$$

Projection of a 3D object to a plane through a camera consist of rotation around x-axis θ, y-axis ϕ and z-axis ψ, translation along x-, y-, z- axis, t_x, t_y, and t_z, camera focal length f_x, f_y and camera centers c_x and c_y.

$$\begin{aligned}\mathbf{H} &= \mathbf{MR} \\ &= \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\theta)\cos(\phi) & \sin(\theta)\cos(\psi) + \cos(\theta)\sin(\phi)\sin(\psi) & t_x \\ -\sin(\theta)\cos(\phi) & \cos(\theta)\cos(\psi) - \sin(\theta)\sin(\phi)\sin(\psi) & t_y \\ \sin(\phi) & -\cos(\phi)\sin(\psi) & t_z \end{bmatrix}\end{aligned}$$

As it is expensive to extract the camera parameters of a picture [23], I consider each image as a 2D plane and assume that the pictures has been a projection of the plane with the default camera parameters where the focal length was set to 1 and all translations and rotations parameters to zeros. To simplify our model, I assume the cameras are centered at zero ($c_x = c_y = 0$) and the vertical and the horizontal focal lengths are equal to each other ($f_x = f_y$) resulting in the simplified version of $\mathbf{H}$:

$$\begin{aligned}\mathbf{H} &= \mathbf{MR} \\ &= \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\theta)\cos(\phi) & \sin(\theta)\cos(\psi) + \cos(\theta)\sin(\phi)\sin(\psi) & t_x \\ -\sin(\theta)\cos(\phi) & \cos(\theta)\cos(\psi) - \sin(\theta)\sin(\phi)\sin(\psi) & t_y \\ \sin(\phi) & -\cos(\phi)\sin(\psi) & t_z \end{bmatrix}\end{aligned}$$



(a) Rotation around x-axis

(b) Rotation around y-axis

(c) Rotation around z-axis

(d) Translation along x-axis

(e) Translation along y-axis

(f) Translation along z-axis

(g) Focal length

Figure 5.1: Types of transformations applied. In each panel, the center (3rd column) shows the original images.

5.2.2 Image projection as linear operation

When camera transformations are applied to an image, the projected images become of different shape or resolution than the original image. For learning the probabilistic models on the whole scene images, it is required that all the normalized images are of the same size. To tackle this issue, I modeled the transformations as a matrix multiplication form. I assumed that the original image is extended at its edges and corners by its mirrored form to avoid the boundary issues such as some pixels of the projected images being blank.

Given a projection matrix $\mathbf{H}$, I model the intensity of the projected image $\mathbf{y}'$ as a weighted sum of intensities of the original image $\mathbf{y}$.

$$\mathbf{y}'(\boldsymbol{\alpha}) \sim \mathcal{N}\left(\sum_{\mathbf{a}} s_{\boldsymbol{\alpha},\mathbf{a}}\mathbf{y}(\mathbf{a}), \Sigma_\alpha\right)$$

or in the matrix form,

$$\mathbf{y}' \sim \mathcal{N}\left(\mathbf{S}\mathbf{y}, \Sigma\right)$$

where $\mathbf{y}$ and $\mathbf{y}'$ are an origin image and its projection in vector forms, $\mathbf{y}(\mathbf{a})$ corresponds to an entry of $\mathbf{y}$. The weight assigned to $\mathbf{y}(\mathbf{a})$ for $\mathbf{y}'(\boldsymbol{\alpha})$, $s_{\boldsymbol{\alpha},\mathbf{a}}$, which corresponds to is $(\boldsymbol{\alpha}, \mathbf{a})$ th entry of $\mathbf{S}$, is formulated as

$$\begin{aligned}
&\log(s_{\boldsymbol{\alpha},\mathbf{a}}) \propto -\frac{1}{2\sigma^2}\left(\alpha - \frac{h_{11}a + h_{12}b + h_{13}}{h_{31}a + h_{32}b + h_{33}}\right)^2 - \frac{1}{2\sigma^2}\left(\beta - \frac{h_{21}a + h_{22}b + h_{23}}{h_{31}a + h_{32}b + h_{33}}\right)^2 \\
&\propto -\frac{1}{2\sigma^2}\left(\alpha(h_{31}a + h_{32}b + h_{33}) - (h_{11}a + h_{12}b + h_{13})\right)^2 \\
&-\frac{1}{2\sigma^2}\left(\beta(h_{31}a + h_{32}b + h_{33}) - (h_{21}a + h_{22}b + h_{23})\right)^2
\end{aligned}$$

$h_{..}$ are functions of the parameters $f, \theta, \psi, \phi, t_x, t_y$ and t_z. Figure 5.12 shows how manipulating each parameter changes the way original images are projected. The above formulation corresponds to applying Gaussian filters to the original image $\mathbf{y}$ to get its projected image $\mathbf{y}'$. σ determines the width of the Gaussian filters; if it is set to too small value each pixel in the projected image would be drawn from one pixel of the original image and may cause the projected image to be blocky. On the other hand, high values of σ would return blurry projected images. I fixed the σ as three pixels.

5.2.3 Image normalization algorithm

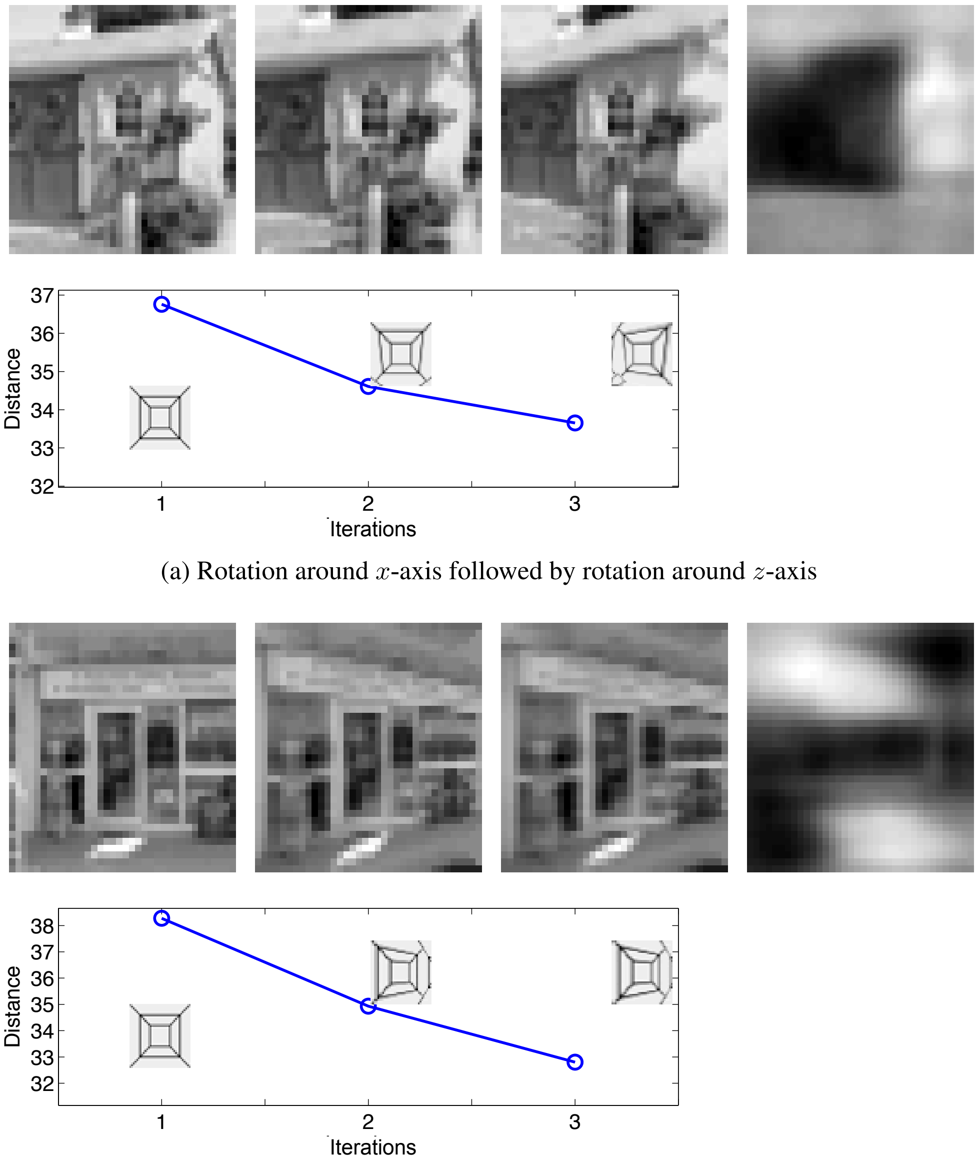
To find a set of representative points or canonical points and let the data points to be normalized to them, I start with a random sample from the dataset as a canonical point. Then, I normalize samples in the dataset so that the sample gets closer to one of the canonical points. If a normalized sample is close enough to one of the canonical points, I update the canonical point as the weighted linear sum of the canonical image and the normalized sample. If not, the sample is added as a new canonical point. Here, I used the euclidean distances between points as the distance measure $d(\mathbf{x}, \mathbf{a})$ and the distance between a sample and a set of points was defined as the minimum distance between the sample and the elements of the canonical set.

The set of physically plausible transformations may vary for different datasets and applications. In this work, the set of transformations consist of 70 types along seven types of camera parameters ($f, \theta, \psi, \phi, t_x, t_y$ and t_z) illustrated in Figure 5.1 and ten values per each parameter.

Figure 5.2 demonstrates lines 7–9 of the Algorithm applied to scene images and their corresponding canonical images. While the euclidean distance is not the most accurate measure of the dissimilarities between high-dimensional samples, searching for transformations which projects the input onto another image with similar structures allows the overall pattern becomes better matched to their corresponding canonical images. I run the algorithm several times and use the set of canonical images and the corresponding normalized dataset which have the minimum mean distance between the normalized images and the set of canonical images.

Algorithm 1 Data normalization

```
 1: Input : $\mathbf{X} = \{\mathbf{x}_1, \cdots, \mathbf{x}_N\}$ and $F$ = a set of physically plausible transformations
 2: $\tau$ = a predefined threshold for distances by user
 3: Output : a set of canonical points $\mathbf{A}$, normalized dataset $\mathbf{Y} = \{\mathbf{y}_1, \cdots, \mathbf{y}_N\}$
 4: initialize $\mathbf{A} = \{\mathbf{a}_1 = \mathbf{x}_1\}$ and $n_1 = 1$
 5: for $i := 2$ to $N$ do
 6:     $\mathbf{y}_i = \mathbf{x}_i$
 7:     while not converged do
 8:         $f_m = \arg\min_{f \in F} d(f(\mathbf{y}_i), \mathbf{A})$
 9:         $\mathbf{y}_i = f_m(\mathbf{y}_i)$
10:     end while
11:     if $(d(\mathbf{y}_i, \mathbf{A}) < \tau)$ then
12:         $l = \arg\min_l d(\mathbf{y}_i, \mathbf{a}_l)$
13:         $\mathbf{a}_l \leftarrow \frac{n_l \mathbf{a}_l + \mathbf{y}_i}{n_l + 1}$
14:         $n_l \leftarrow n_l + 1$
15:     else
16:         $\mathbf{A} \leftarrow \mathbf{A} \cup \mathbf{x}_i$
17:     end if
18: end for
```



(a) Rotation around x-axis followed by rotation around z-axis

(b) Rotation around z-axis followed by translation along x-axis

Figure 5.2: Illustration of the image normalization. The first column shows the original images. The scenes in the second column is the transformed version of the original images in the first column. The types of transformations and the degrees of such transformations were selected such that the transformed scene image is closest to the corresponding canonical images shown in the last columns. The images in the third column is the transformed version of the images in the second column. The transformation which minimizes the distance between the transformed image to the canonical point is selected until the distance converges. In both examples in (a) and (b), they converged in two iterations but the number of iterations until convergence varies between zero to seven in the dataset. Also note that a sample image can be closest to one canonical point at an iterations but be closer to other at later iterations. The plots in the lower rows demonstrate the decrease in the euclidean distances. The insets visualize the same transformations applied to the images through a prototypical image.

