HIERARCHICAL DENSITY ESTIMATION FOR IMAGE CLASSIFICATION

BY

ZHEN LI

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2010

Urbana, Illinois

Adviser:

Professor Thomas S. Huang

ABSTRACT

Histogram (bag-of-words) and Gaussian mixture models (GMMs) have been widely used in patch-based image classification problems. Despite the satisfactory results reported, both methods suffer from a number of disadvantages. For instance, a histogram may be easy to learn but has a large quantization error; on the contrary, Gaussian mixture model based methods have better modeling capabilities but are inefficient in both learning and testing. In this thesis, we present a novel hierarchical density estimation approach for image classification. This new approach partitions the feature space into small regions using a tree structure. For each region, "local" distribution is characterized by class-conditional Gaussians via hierarchical *maximum a posteriori* (MAP) estimation. We further enhance the parameter estimation by smoothing over a collection of randomized trees. This new approach enjoys the merits of superior modeling capability, robust parameter estimation, and efficient testing. Experiments on scene classification demonstrate both the effectiveness and efficiency of this new approach.

To my parents, for their love and support

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my adviser, Prof. Thomas S. Huang, for his guidance and support during my years of study. I thank my parents for always believing in and nurturing my abilities. I thank all my friends, near and far, for their support and care, and being a source of inspiration.

Many people have contributed in various ways to make this journey more exciting and enjoyable. A special mention to my collaborators: Prof. Mark Hasegawa-Johnson, Dr. Xi Zhou, and Dr. Hao Tang. Without their help, this work could not have been accomplished. Finally, I am grateful to all former and current members of the Image Formation and Processing (IFP) laboratory, where I have spent my time in pursuing a master's degree.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 Background and Related Work

Many of the recent visual recognition tasks have been based on image patch features [1],[2],[3]. Images are divided into small patches, over which visual descriptors such as *scale-invariant feature transforms* (SIFTs) [1] and *local binary patterns* (LBPs) [4] are extracted. As an analogy to document classification, Fei-Fei and Perona [5] introduced the *bag-of-words* (BOW) method to model the patch distribution. In their approach, a histogram is formed for each image by quantizing patch features to one of the visual "words" in the learned "codebook." Recognition is then performed by the Bayesian decision theorem [5] or discriminant classifiers such as *support vector machine* (SVM) [6], *nearest neighbor* (NN) [7], or *AdaBoost* [8].

Despite its popularity, however, the conventional bag-of-words method has some intrinsic issues such as large quantization error and lack of discriminability. Several approaches have been proposed in the literature to overcome these limitations [9],[10],[11],[12],[13]. In particular, the *Gaussian mixture model* (GMM) emerged recently as an alternative to histograms due to its superior modeling capability. GMM is attractive for this purpose because it has well-studied training algorithms, it generalizes well to unseen data, and, more importantly, it can theoretically approximate arbitrary distribution with a sufficient number of mixtures. Methods based on GMM have achieved good performance in age estimation, object recognition, and video event analysis [14],[15],[16].

GMM based methods are not without limitations, though. First, GMM is usually trained for each class independently, while discriminative training, namely, joint training with data from all classes, was shown to yield better performance in many application scenarios [12],[9]. Second, in order to achieve good modeling capability, a large number of mixtures are needed in practice. This is however at

the risk of over-fitting especially with limited training data. Third, the classification (testing) process is slow as it requires evaluating feature vectors on hundreds or even thousands of Gaussians [17].

1.2 Hierarchical Distribution Estimation

In this thesis, we propose a hierarchical method to model the distribution of patch features. The hierarchical structure is supported by a tree [18] and further enhanced by a random forest [19]. In a discriminative manner, we partition the feature space into small regions using a tree structure. The "local" distribution of patch features within each region is then modeled by a *localized Gaussian*. Inspired by the work in speech recognition [20], we estimate the parameters of these localized Gaussians using a hierarchical *maximum a posteriori* (MAP) approach: the parameter estimation at a parent node works as the *prior* for that at its child nodes. Finally, a random forest is employed to obtain a smoothed estimation by aggregating the strength of randomized trees, while in the meantime, over-fitting is mostly eliminated.