5.3 Canonical scene images

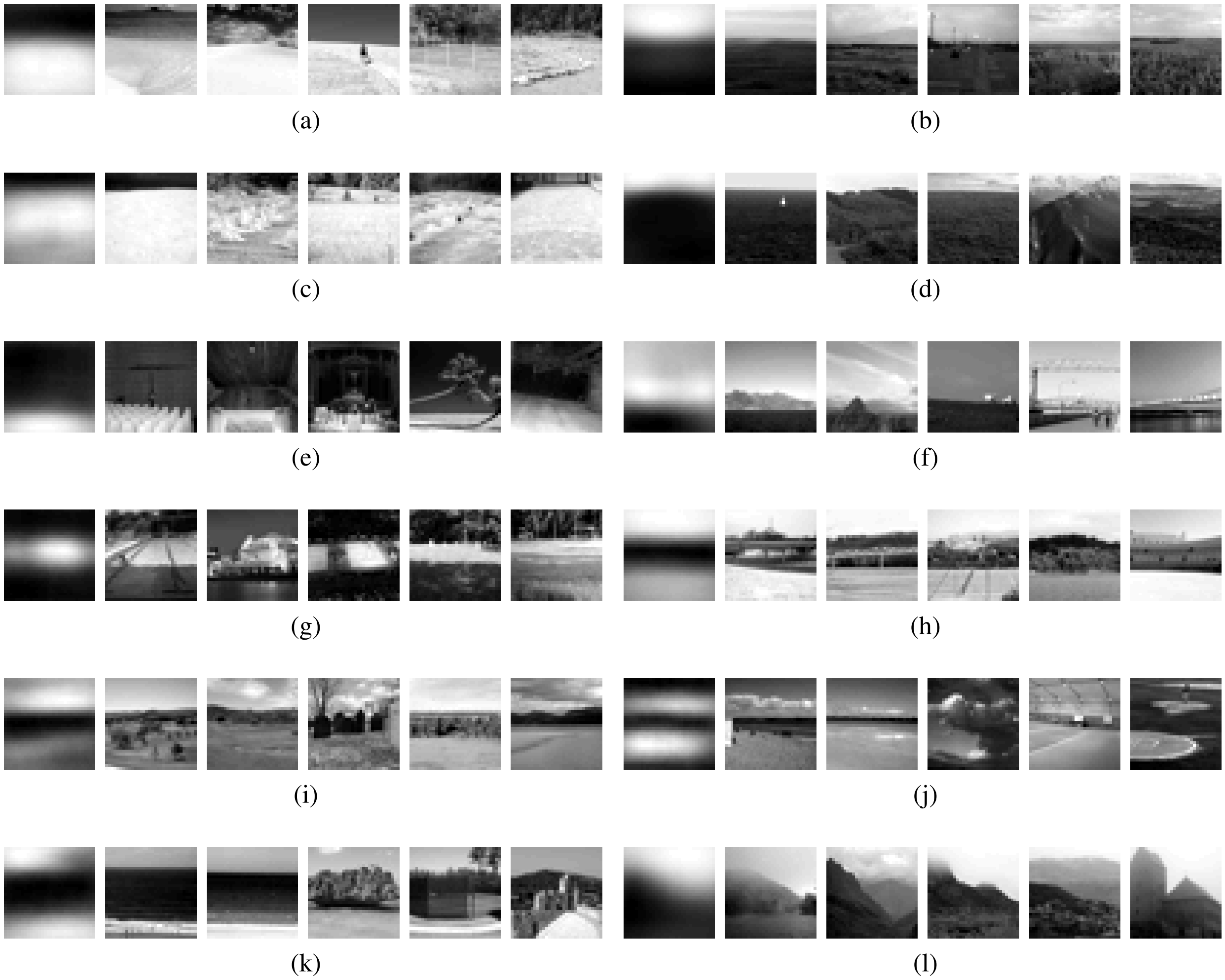
I applied the image normalization procedure to the SUN database [75]. I used grayscale scene images with the resolution of 36×36 and measured the distance between the center 32×32 regions for boundary issues in applying the transformations. The number of canonical points $\mathbf{A}$ is dependent on the threshold τ; τ set to an extremely small value would result in all the samples in the dataset as the canonical images and an extremely high value would result in one canonical image which is not informative. I set the τ as 10 for 32×32 scene images whose pixel values ranged from 0 to one because this value results in perceptually similar scene images to be assigned to the same canonical clusters and also return reasonable number of clusters. I exclude the canonical points with few images assigned to them (n_l) from further analysis. The resulting 146 canonical images covered 90% of data points and the rest were unassigned to any canonical images. I assigned the rest to default category and assumed they were derived from a distribution centered around zero.

Figures 5.3–5.4 show samples of canonical images emerged from the image normalization process in the first column together with scene images assigned to the them. Note that the canonical images indeed capture the common structures shared in the assigned images; horizontal lines located at multiple vertical positions (Figure 5.3a–5.3f), horizontal bands prominent in distant scenes (Figure 5.3g – 5.3h), vertical structures (Figure 5.4a, 5.4c and 5.4d) and converging lines (Figure 5.4n – 5.4q). As the canonical images are acquired by the weighed sum of scene images (line 13 in Algorithm 1) the high frequency structures are canceled out and only the low spatial frequency structures remain. The fact that some pairs of canonical images (Figures 5.3a and 5.3b, 5.3c and 5.3d, 5.3e and 5.3f) have similar structures in reversed phase suggests that even the scenes with the similar overall layouts are centered around distinct canonical images forming separate clusters.

As only the euclidean distance between images and canonical images was used in the image normalization process, there were variations between the images assigned to a same cluster around a canonical image. However, it is worth noting that the natural clusters formed by the image normalization process reflects the hierarchical semantic scene categories in SUN database; each cluster was dominated by scene images of certain semantic categories especially at the high and mid level categories (Figure 5.5). The SUN database is hierarchically organized at three levels; the high level categories subdivide the whole dataset into indoor, outdoor and outdoor-manmade scenes (Semantic categories 1,2 and 3 in Figure 5.5a). The mid level categories are based on the functions of scenes; cultural, industrial, forest or fields, home or hotel, etc. (Semantic categories 1–16 in Figure 5.5b). The leaf level categories are selected by the Wordnet terminology and search engine images [75] (Semantic categories 1–397 in Figure 5.5c).

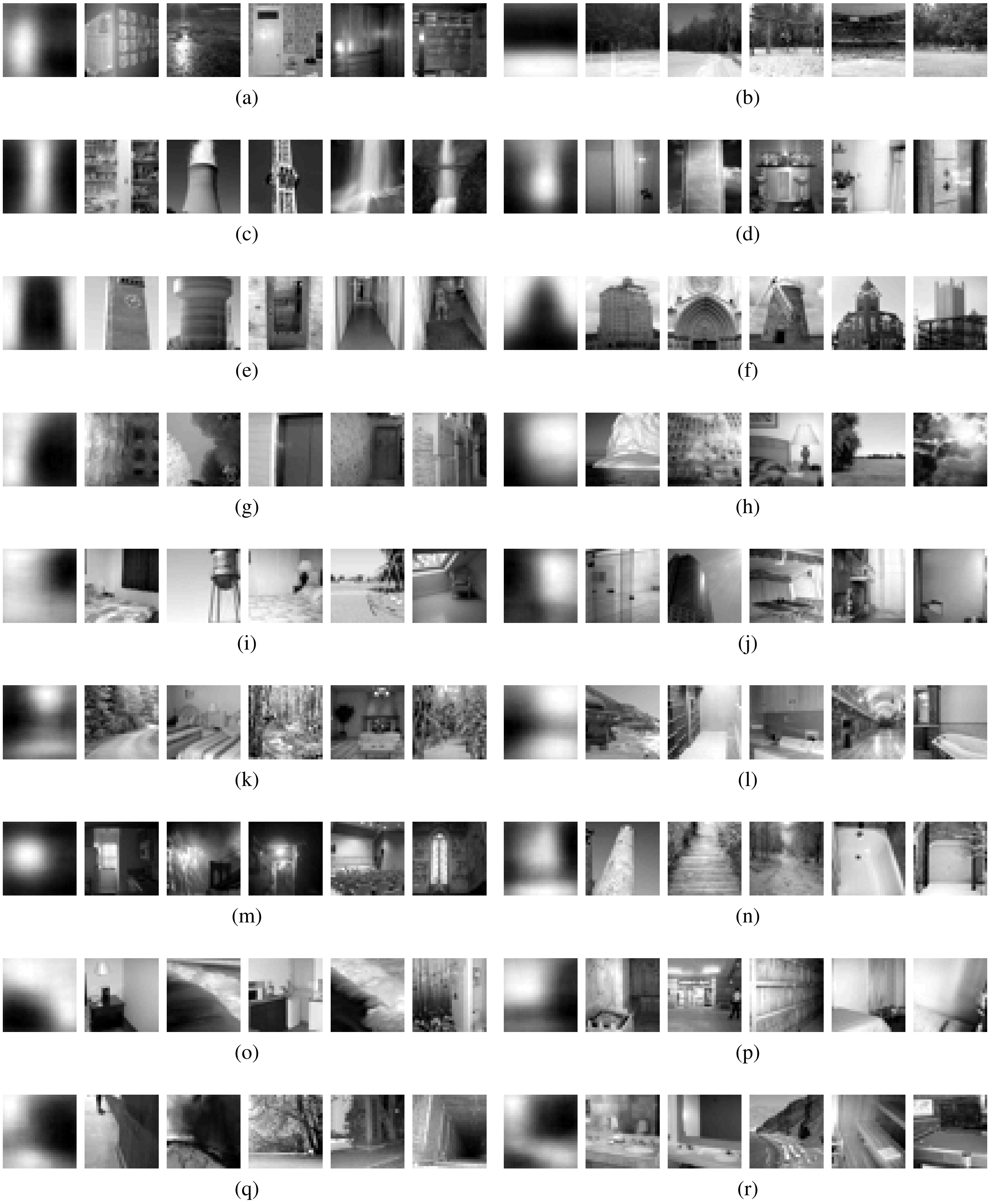
Figure 5.6 shows subsets of canonical images dominated by one of the three high-level categories. Note that the canonical points well represent the prominent structures in each high level category. The ones dominated by the indoor scenes capture the structures of indoor scenes defined by walls and floors with foreground objects (Figure 5.6a); the ones dominated by the outdoor natural scenes capture the horizontal line structures that are prominent among spacious and open outdoor natural scenes (Figure 5.6b). Outdoor manmade scenes which consist of open and spacious natural backgrounds and manmade elements such as buildings in the foreground are clustered around the canonical images shown in Figure 5.6c. This pattern holds at the mid-level semantic categories as well (Figure 5.7). In Figure 5.6–5.8, the numbers above each canonical scenes indicate the ratios of each semantic categories among the image assigned to each canonical image in the image normalization process.

At leaf level with 397 semantic categories, the correspondence between each semantic category and natural clusters become more subtle (Figure 5.5c). Most of the clusters around canonical images consist of several leaf-level categories without any dominant categories. However, the canonical scene images relatively dominated by certain leaf-level categories do show the common structures in the semantic categories (Figure 5.8). Another interesting observation is that some pairs of distinct leaf level categories share very similar distribution of scene images along the clusters around canonical scene images. The pairs of leaf-level scene categories which have the most similar distributions are shown in Figure 5.9. Note that although the leaf level scene categories are distinct at the leaf level, the scene categories share similar scene images at the holistic level; church outside and castle, living room and bedroom, casino indoor and bar, stage indoor and discotheque.



(a) (b) (c) (d) (e) (f) (g) (h) (i) (j) (k) (l)

Figure 5.3: Canonical images in the first columns and normalized scene images to the corresponding canonical images. See text for more detail.



(a) (b)

(c) (d)

(e) (f)

(g) (h)

(i) (j)

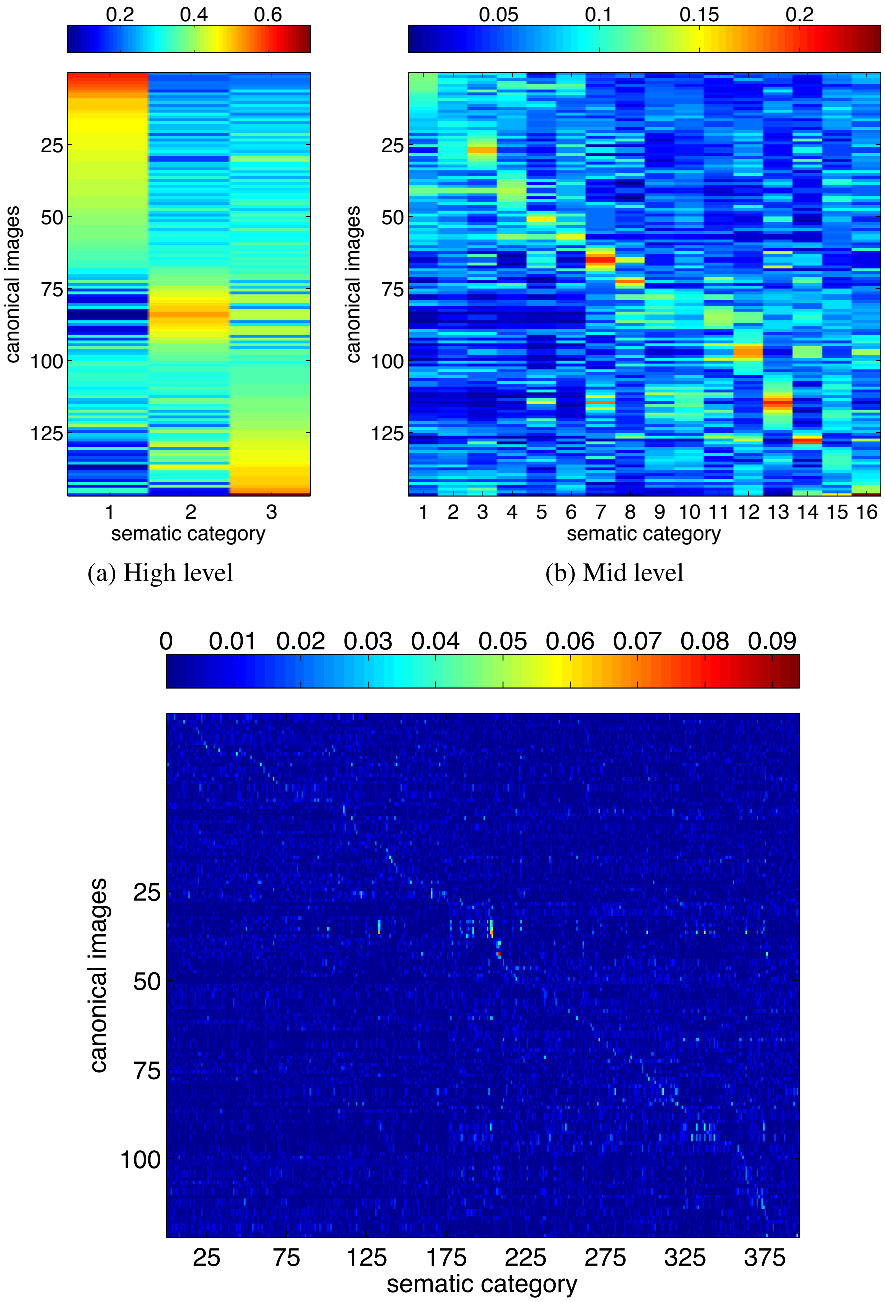
(k) (l)

(m) (n)

(o) (p)

(q) (r)

Figure 5.4: Canonical images in the first columns and normalized scene images to the corresponding canonical images. See text for more detail.