The advantages of this proposed approach are listed as follows.

1. Discriminative training is achieved during tree construction.
2. The parameters of localized Gaussians are reliably estimated via both hierarchical MAP estimation and the aggregation of multiple randomized trees.
3. The hierarchical tree structure enables fast localization of a feature vector into a small region, in which case the density function only needs to be computed locally and thus efficiently.

Experiments on scene classification demonstrate that our approach achieves both good performance and high efficiency, especially when there are only a few training samples.

1.3 Thesis Outline

The remainder of this thesis is arranged as follows. In Chapter 2, we briefly describe the procedure of constructing a hierarchical structure by tree. Chapter 3

provides the details of the hierarchical density estimation method. In Chapter 4, we introduce the random forest approach to obtain smoothed parameter estimation from multiple trees. The experimental results on scene classification are reported in Chapter 5, and discussions and conclusions are given in Chapter 6.

CHAPTER 2

TREE CONSTRUCTION

This chapter illustrates the procedure of constructing the hierarchical structure of the feature space by growing a binary tree.

A tree (as shown in Figure 2.1) partitions the feature space into small regions, each of which corresponds to a leaf node in the tree. Data travels from the top (root) node to a terminal (leaf) node along a path determined by binary "questions" asked by the nodes it encounters.

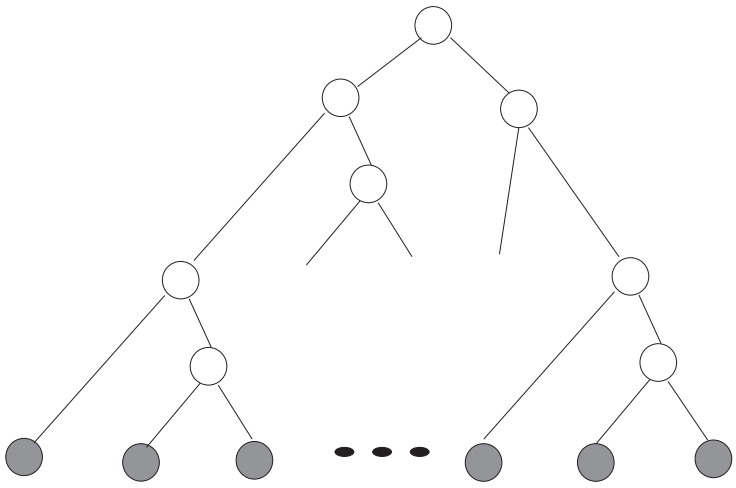

Figure 2.1: A tree.

At each node, as we deal with data from multiple classes, one straightforward criterion targeted at a later classification stage is to split the data into two parts such that each of them are as "pure" as possible. Given a properly defined *purity* or *impurity* measure, the objective is therefore to maximize the purity increase, or decease of impurity, namely,

$$\max_{\text{split of node } \ell} \Delta I(\ell) = I(\ell) - P_L I(\ell_L) - P_R I(\ell_R), \tag{2.1}$$

where ℓ_L and ℓ_R are the left and right child of node ℓ, and P_L and P_R are the

probabilities of data reaching these two nodes from their parent, respectively. One of the most popular impurity measure is the *entropy impurity*, namely,

$$I(\ell) = \sum_j p(c_j|\ell) \log p(c_j|\ell) \tag{2.2}$$

with $p(c_j|\ell)$ being the probability of j-th class at node ℓ.

Various types of "questions" can be asked about feature vectors. For example, one can choose one dimension of the feature vector and apply a threshold. This is equivalent to partitioning the feature space by a coordinate-orthogonal plane. More generally, thresholding on the output of arbitrary linear or non-linear functions of the feature vector can be explored. Apparently, the constraints on the "questions" that can be asked define the set of splits upon which the objective function in (2.1) is to be optimized.

There exists no theoretical solution for the splitting problem in (2.1), though. While a "better" split can generally be found in a large split set, a higher risk of over-fitting this split might suffer. Furthermore, as the optimization is only done locally and greedily, the "best" split at the current node does not necessarily guarantee that the successive locally optimized decisions lead to the global optimum. On the other hand, the pursuit of optimality of a single tree can be largely alleviated by the use of a random forest. Therefore, we can simply adopt a randomized single-dimensional thresholding strategy, i.e., to find a sub-optimal split by trying a set of thresholds on some random dimensions. The tree construction algorithm is given in Algorithm 1.

Algorithm 1 Tree Construction

Input: A set of data points X; minimum leaf node size N; maximum tree depth K;

Output: Tree T

Assign X as root
repeat
  Find a leaf node ℓ with $|\ell| \geq N$ and level $(\ell) \leq K$
  Randomly choose dimensions: $\{m\}$
  For each m, randomly choose thresholds: $\{\nu_k^m\}$
  for each threshold ν_k^m **do**
    for each $x \in \ell$ **do**
      if $x^m > \nu_k^m$ **then**
        Assign x to ℓ_L
      else
        Assign x to ℓ_R
      end if
    end for
    Compute $J_k^m = I(\ell) - P_L I(\ell_L) - P_R I(\ell_R)$
  end for
  Find the maximum J_k^m and split node ℓ w.r.t. m and ν_k^m
until $|\ell| < N$ or level $(\ell) > K, \forall \ell \in T$

CHAPTER 3

HIERARCHICAL DENSITY ESTIMATION

In this chapter, we present details of the proposed hierarchical density estimation method.

3.1 Localized Gaussians

Given a tree structure, one option for density estimation is to do it independently for each leaf node (local region). One common way is to compute a *probability mass function* (PMF) for node ℓ as $p(\ell|c_j) = \frac{\#(\ell,c_j)}{\#(c_j)}$, where $\#(c_j)$ is the total number of samples of the j-th class in the entire training set, while $\#(\ell, c_j)$ is the number of samples of the j-th class that reaches node ℓ.

The major drawbacks of the above PMF method are poor modeling capacity and large quantization error. Particularly, even for a tree of moderate depth, the local region specified by a leaf node could be substantially large (due to high dimensionality of the feature space). All samples within that region will be associated with the same PMF value, which is a rather coarse approximation to the true density function we are trying to estimate. In addition, a deep tree is very likely to contain leaf nodes with fewer training samples. Under this condition the counting based PMF can be easily over-trained.

Therefore, instead of counting the number of samples, we can assume a unimode Gaussian for each class at a leaf node. From the feature space point of view, this is equivalent to estimating a collection of Gaussians within small regions locally. Figure 3.1 illustrates localized Gaussians for a single class, associated with the corresponding tree structure. The parameters of these localized Gaussians can then be estimated by the *maximum likelihood* (ML) or *maximum a posteriori* (MAP) methods.

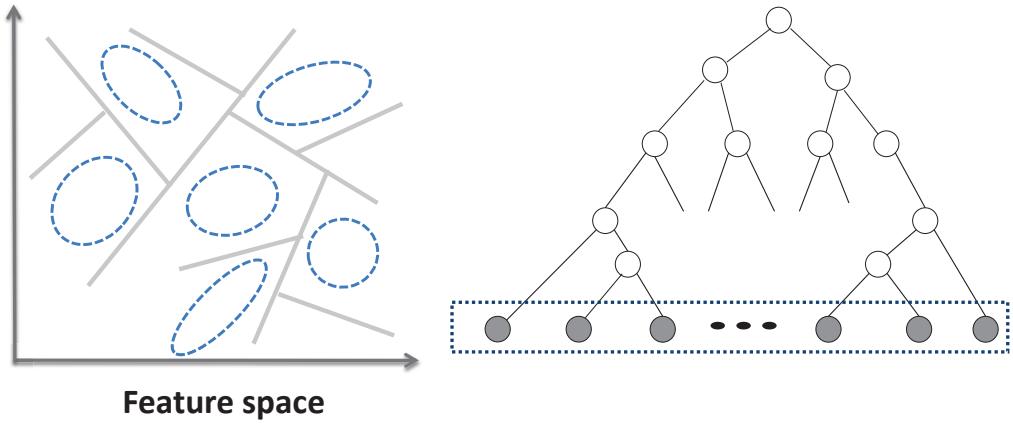

Figure 3.1: Localized Gaussians (for a single class).