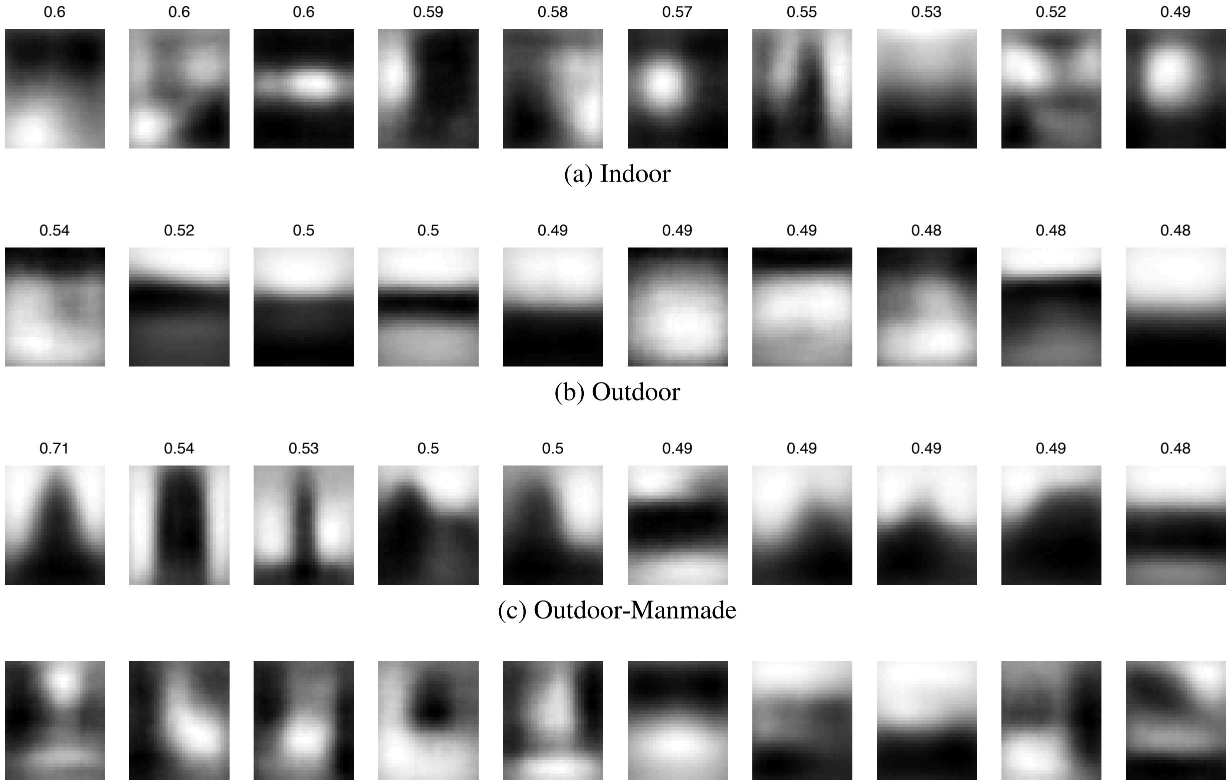


(a) High level

(b) Mid level

(c) Leaf level

Figure 5.5: Ratios of semantic categories assigned to each cluster around its canonical image. In each panel, the rows correspond to each of 147 clusters around each canonical point and the columns to categories at each level. The scale indicates the relative fraction of each semantic category compared to the the categories of all the scenes assigned to a canonical cluster. For example, in (b), a value of 0.2 for the canonical image 128 for mid-level semantic category 14 indicates that 20% of the scenes in that canonical cluster are examples of mid-level semantic category 14, Industrial and construction scenes (Figure 5.7n)



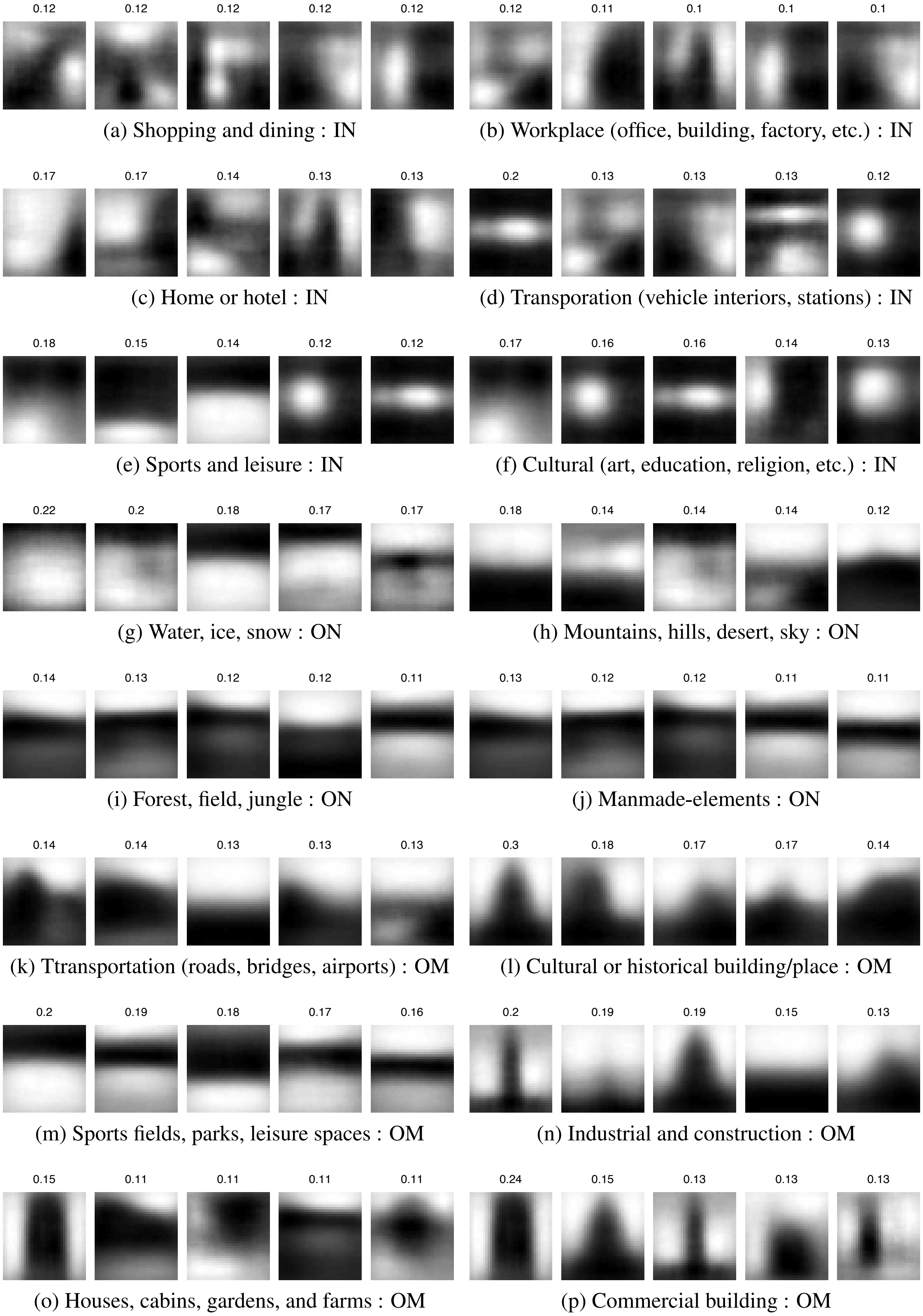
(a) Indoor

(b) Outdoor

(c) Outdoor-Manmade

(d) Common

Figure 5.6: Canonical images whose clusters are dominated by (a) Indoor (b) Outdoor and (c) Outdoor-Manmade images. (d) shows canonical images whose clusters were assigned scene images from all of the three categories. The numbers above each canonical image correspond to the ratio of each high-level scene categories in the clusters centered around each canonical images by the image normalization process. For instance, 60% of scene images assigned to the first canonical scene in (a) were indoor scene images.



(a) Shopping and dining : IN

(b) Workplace (office, building, factory, etc.) : IN

(c) Home or hotel : IN

(d) Transporation (vehicle interiors, stations) : IN

(e) Sports and leisure : IN

(f) Cultural (art, education, religion, etc.) : IN

(g) Water, ice, snow : ON

(h) Mountains, hills, desert, sky : ON

(i) Forest, field, jungle : ON

(j) Manmade-elements : ON

(k) Ttransportation (roads, bridges, airports) : OM

(l) Cultural or historical building/place : OM

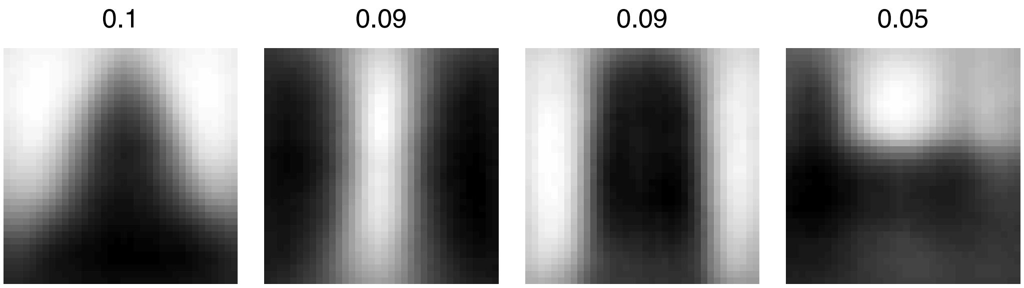
(m) Sports fields, parks, leisure spaces : OM

(n) Industrial and construction : OM

(o) Houses, cabins, gardens, and farms : OM

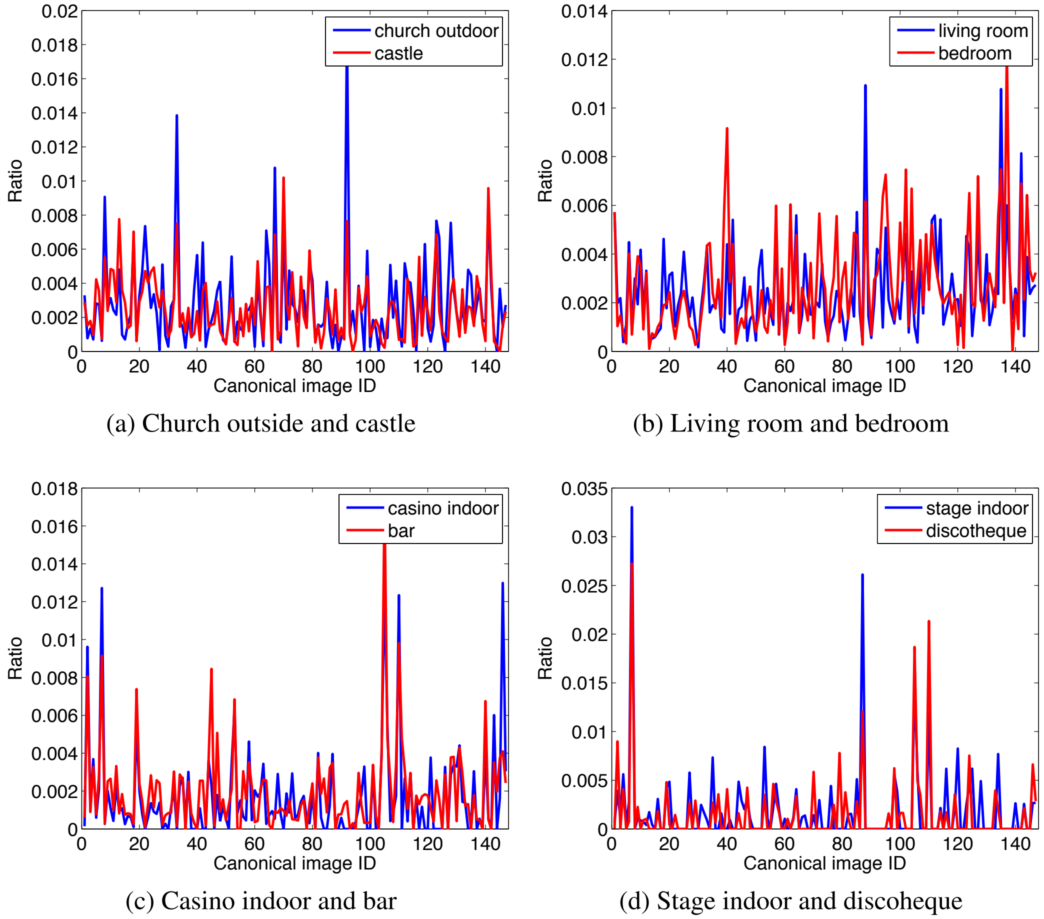
(p) Commercial building : OM

Figure 5.7: Canonical images whose clusters are dominated by each of 16 mid-level semantic categories. Acronyms stand for the high-level semantic categories; Indoor (IN), Outdoor-natural (ON) and Outdoor-manmade (OM).



(a) Pagoda (b) Waterfall plunge (c) Doorway outdoor (d) Street

Figure 5.8: Canonical images whose clusters are dominated by some of 397 leaf-level semantic categories. The figures above the images indicate the ratio of each leaf-level category in the cluster around the canonical image. The cluster around the canonical image shown in (a) included high ratios of Tower and Temple (South Asia) scenes.



(a) Church outside and castle (b) Living room and bedroom

(c) Casino indoor and bar (d) Stage indoor and discoheque

Figure 5.9: The distributions among canonical clusters of distinct leaf level scene categories whose scene images have common holistic properties are similar. For examples, Indoor Stage and Discotheque are separate scene categories at the leaf level semantic scene categories but the images belonging to each semantic category are similar especially at the holistic level. Their distributions among canonical clusters also share similar patterns.

5.4 Covariance units

I adopted the canonical images emerged in the image normalization process as means around which scene images are drawn and trained the density coding model discussed in Chapter 3. The extended model can be formulated as follows,

$$\mathbf{x}|\mathbf{a}_l, \mathbf{y} \sim \mathcal{N}(\mathbf{a}_l, \mathbf{C}(\mathbf{y})) \tag{5.1}$$

where the covariance matrix $\mathbf{C}(\mathbf{y})$ was formulated as

$$\log(\mathbf{C}(\mathbf{y})) = \sum_j y_j \mathbf{A}_j = \sum_j y_j \sum_k w_{j,k} \mathbf{b}_k \mathbf{b}_k^T \tag{5.2}$$

where y_j corresponds to the j th element of $\mathbf{y}$. Note that even though the conditional distributions are centered around different canonical images, for reasons of learnability, $\mathbf{b}_k$s and $w_{j,k}$ that define the covariance are shared among all canonical clusters.

The prior probability of clusters centered around each canonical image $\mathbf{a}_l$ was assumed to be uniform and to enforce the model parameters to learn a compact representation of covariance matrices, the model uses a laplacian prior on $\mathbf{y}$,

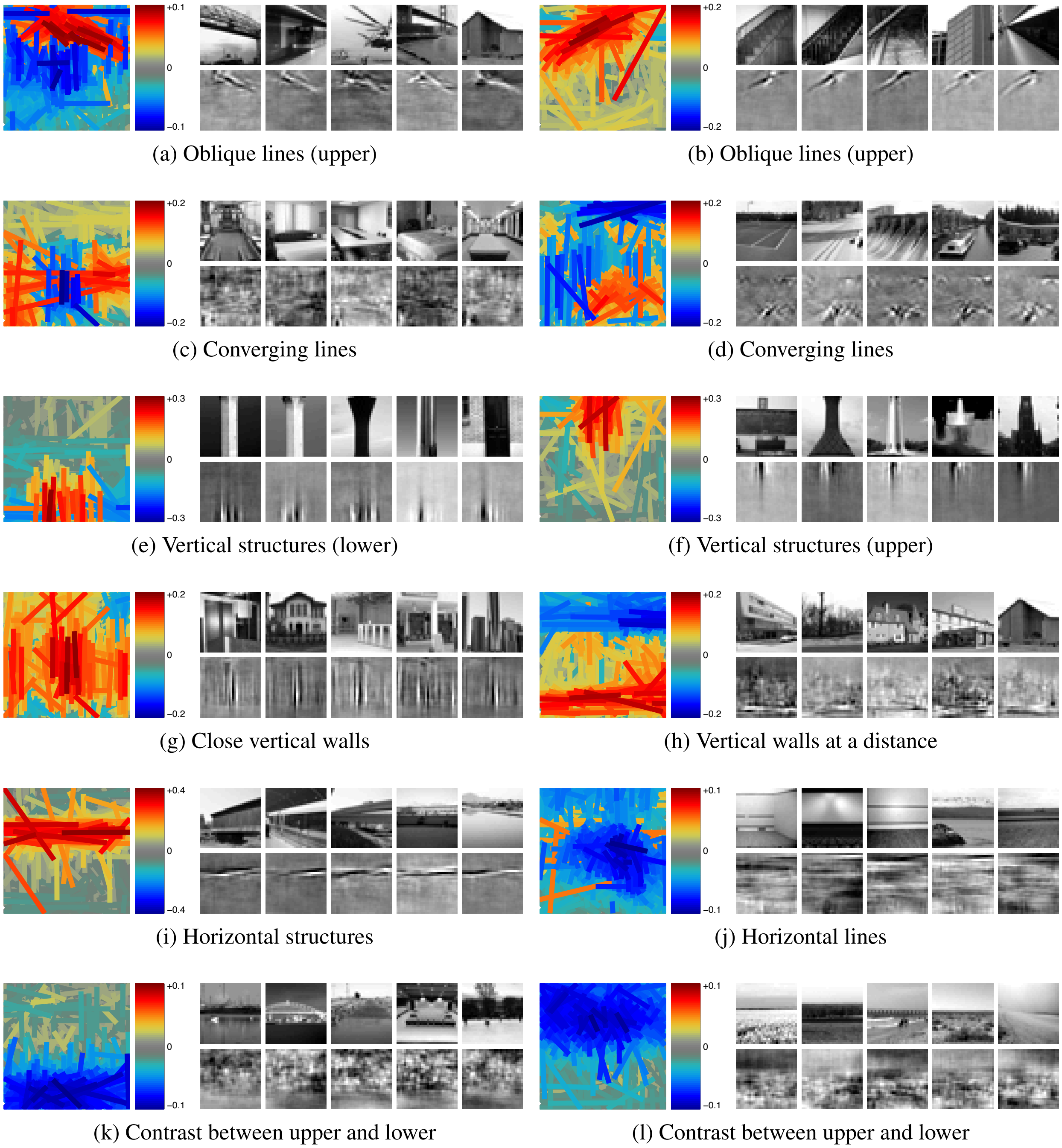
$$\log p(\mathbf{y}) \propto -\sum_j |y_j| \tag{5.3}$$

I used the same optimization techniques with those discussed in Chapter 3.

Figure 5.10 visualizes the covariance units learned with the canonical images from Section 5.3 incorporated into the model. For each unit $\mathbf{A}_j$, the scene images which was most active to the unit, i.e., which has the highest value of y_j, and the synthetic images generated by using each unit as a covariance matrix are displayed. The covariance units still capture the global structures prominent in scene images. Note that even though certain images are active to the same covariance unit such as the one shown in Figure 5.10e, they have reversed sign. For instance, the second and the third most active scene images to the unit do share the vertical structures in the upper part of the scenes but one describes a bright tower in a dark background and the other a dark tower in a light background. The the pair of the second most active scene image and the fifth most active scene image in Figure 5.10e also illustrates this. Also worth noting is that some units encode patterns that were not captured by the density coding model with fixed zero mean assumptions; the unit in Figure 5.10k encodes the high frequency structures in the upper side of images and the opposite in the lower side. Figure 5.10l captures the opposite patterns to the unit in Figure 5.10k.

One natural question would be to what extent do the means and the covariances each capture the patterns in the data and how they are related. Even though the prior distributions of clusters defined by each canonical image and the covariance latent variables are modeled as independent, the empirical distribution of the latent variables in each cluster suggests covariance units are more active in certain clusters than others. To illustrate this relationship, I present the canonical images whose clusters show the most active responses to each covariance units in Figure 5.11. For this, I pick clusters whose mean responses to a certain unit $\mathbf{A}_j$, y_j, is highest. The covariance units are most active when the distributions are centered around canonical images which share the similar layout with them. For instance, the unit in Figure 5.11a is more active among scene images centered around canonical images with horizontal lines located on a similar vertical position as that in the covariance unit. While the canonical images share similar horizontal structures, the phases of the horizontal lines varies. This suggests that the covariance units encode the contrast whereas the canonical images gather scene images with similar overall light patterns.

Under this framework, an image $\mathbf{x}$ normalized to a canonical image $\mathbf{a}_l$ is more likely in the cluster around the canonical image than any others; the conditional likelihood of the image in the cluster around the canonical image is higher than the conditional likelihood in any other canonical clusters. Based on this observation, once the canonical images and the clusters around them were discovered, they were not further optimized while training the model parameters governing the conditional covariance matrices. The model can be extended by incorporating the probability of an image $\mathbf{x}$ to be assigned to a canonical image $\mathbf{a}_l$ into Eq.5.1. Under such framework, the means and the covariance units can be alternatively estimated in an expectation-maximization fashion. The iterative estimations could significantly change how certain image properties are split between means and covariances.



(a) Oblique lines (upper)

(b) Oblique lines (upper)

(c) Converging lines

(d) Converging lines

(e) Vertical structures (lower)

(f) Vertical structures (upper)

(g) Close vertical walls

(h) Vertical walls at a distance

(i) Horizontal structures

(j) Horizontal lines

(k) Contrast between upper and lower

(l) Contrast between upper and lower

Figure 5.10: Covariance units $\mathbf{A}_j$s. Scene images with the highest activity to the units (upper rows). Synthetic scenes generated from the multivariate Gaussian distributions with each unit as covariance matrices.

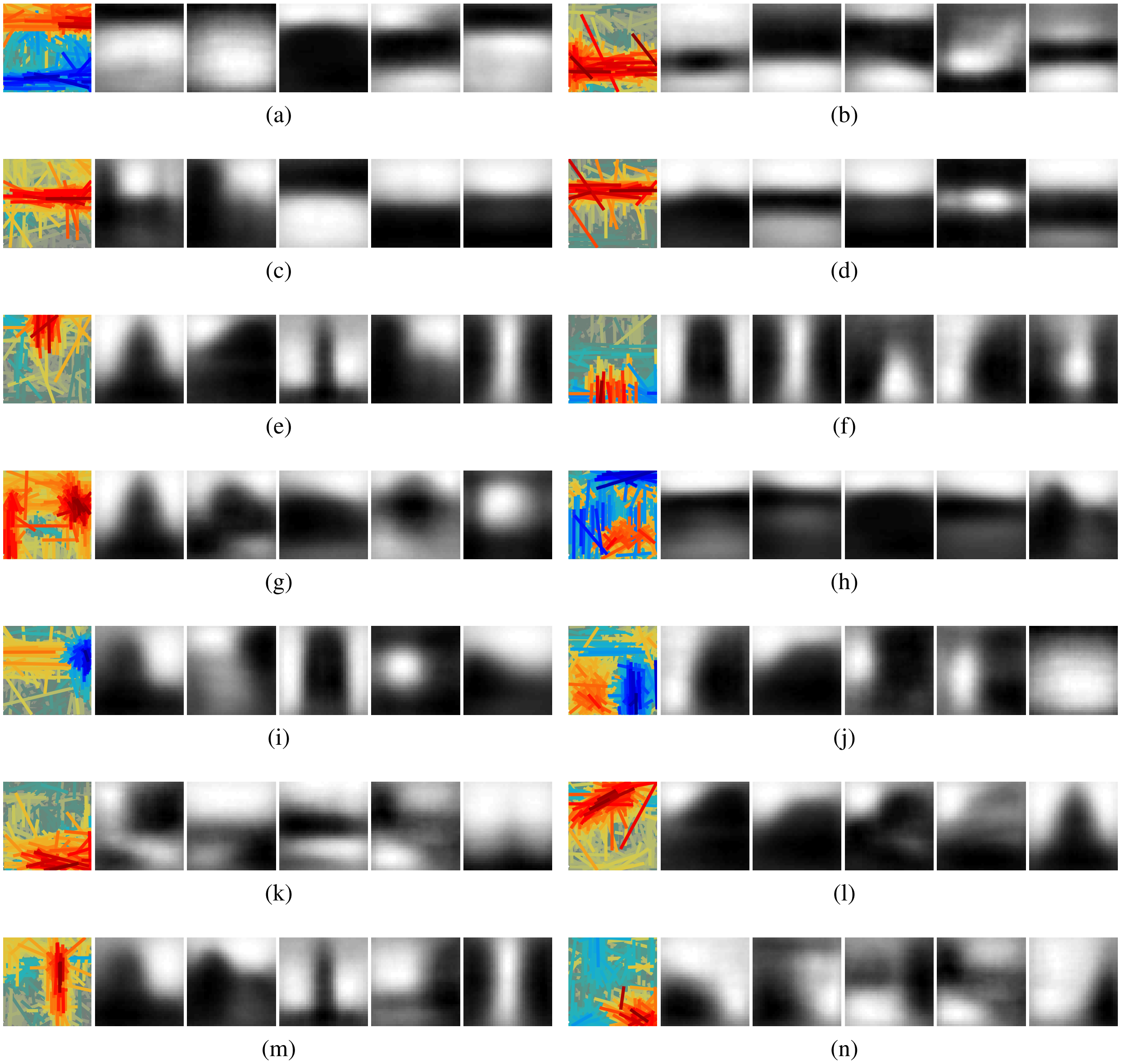


Figure 5.11: Left: visualization of covariance units $\mathbf{A}_j$. Right: scene images assigned to the canonical points shown on the right columns have active responses to the corresponding covariance units.