3.2 Hierarchical MAP Estimation

The number of parameters of localized Gaussians increases exponentially as the tree goes deeper. Even if we assume diagonal covariance matrices of Gaussians instead of full ones, the total number of parameters is still as high as $2^{K+1} \times d \times C$, where K is the maximum level of the tree (root node at level 0), d is the dimension of the feature vector, and C the total number of classes.

The Gaussian parameters at leaf nodes can be well estimated if the number of training samples is sufficiently large. However, that is seldom true in practice, especially for deep trees. In that case estimating so many parameters is very likely to be unreliable and thus prone to over-fitting. The unstable estimation will greatly degrade the performance in the end.

A straightforward way to deal with this issue is to take advantage of the hierarchical structure of the tree. As can be expected, if we truncate the tree to some level, the density estimation at these new leaf nodes will become more reliable, whereas the modeling capability is sacrificed to some extent. It seems difficult to strike a balance between robust estimation and good modeling capability. However, the inherent hierarchical structure in the tree indicates the use of a *coarse-to-fine* strategy [20] to achieve both estimation robustness and modeling capability

The intuition here is that the final estimate at a leaf node could be some weighted combination of estimates by itself and from all its ancestor nodes. If few samples reach the leaf node, the estimate from its parent or grandparent should be

"trusted" to a larger extent, or in other words, should be given a higher weighting. When assuming hierarchical priors, the *maximum a posteriori* (MAP) estimation (Bayesian learning) yields desired effect.

3.2.1 Basic MAP estimation

The idea of MAP estimation is to leverage on some prior knowledge of parameters rather than fully relying on the training data. In this method, the parameters are regarded as random variables whose prior probability is assumed. Instead of maximizing the likelihood of training data given parameters, as in the *maximum likelihood* ML estimation, MAP estimator tries to maximize the joint probability of the parameters and the training data, i.e.,

$$\begin{aligned} \hat{\theta} &= \arg\max_{\theta} p(\theta|X) \\ &= \arg\max_{\theta} \frac{p(X|\theta)p(\theta)}{P(X)} \\ &= \arg\max_{\theta} p(X|\theta)p(\theta), \end{aligned} \tag{3.1}$$

where X is the training data and θ is the set of parameters to be estimated.

The MAP estimation provides a framework for incorporating prior information in the training process. This is particularly useful in dealing with problems posed by sparse training data when the ML approach fails to give an accurate estimation. From another point of view, MAP estimation can be regarded as a way of parameter smoothing. It is well-known that MAP estimation is asymptotically convergent to ML estimation [21]. As the amount of training data increases from zero to infinity, the MAP approach will generate an estimation that smoothly varies from completely prior to ML estimation.

3.2.2 Hierarchical priors

To solve the MAP objective function in (3.1), one is free to assign arbitrary density function for $p(\theta)$ if no specific knowledge about the parameter set θ is known. In the case of a hierarchical structure, however, it is intuitive to set the priors at one level to be the estimates from upper levels. Figure 3.2 illustrates a path of the structural way of imposing priors from top to bottom, where θ_k is the parameter

set at the k-th level. Note that, for simplicity, we drop the node ID and class dependency in the following derivations.

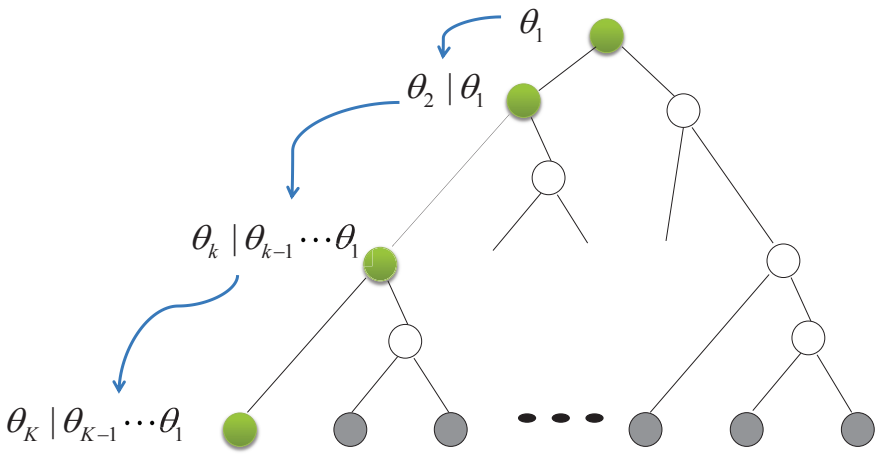

Figure 3.2: Hierarchical priors.