5.5 Synthetic images with the extended model

In this section, I discuss the benefits of the extended model in terms of image synthesis. For a new scene image, I assign it to one of the canonical images $\mathbf{a}_l$ by the image normalization process and infer the latent variables $\hat{\mathbf{y}}$ for covariance matrices $\mathbf{C}_{\text{mean+cov}}(\hat{\mathbf{y}})$ as shown in Eq. 5.1. With the mean and the covariance matrix, I can draw random samples $\mathbf{s}_{\text{mean+cov}}$ from the corresponding multivariate Gaussian distributions by

$$\mathbf{s}_{\text{mean+cov}} = \mathbf{a}_l + \mathbf{L}\mathbf{r} \tag{5.4}$$

where $\mathbf{L}$ corresponds to the Cholesky decomposition of $\mathbf{C}_{\text{mean+cov}}(\hat{\mathbf{y}})$ and $\mathbf{r}$ a random Gaussian vector drawn from zero mean with the identity matrix as the covariance matrix.

I compare the synthesis performance of the extended model to the original density coding model with zero mean assumption and the covariance matrices specialized for each image

$$\mathbf{s}_{\text{cov}} = \mathbf{L}\mathbf{r} \tag{5.5}$$

where $\mathbf{L}$ refers to the Cholesky decomposition of $\mathbf{C}_{\text{cov}}(\hat{\mathbf{y}})$ whose parameters were learned with the zero mean assumption in Chapter 3.

Lastly random samples by adding Gaussian noise (i.e., assuming that the covariance matrix equals to an identity matrix) to the canonical images $\mathbf{a}_l$ corresponding to each new images were considered.

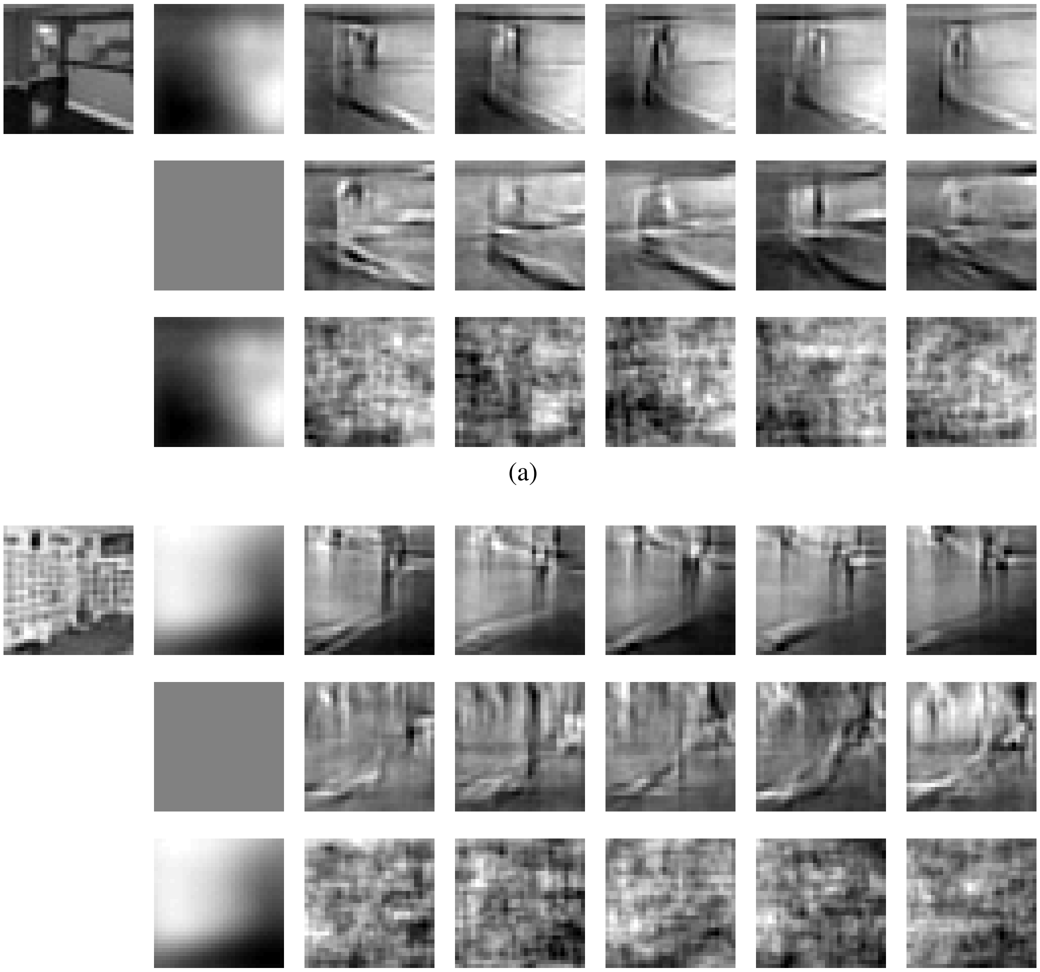
$$\mathbf{s}_{\text{mean}} = \mathbf{a}_l + \mathbf{I}\mathbf{r} \tag{5.6}$$

where $\mathbf{I}$ refers to the identity matrix.

As shown in Figure 5.12, when a random sample is drawn around the corresponding canonical point, it fails to capture the intrinsic patterns when the covariance information is ignored. As discussed in Chapter 3, the covariance only model sometimes fails to capture long-range structures and even when they do, they only capture the existence of contrasts. When the canonical points are adopted, they anchor the synthetic images around canonical images which carries the overall phase of a scene image, letting the synthetic images closer to the original images. Also, the extended model results in significantly better mean squared error ($p < 0.01$) then both the the covariance only and the mean only conditions. Among the covariance matrices and the means, the covariances condition does significantly better in terms of MSE ($p < 0.01$). Note that the dimensionality of the covariance latent variables $\mathbf{y}$ was 60, which is less than 6% of the dimensionality of the resolution of input. A single cluster selected from 147 can be represented with less than 8 bits which does not suffice to represent more than a pixel even after JPG compression.

Condition	MSE
mean + covariance	0.027
covariance	0.032
mean	0.035

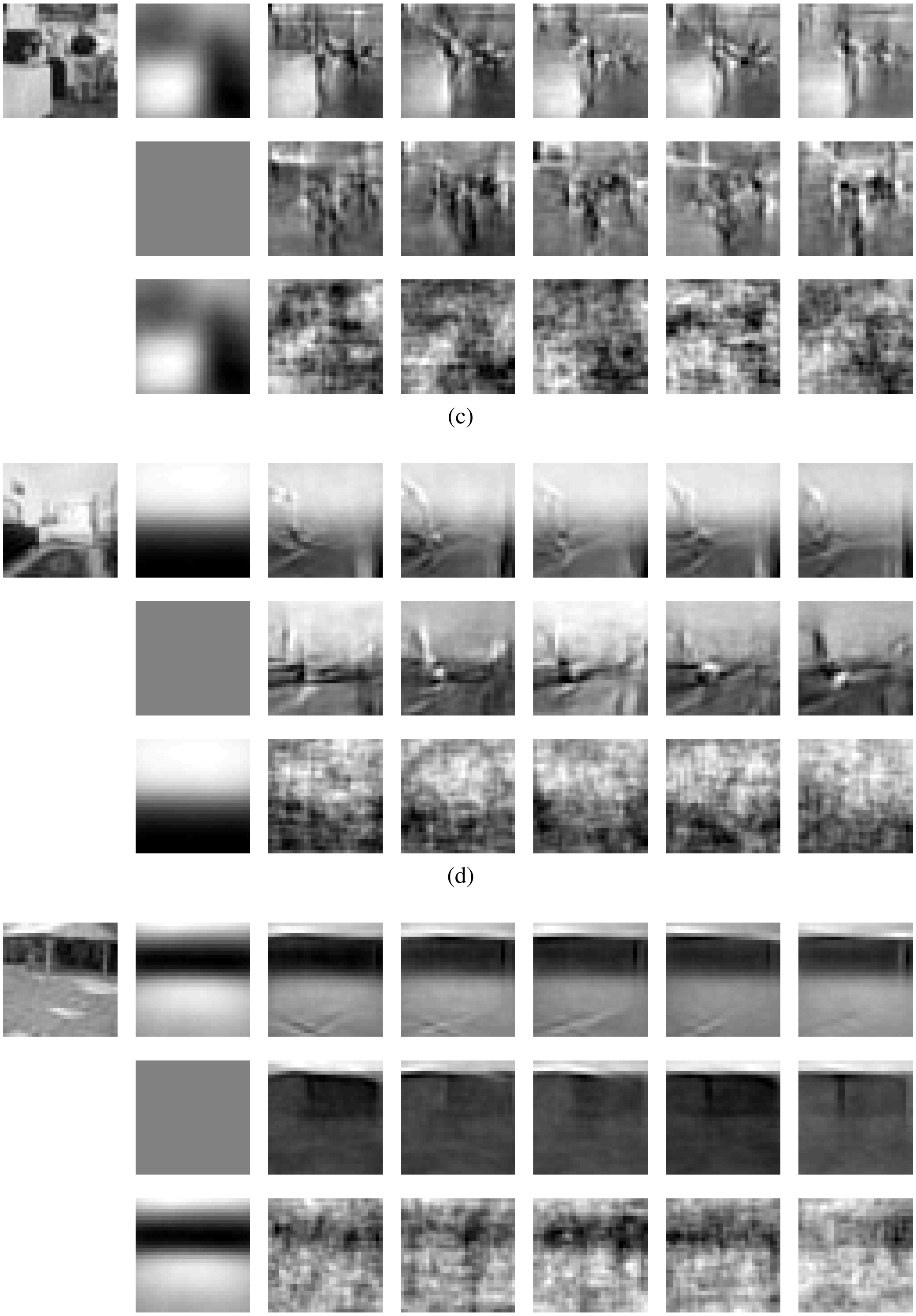
Table 5.1: Mean squared error between the synthetic images generated with three different models and the original images. The result is obtained from synthesizing scene images of 5000 samples. mean + covariance conditions significantly outperforms the ones using only covariance matrices or means. The covariance only condition does significantly better than the mean only condition.



(a)

(b)

Figure 5.12: Synthetic images; mean + covariance (first row), covariance (second) and mean (third). The top row and the first column; original images. The second column shows the means of the multivariate Gaussian distributions for each condition.