We further assume that an estimation obtained at a given node depends on prior information only from its immediate parent. The problem is then simplified to maximizing the MAP objective function recursively, starting from the root node. For a node at the k-th level, we obtain MAP estimation by

$$\hat{\theta}_k = \arg\max_{\theta_k} p(X_k|\theta_k)p(\theta_k|\hat{\theta}_{k-1}), \tag{3.2}$$

where X_k denotes the subset of data that reaches a given node at level k, and $p(\theta_k|\hat{\theta}_{k-1})$ is the prior distribution imposed by its parent at the $(k-1)$-th level. For the root node, no prior is assumed (or the prior is assumed to be uniform) and hence

$$\begin{aligned} \hat{\theta}_0 &= \arg\max_{\theta_0} p(X_0|\theta_0) \\ &= \arg\max_{\theta_0} p(X|\theta_0) \end{aligned} \tag{3.3}$$

degenerates to ML estimation.

3.2.3 MAP estimation based on hierarchical priors

Now we solve the objective function in (3.2). For a node at the k-th level, let the parameter set $\theta_k = \{\mu_k, \Sigma_k\}$ be the *mean* and *covariance* of data that reaches

a given node. To make the MAP estimation in (3.2) tractable, it is common to assume *conjugate prior* for random variable θ_k [21]. For a multivariate Gaussian distribution, the joint conjugate prior of mean and covariance is a *Normal-Inverse-Wishart* distribution:

$$\begin{aligned} p(\theta_k|\hat{\theta}_{k-1}) &= p(\mu_k, \Sigma_k|\hat{\mu}_{k-1}, \hat{\Sigma}_{k-1}) \\ &\propto |\hat{\Sigma}_{k-1}|^{-\frac{\xi_k}{2}} \exp\left\{-\frac{\tau_k}{2}(\mu_k - \hat{\mu}_{k-1})^T \Sigma_k^{-1}(\mu_k - \hat{\mu}_{k-1})\right\} \\ &\quad \cdot \exp\left\{-\frac{1}{2}\text{tr}(\hat{\Sigma}_{k-1}\Sigma_k^{-1})\right\} \end{aligned} \tag{3.4}$$

for $k = 1, \cdots, K$, with $\tau_k > 0$, and $\xi_k > -1$ being the two additional control parameters specified by external constraints.

Substituting (3.4) into (3.2) and differentiating with regard to μ_k and Σ_k, respectively, we arrive at the MAP solution:

$$\hat{\mu}_k = \frac{n_k\tilde{\mu}_k + \tau_k\hat{\mu}_{k-1}}{n_k + \tau_k} \tag{3.5}$$

$$\hat{\Sigma}_k = \frac{1}{n_k + \xi_k}\left[\hat{\Sigma}_{k-1} + n_k\tilde{\Sigma}_k + \frac{n_k\tau_k}{n_k + \tau_k}(\tilde{\mu}_k - \hat{\mu}_{k-1})(\tilde{\mu}_k - \hat{\mu}_{k-1})^T\right] \tag{3.6}$$

for $k = 1, \cdots, K$. In the above equations, n_k is the number of training samples at the given node, and $\tilde{\mu}_k$ and $\tilde{\Sigma}_k$ are the *empirical* mean and covariance given by

$$\tilde{\mu}_k = \frac{1}{n_k}\sum_{i=1}^{n_k} x_i \tag{3.7}$$

$$\tilde{\Sigma}_k = \frac{1}{n_k}\sum_{i=1}^{n_k}(x_i - \tilde{\mu}_k)(x_i - \tilde{\mu}_k)^T. \tag{3.8}$$

Note that this is essentially the ML estimation. As no prior distribution is assumed for the root node, the mean and covariance are simply obtained by (3.7) and (3.8).