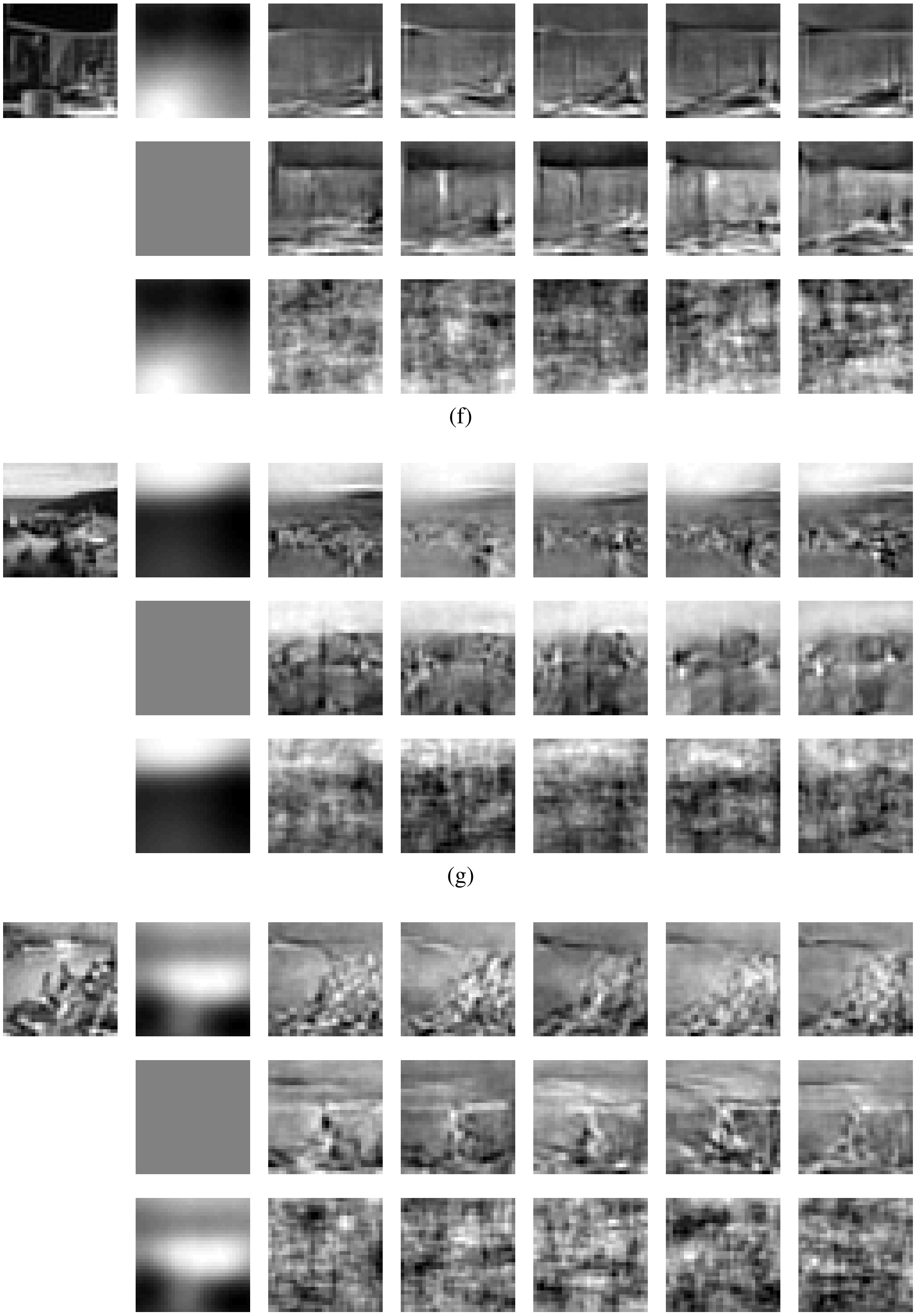


(c)

(d)

(e)

Figure 5.12: Synthetic images; mean + covariance (first row), covariance (second) and mean (third). The top row and the first column; original images. The second column shows the means of the multivariate Gaussian distributions for each condition.



(f)

(g)

(h)

Figure 5.12: Synthetic images; mean + covariance (first row), covariance (second) and mean (third). The top row and the first column; original images. The second column shows the means of the multivariate Gaussian distributions for each condition.

5.6 Image retrieval

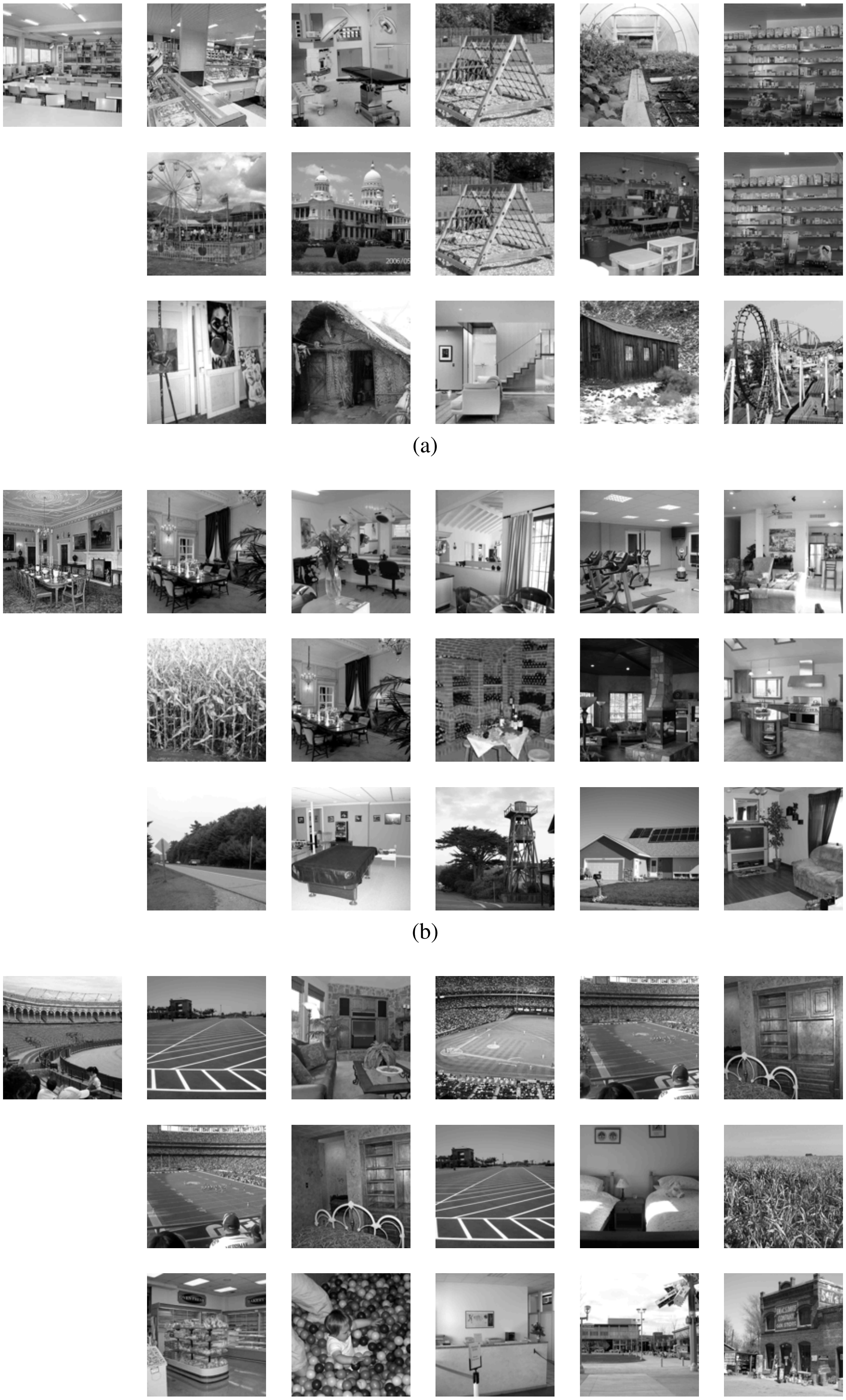
The probabilistic framework of the extended model allows to retrieve images based on the following similarity measure based on reciprocal likelihood of pairs of images; the sum of likelihood of one of two images based on the other's best distributions. For two samples, $\mathbf{s}$ and $\mathbf{t}$, I estimate the best multivariate Gaussian distributions optimized for each sample. When the set of mean and covariance matrix optimized for $\mathbf{s}$ is $\{\mathbf{a}_s, \mathbf{C}(\hat{\mathbf{y}}_s)\}$ and $\{\mathbf{a}_t, \mathbf{C}(\hat{\mathbf{y}}_t)\}$ for $\mathbf{t}$, I use

$$\log p(\mathbf{t}|\mathbf{a}_s, \mathbf{C}(\hat{\mathbf{y}}_s)) + \log p(\mathbf{s}|\mathbf{a}_t, \mathbf{C}(\hat{\mathbf{y}}_t)) \tag{5.7}$$

as the similarity measure between $\mathbf{s}$ and $\mathbf{t}$. This measure considers the likelihood of one of the pair given the distributions optimized for the other and prevents the images near the canonical points from dominating the retrieval results. For a query image $\mathbf{s}$, I retrieved the scene images with the highest values of this probabilistic similarity measure.

I show examples of the retrieved images using the extended model together with the retrieval results based on the density coding model [29] and GIST features [47] in Figure 5.13. I used the SUN database as the image pool from which I retrieved the most similar images using 32×32 grayscale scenes. The results are ordered such that the results with the highest similarity measure values are shown in the first column and the fifth on the fifth column. For GIST, I retrieved images whose GIST features are most closest to the features of a query image in terms of the euclidean distance. Note that the extended model and the density coding model was trained on grayscale 32×32 images and the examples in Figure 5.13 are shown in 128×128 resolution only for the visualization purpose.

The extended model and the original density component model often retrieves similar results but the extended model returns images with better layout and semantic matches; in Figure 5.13a, the third and the fifth most similar images of the two models coincide but the extended model's most similar and the second similar scenes are perceptually and semantically closer to the query. As most of the training examples in SUN database describe scenes at the holistic level, the models adapt to their global structures and the local objects are naturally considered as noise since they are not very common among the training examples. Also, as I trained the model on the reduced resolutions of 32×32, the high-frequency structures that capture the characteristics of local objects or textures tend to be discarded. For these reasons, the models fail to retrieve similar scene images when the query images mainly consist of an object with the layout of scenes obstructed by it. For instance, as demonstrated in Figure 5.13h, the model fails to recognize the human in the middle and only captures its global layout retrieving scene images with vertical structures pointed at the top. In Figure 5.13i, the first and the third retrieved scene images describe waterfalls with similar layout to the query. However, the other retrieved scene images suggest that the model captures the overall pattern of the bright vertical structures in the middle of the images.

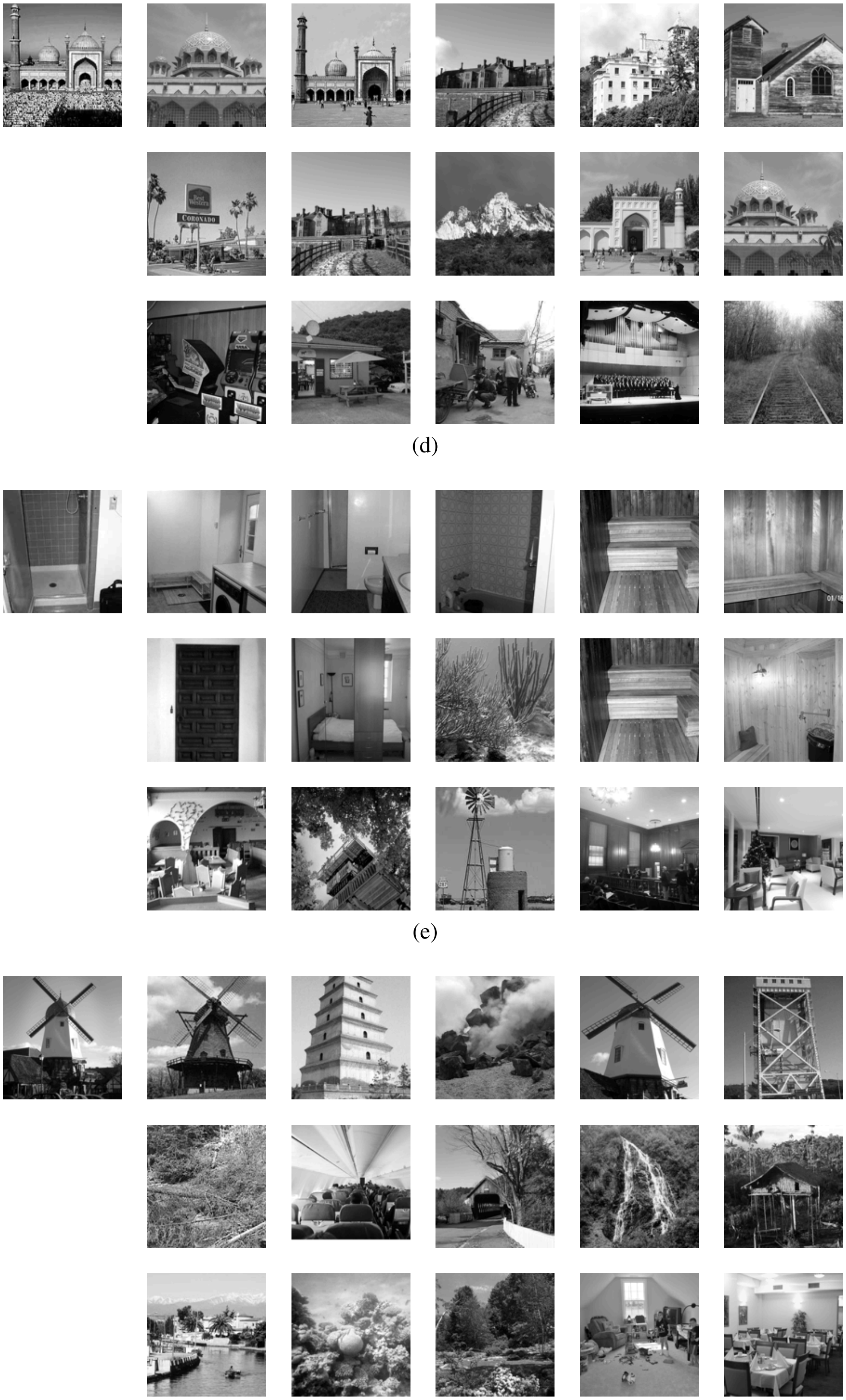


(a)

(b)

(c)

Figure 5.13: Retrieval: mean+cov (top row), cov (second row) and GIST (third row). See text for discussion.

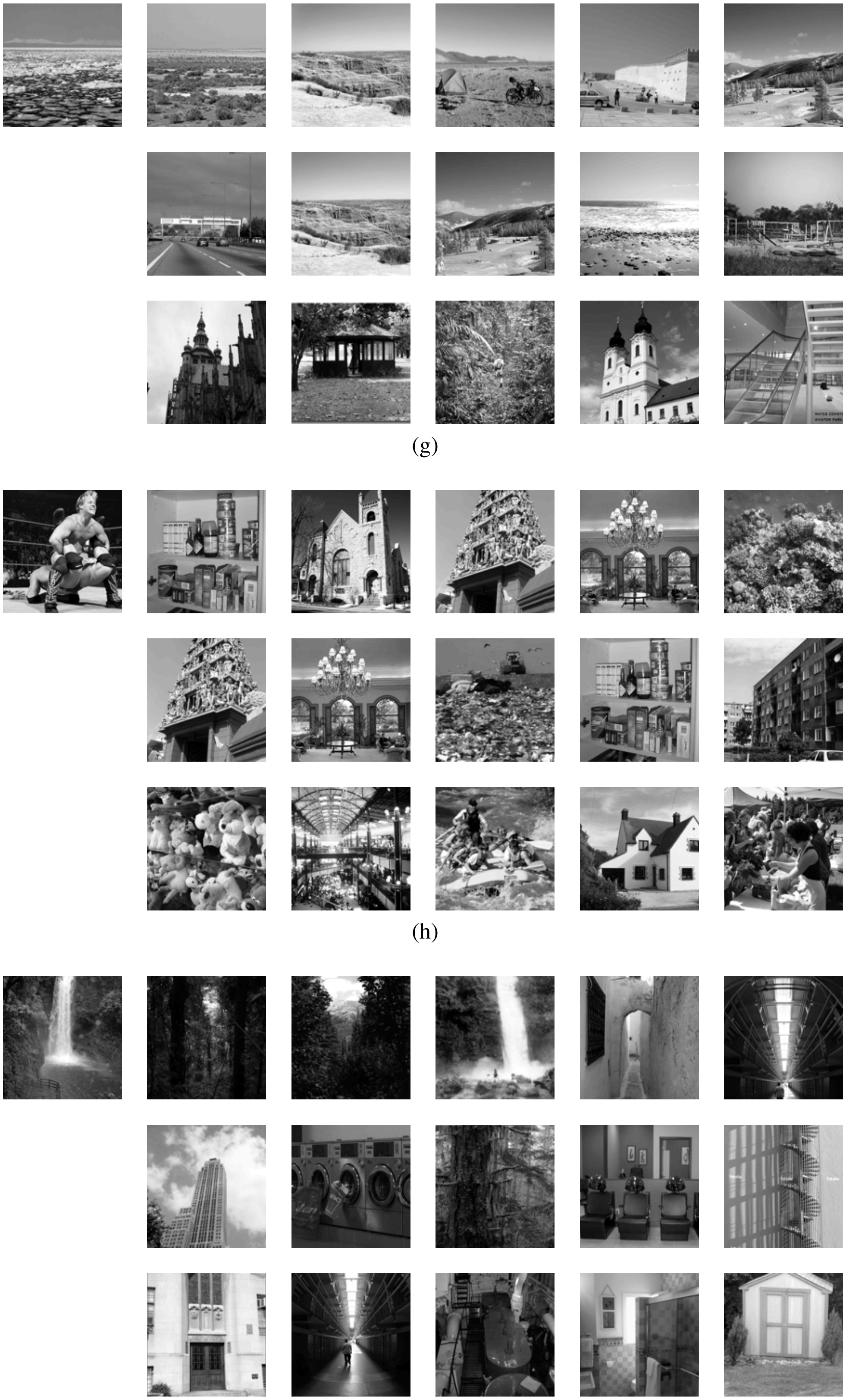


(d)

(e)

(f)

Figure 5.13: Retrieval: mean+cov (top row), cov (second row) and GIST (third row). See text for discussion.



(g)

(h)

(i)

Figure 5.13: Retrieval: mean+cov (top row), cov (second row) and GIST (third row). See text for discussion.

5.7 Conclusion

We propose a general framework for manifold learning with canonical points and the covariance structures. We developed a data normalization process for acquiring a representative set of canonical points that are close enough to the manifolds. This approach allows to tighten the original distributions of dataset around the canonical data points. When this model is applied to the whole scene images, the model parameters reveal that the overall light patterns are encoded by the canonical images and the covariance units the global contrast patterns. The new approach inherently learns the local manifold structures more closely and with more flexible assumptions and results in improved the image synthesis and the image retrieval.

Chapter 6

Conclusion

This thesis has explored learning the statistical properties of the whole scene images. This is the first approach on training adaptive representations on the whole scene level as far as I am aware of. Specifically, this thesis has focused on learning compact representations for encoding the holistic properties of scene images. Our results suggest that the more flexible and the more detailed models learn specialized structures with holistic properties that did not emerge with simpler models with more conservative assumptions. In addition, these specialized structures to the holistic structures of scene images predict the high level properties such as perceptual layouts or semantic categories. Chapter 2 concludes that letting the independent component model learn tailored representations for natural subcategories of scenes reveal the holistic basis functions. Chapter 3 further explores when each scene image is assigned with one distribution specialized for itself and discusses the efficient representations for their spatial layout properties. Chapter 5 extends the previous approach and learns the local manifold structures by anchoring the distributions around canonical scene images.

Learning from data by training adaptive representations has been very popular research topic recently. The probabilistic distributions developed in such studies derive from Restricted Boltzmann Machines which model the joint distributions of the data and latent variables [5]. Other popular approaches are to build layers of filters such that the reconstruction error in the bottom layers is minimized. These studies are based on layers of filters either by reducing the dimensionality of the input by pooling [77] or by training a lower dimensional filters adapted to the latent or hidden variable from the lower layers [35]. The layer-based models allow to train probabilistic models on high resolution images and have been successful for many applications from document recognition [34] to object recognition [52].

My approach differs from the general deep learning research first in that the probabilistic models I use directly constraint the latent or hidden variables to be sparse; the cores of the models are based on the conditional distribution of data given the latent variables and therefore I can directly apply sparse priors on the latent variables. This allows the model parameters to represent the properties of the data much more efficiently and compactly. On the other hand, most of the probabilistic models adopted in the deep learning literatures usually start with the joint distributions of data and hidden variables and therefore the models require complicated procedures if the sparse representations are desired. Another difference is that I train hierarchical models whose parameters have layered structures based on weight sharing. This approach allows to dramatically reduce the dimensionality of the latent variables; the latent variables for encoding an image of 32×32 in Chapter 3 and 5 were of dimensionality which corresponds to 6% of the input data. This contrasts with the fact that the size of the models in the deep learning literature tends to be much larger. Lastly, the image normalization technique discussed in Chapter 5 covers wider variety of invariances such as rotations, scaling and translations whereas the deep learning models dependent on pooling algorithms are invariant to local translations.

The layer-based algorithms are inspired by the observation that the single layer algorithms such as sparse coding end up learning localized edge-like structures [50, 52]. However, my observation suggests that when the dataset itself consists of underlying regularities like scene images due to 3D shape of the world or when the dataset can be normalized such that the regularities in the dataset is enhanced without destroying the contents of the dataset, the models which do not build up by stacking up layers can reveal model parameters with holistic structures. This is observation especially holds when the probabilistic models are allowed to learn specialized distributions for natural clusters of datasets.

This thesis has its limitation in that it studies the properties of rather low resolution scene images. However, as the previous psychophysics studies suggest [68], I observe that even the information contained in 32×32 scene images is predictive of the high-level properties of the scenes such as the spatial layouts or semantic categories. This may be because the holistic structures can be captured by the low-spatial frequencies whereas the local information of scene images are much more irregular and spread over space. However, for more detailed discriminations between semantic categories at the leaf levels, such as discriminating between church outside and cathedrals, it is required to encode high-frequency information on top of the holistic properties [51].

One of the future directions for studying the statistical properties of natural visual scenes would be to extend the current framework to higher resolution images. Chapter 4 explored this direction in terms of learning the statistical properties of separate bands of Laplacians. However, encoding all the local structures of scene images and computing the probabilistic similarity using all of them, especially for large size images, are computationally expensive. In addition, some local regions of scene images, such as sky, are common between scene images and therefore are not highly informative of a scene for discriminating it from others. The fact that humans saccade only to informative regions of a scene rather than uniformly spreading their attention supports the idea that encoding only such informative local regions is sufficient for subsequent processing of a scene.

Objects within scenes with similar spatial layouts tend to be located approximately at similar positions. For instance, as illustrated in Figure 6.1, in street scenes with converging lines, salient objects are usually located on the streets, but it is less likely that objects are located in the sky regions or on the side building walls. A representation that well captures the global structures of scene images among complex structures across multiple scales and locations should be informative of locations of salient objects.

Motivated by these observations, we suggest the future direction for investigating the role of global structures discussed in this thesis as the context for predicting the "informativeness" of the local regions in a probabilistic framework. Examples of "informative" local regions are illustrated in Figure 6.1. The regions in the red boxes in the first two examples do not add much to the spatial layout properties of each scene; even though the first one does have strong edge structure, this edge aligns well with the global spatial layout of the scene. Therefore registering the local structures of these regions will not be informative of the scenes. On the other hand, the blue boxed regions contain local structures that cannot be inferred just from the scenes' spatial layout, and hence are informative. Encoding only the local structures from the informative local regions has be reported to be useful for object priming [69], content based image retrieval [43, 62] as well as image segmentation [2].

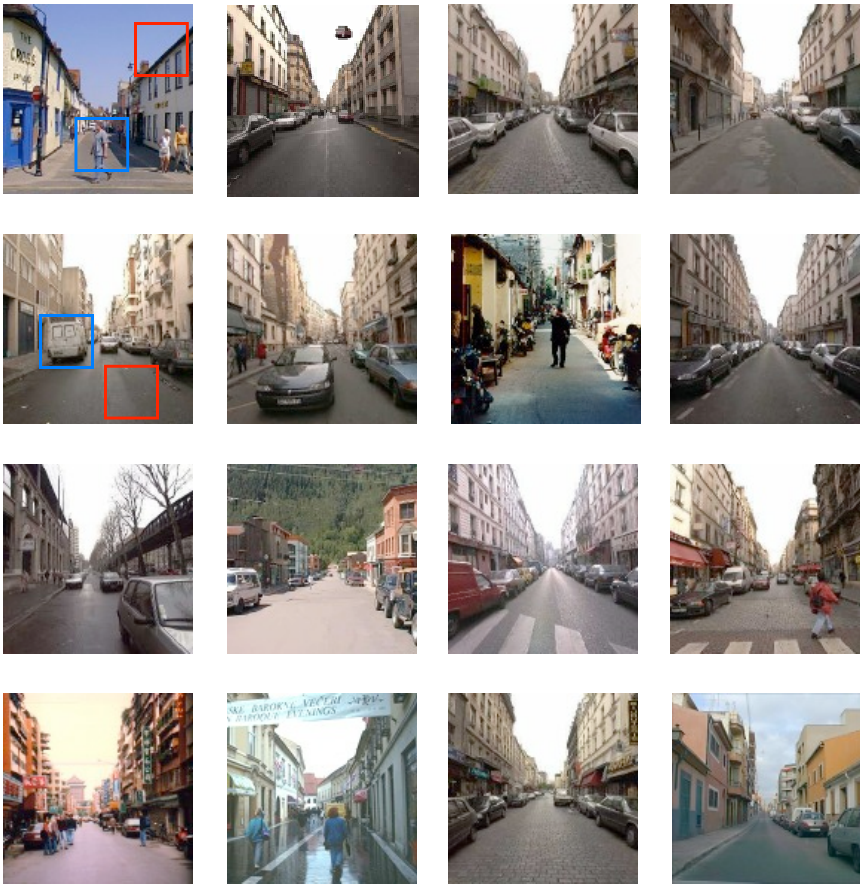


Figure 6.1: Street scenes with similar spatial layouts. Objects are located on similar positions in different scenes but similar spatial layouts. Boxes in the first two examples illustrate non-informative (red) and informative (blue) local regions.

As was demonstrated in this thesis, learning the compact and efficient representation reveals the prominent and common patterns in the large-scale dataset. Another interesting future direction of this thesis is to apply the frameworks to other image types such as face or objects or even to other domains such as gene expressions or social network frameworks. This framework is especially useful because it allows the high-dimensional data to be captured by much lower dimensional latent variables. Also, the probabilistic framework is a great way of defining a similarity measure based on their statistical properties which is often a very difficult task for high-dimensional data.

Bibliography

[1] P. A. Absil, R. Mahony, and R. Sepulchre. *Optimization algorithms on matrix manifolds*. Princeton University Press, 2008. 3.2.1, 3.2.3

[2] Radhakrishna Achanta, Francisco Estrada, Patricia Wils, and Sabine Süsstrunk. Salient region detection and segmentation. In *Computer Vision Systems*, pages 66–75. Springer, 2008. 6

[3] Amr Ahmed, Kai Yu, Wei Xu, Yihong Gong, and Eric Xing. Training hierarchical feed-forward visual recognition models using transfer learning from pseudo-tasks. pages 69–82, 2008. 5.1

[4] Doru C Balcan and Michael S Lewicki. Adaptive coding of images via multiresolution ica. In *IEEE ICASSP*, pages 1021–1024, 2009. 4.1

[5] Yoshua Bengio, Pascal Lamblin, Dan Popovici, and Hugo Larochelle. Greedy layer-wise training of deep networks. *Advances in neural information processing systems*, 19:153, 2007. 6

[6] Anna Bosch, Andrew Zisserman, and Xavier Munoz. Representing shape with a spatial pyramid kernel. In *ACM international conference on Image and video retrieval*, pages 401–408. ACM, 2007. 1, 4.1, 4.5.1

[7] Y-L Boureau, Francis Bach, Yann LeCun, and Jean Ponce. Learning mid-level features for recognition. In *IEEE CVPR*, pages 2559–2566, 2010. 4.1

[8] P. Burt and E. H. Adelson. The laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, 31(4):532–540, 1983. 4.1, 4.2.1

[9] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *IEEE CVPR*, volume 1, pages 886–893, 2005. 4.1, 4.5.1

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE CVPR*, pages 248–255, 2009. 4.1

[11] Jia Deng, Alexander C Berg, Kai Li, and Li Fei-Fei. What does classifying more than 10,000 image categories tell us? In *ECCV*, pages 71–84, 2010. 4.1

[12] Minh N Do and Martin Vetterli. Framing pyramids. *IEEE Signal Processing*, 51(9):2329–2342, 2003. 4.3

[13] Eizaburo Doi and Michael S Lewicki. Sparse coding of natural images using an overcomplete set of limited capacity units. *NIPS*, 17:377, 2004. 1

[14] N. R. Draper and H. Smith. *Applied Regression Analysis*. Hoboken, NJ: Wiley-Interscience,

1998. 4.5.2

[15] D Dueck and B Frey. Non-metric affinity propagation for unsupervised image categorization. *International Conference on Computer Vision*, 2007. 5.1

[16] K.A. Ehinger, J. Xiao, A. Torralba, and A. Oliva. Estimating scene typicality from human ratings and image features. In *Proceedings of the 33rd Annual Conference of the Cognitive Science Society*, 2011. 1

[17] Li. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *IEEE CVPR*, pages 524–531, 2005. 1, 4.5.1

[18] B Frey and D Dueck. Clustering by passing messages between data points. *Science*, 2007. 5.1

[19] J. C. Gilbert and J. Nocedal. Global convergence properties of conjugate gradient methods for optimization. *SIAM Journal on Optimization*, 2(1):21–42, 1992. 3.2.4

[20] M. R. Greene and A. Oliva. Recognition of natural scenes from global properties: Seeing the forest without representing the trees. *Cognitive Psychology*, 58(2):137 – 176, 2009. 1, 4.1, 4.5.2

[21] Abhinav Gupta, Martial Hebert, Takeo Kanade, and David M. Blei. Estimating spatial layout of rooms using volumetric reasoning about objects and surfaces. In J.D. Lafferty, C.K.I. Williams, J. Shawe-Taylor, R.S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 1288–1296. Curran Associates, Inc., 2010. 5.1

[22] Geoffrey Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006. 1

[23] Derek Hoiem and Silvio Savarese. *Representations and Techniques for 3D Object Recognition and Scene Interpretation*. Morgan & Claypool Publishers, 2011. 5.2.1

[24] Derek Hoiem, Alexei A Efros, and Martial Hebert. Putting objects in perspective. *International Journal of Computer Vision*, 80(1):3–15, 2008. 5.1

[25] A. Hyvrinen and E. Oja. Independent component analysis: algorithms and applications. *Neural Networks*, 13(4-5):411 – 430, 2000. ISSN 0893-6080. doi: DOI:10.1016/S0893-6080(00)00026-5. 1, 2.2.2

[26] Phillip Isola, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. What makes an image memorable? In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 145–152, 2011. 1

[27] Phillip Isola, Jianxiong Xiao, Devi Parikh, Antonio Torralba, and Aude Oliva. What makes a photograph memorable? 2013. 1

[28] C. Kanan and G. Cottrell. Robust classication of objects, faces, and flowers using natural image statistics. In *IEEE CVPR*, 2010. 4.1

[29] Y. Karklin and M. S. Lewicki. Emergence of complex cell properties by learning to generalize in natural scenes. *Nature*, 457:83–86, 2009. 3.1, 3.2.1, 3.2.4, 4.1, 5.6

[30] Koray Kavukcuoglu, Marc'Aurelio Ranzato, Rob Fergus, and Yann LeCun. Learning invariant features through topographic filter maps. In *Proc. International Conference on*

Computer Vision and Pattern Recognition (CVPR'09). IEEE, 2009. 1

[31] Hugo Larochelle and Yoshua Bengio. Classification using discriminative restricted boltzmann machines. In *Proceedings of the 25th international conference on Machine learning*, pages 536–543. ACM, 2008. 1

[32] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. *IEEE CVPR*, 2:2169 – 2178, 2006. 1, 3.2.5, 4.1, 4.5.1

[33] Q. V. Le, J. Ngiam, A. Coates, A. Lahiri, B. Prochnow, and A. Y. Ng. On optimization methods for deep learning. In *ICML*, 2011. 3.2.2

[34] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 5.1, 6

[35] Honglak Lee, Chaitanya Ekanadham, and Andrew Y. Ng. Sparse deep belief net model for visual area v2. In J.C. Platt, D. Koller, Y. Singer, and S.T. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 873–880. Curran Associates, Inc., 2008. 6

[36] Honglak Lee, Yan Largman, Peter Pham, and Andrew Y. Ng. Unsupervised feature learning for audio classification using convolutional deep belief networks. In *Advances in Neural Information Processing Systems 22*, pages 1096–1104. 2009. 1

[37] T. Lee and M.S. Lewicki. Unsupervised image classification, segmentation, and enhancement using ica mixture models. *IEEE Transactions on Image Processing*, 11(3):270 –279, 2002. 2.1

[38] T. Lee, M.S. Lewicki, and T. Sejnowski. Ica mixture models for unsupervised classification of non-gaussian classes and automatic context switching in blind signal separation. *IEEE Transitions on Pattern Analysis and Machine Intelligence*, 22(10):1078–1089, 2000. 2.1, 2.2.1

[39] W. Lee and M.S Lewicki. Adaptive representations of scenes based on ica mixture model. In *Big Vision Workshop at NIPS*, 2012. 1

[40] W. Lee and M.S Lewicki. Learning global properties of scene images based on their correlational structures. In *Deep Learning and Unsupervised Feature Learning Workshop at NIPS*, 2012. 1

[41] Yuanqing Lin, Fengjun Lv, Shenghuo Zhu, Ming Yang, Timothee Cour, Kai Yu, Liangliang Cao, and Thomas Huang. Large-scale image classification: fast feature extraction and svm training. In *IEEE CVPR*, pages 1689–1696, 2011. 4.1

[42] Dong Liu and Jorge Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical Programming*, 45(1):503–528, 1989. 3.2.4

[43] Tie Liu, Jian Sun, Nan-Ning Zheng, Xiaoou Tang, and Heung-Yeung Shum. Learning to detect a salient object. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007. 6

[44] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of

the spatial envelope. *International Journal of Computer Vision*, 42:145–175, 2001. 1, 4.1, 4.5.1, 4.5.2

[45] A. Oliva and A. Torralba. Building the gist of a scene: the role of global image features in recognition. volume 155 of *Progress in Brain Research*, pages 23 – 36. 2006. 1, 4.1

[46] A. Oliva, A. Torralba, M. Castelhano, and J. Henderson. Top-down control of visual attention in object detection. *In Proceedings of International Conference on Image Processing*, 1:253–256, 2003. 4.1

[47] Aude Oliva and Antonio Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International journal of computer vision*, 42(3):145–175, 2001. 5.6

[48] Aude Oliva and Antonio Torralba. Scene-centered description from spatial envelope properties. In *Biologically Motivated Computer Vision*, pages 263–272, 2002. 4.5.2

[49] B. A. Olshausen and D. J. Field. Natural image statistics and efficient coding. In *Network: Computation in Neural Systems, 7:333–339*, pages 333–339, 1996. 1, 4.1

[50] Bruno A Olshausen and David J Field. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision research*, 37(23):3311–3325, 1997. 6

[51] Soojin Park, Timothy F Brady, Michelle R Greene, and Aude Oliva. Disentangling scene content from spatial boundary: complementary roles for the parahippocampal place area and lateral occipital complex in representing real-world scenes. *The Journal of Neuroscience*, 31(4):1333–1340, 2011. 6

[52] Rajat Monga Matthieu Devin Kai Chen Greg S. Corrado Jeff Dean Quoc V. Le, MarcAurelio Ranzato and Andrew Y. Ng. Building high-level features using large scale unsupervised learning. *ICML*, 2011. 6

[53] Rajat Raina, Alexis Battle, Honglak Lee, Benjamin Packer, and Andrew Y Ng. Self-taught learning: transfer learning from unlabeled data. In *ICML*, pages 759–766, 2007. 4.1

[54] M Ranzato and G Hinton. Modeling pixel means and covariances using factorized third-order boltzmann machines. *Computer Vision and Pattern Recognition*, 2010. 5.1

[55] MarcAurelio Ranzato, Fu-Jie Huang, Y-Lan Boureau, and Yann LeCun. Unsupervised learning of invariant feature hierarchies with applications to object recognition. In *Proc. Computer Vision and Pattern Recognition Conference (CVPR'07)*. IEEE Press, 2007. 1

[56] M. Ross and A. Oliva. Estimating perception of scene layout properties from global image features. *Journal of Vision*, 10:1–25, 2010. (document), 1, 2.4.2, 3.2.5, 4.1, 4.1, 4.4, 4.4, 4.5.2

[57] S T Roweis and L K Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–6, 2000. 5.1

[58] P Sand and A Moore. Repairing faulty mixture models using density estimation. *International Conference on Machine Learning*, 2001. 5.1

[59] L Saul and S Roweis. Think globally, fit locally: unsupervised learning of low dimensional manifolds. *The Journal of Machine Learning Research*, 2003. 5.1

[60] Ashutosh Saxena, Sung H Chung, and Andrew Ng. Learning depth from single monocular images. *NIPS*, 18:1161, 2006. 4.1

[61] T. Serre, L. Wolf, and T. Poggio. Object recognition with features inspired by visual cortex. *IEEE CVPR*, 2:994–1000, 2005. 4.1

[62] Ling Shao and Michael Brady. Specific object retrieval based on salient regions. *Pattern Recognition*, 39(10):1932–1948, 2006. 6

[63] Patrice Simard, Bernard Victorri, Yann LeCun, and John S Denker. Tangent prop-a formalism for specifying selected invariances in an adaptive network. pages 895–903, 1992. 5.1

[64] Eero P Simoncelli and William T Freeman. The steerable pyramid: A flexible architecture for multi-scale derivative computation. In *International Conference on Image Processing*, volume 3, pages 444–447, 1995. 4.1

[65] Y Tang and A Mohamed. Multiresolution deep belief networks. *International Conference on Machine Learning*, 2012. 5.1

[66] J-P Tardif. Non-iterative approach for fast and accurate vanishing point detection. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1250–1257. IEEE, 2009. 5.1

[67] J Tenenbaum, V Silva, and J Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 2000. 5.1

[68] A. Torralba. How many pixels make an image? *Visual neuroscience*, 26:123–131, 2009. 2.1, 3.2.5, 6

[69] A Torralba. Contextual priming for object detection. *International Journal of Computer Vision*, 53(2):169–191, 2003. 6

[70] Antonio Torralba and Alyosha Efros. Unbiased look at dataset bias. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011. 1

[71] J. H. van Hateren and D. L. Ruderman. Independent component analysis of natural image sequences yields spatio-temporal filters similar to simple cells in primary visual cortex. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 265(1412): 2315–2320, 1998. 1, 4.1

[72] J.H. van Hateren and A. van der Schaaf. Independent component filters of natural images compared with simple cells in primary visual cortex. *Proceedings of the Royal Society of London Series B: Biological Sciences*, 265(1394):359–366, 1998. 2.3.1

[73] J Verbeek. Learning nonlinear image manifolds by global alignment of local linear models. *IEEE Transactions on Pattern Analysis and Machine Learning*, 2006. 5.1

[74] Lior Wolf and Stanley Bileschi. A critical view of context. *IJCV*, 69(2):251–261, 2006. 4.1

[75] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *IEEE CVPR*, pages 3485–3492, 2010. 2.1, 3.2.5, 3.5, 4.1, 4.3, 4.5.1, 5.3

[76] LeCun Yann and L Bottou. Large scale online learning. *NIPS*, 16:217, 2004. 3.2.2

[77] Matthew D Zeiler, Graham W Taylor, and Rob Fergus. Adaptive deconvolutional networks for mid and high level feature learning. In *IEEE ICCV*, pages 2018–2025, 2011. 4.1, 6

ML
MACHINE LEARNING
DEPARTMENT

School of Computer Science
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213
www.ml.cmu.edu