According to (3.5), the hierarchical MAP estimation of mean parameter can be written as

$$\hat{\mu}_K = \sum_{k=0}^{K} c_k\hat{\mu}_k, \tag{3.9}$$

where the coefficient for the k-th level is

$$c_k = \frac{n_k}{n_k + \tau_k}\prod_{j=k+1}^{K}\frac{\tau_j}{n_j + \tau_j}. \tag{3.10}$$

This indicates that the final mean estimation at a leaf node is essentially a weighted sum of ML estimations of all its ancestor nodes. It is apparent from (3.10) that if the ancestor node is further away from leaf level (k small), its ML estimation contributes less to the final MAP estimation. On the other hand, the weighting increases as more samples reach that node. In an extreme case that the leaf node contains a sufficiently large amount of training data, the coefficient c_K approaches 1 and all other terms vanish. In other words, the MAP estimation degenerates to that given by ML, and prior information is automatically ignored.

It should be noted that the prior parameters τ_k and ξ_k need to be specified for each node. While MAP framework provides no specific ways to calculate these parameters, optimal values can be determined empirically. In particular, we use the same τ and ξ for all the nodes.

The hierarchial MAP estimation algorithm is given in Algorithm 2.

Algorithm 2 Hierarchial MAP Estimation

Input: A set of data points X; tree T

Pass X through T
Estimate for root node $\tilde{\mu}_0$ and $\tilde{\Sigma}_0$ based on (3.7) and (3.8)
Set $\hat{\mu}_0 = \tilde{\mu}_0$ and $\hat{\Sigma}_0 = \tilde{\Sigma}_0$
for each *path* $\wp \in T$ **do**
    Set $k = 1$
    repeat
        Find node $\ell \in \wp$ with level $(\ell) = k$
        Estimate $\tilde{\mu}_k$ and $\tilde{\Sigma}_k$ based on (3.7) and (3.8)
        Estimate $\hat{\mu}_k$ and $\hat{\Sigma}_k$ based on (3.5) and (3.6)
        $k \leftarrow k + 1$
    until leaf node of $\wp$ is reached
end for

CHAPTER 4

RANDOM FOREST

Although the means and covariances could be robustly estimated by the hierarchial approach discussed in Chapter 4, the localized Gaussians still suffer from a boundary problem. That is, the density function turns out to be unstable near boundaries between two or more regions defined by the leaf nodes. A little perturbation in the feature space might lead to a huge leap of probability density values. As we shall see in Chapter 5, this undesirable effect would yield unreliable estimation that decreases the performance.

Random forests (RFs) can be utilized to deal with this problem. A random forest is comprised of a collection of randomized trees [19]. It has been successfully used in many classification and regression tasks [22],[23]. In [22], Caruana et al. compared the performance of random forests with neural nets, boosted trees, and support vector machines. Random forests consistently achieved better performance than all the other classifier.s

In the forest, the probability density functions (PDFs) from different trees are averaged to generate the final density function, namely,

$$p(x) = \frac{1}{M} \sum_{m=1}^{M} p_m(x), \tag{4.1}$$

where $p_m(x)$ is the PDF given by the m-th tree, and M is the total number of trees. The value $p(x)$ is naturally smoothed. In the meantime the boundary problem can be largely alleviated by introducing randomness across trees. As has been pointed out in [19], the generalization error of a forest depends on the strength of the individual trees and the correlation (or diversity) between them. Ideally, the lowest error rate is achieved with uncorrelated and individually strongest trees.

In this work, the diversity among trees is achieved in three aspects.

1. Each tree is constructed based on a subset of all training samples.

2. At each split, a number of dimensions are chosen at random.

3. Some random thresholds are attempted at each dimension to find the optimal split.

CHAPTER 5

EXPERIMENTS

In this chapter, we evaluate the proposed approach on an image classification task. More specifically, we perform scene classification on the Fifteen Scene Database. Both performance and efficiency of our method are investigated and compared with existing approaches.

5.1 Dataset and Experimental Setups

The Fifteen Scene Database is one of the most comprehensive scene category databases used in the literature. It consists of fifteen scene categories, thirteen provided by Fei-Fei and Perona [5] and the other two collected by Lazebnik et al. [6]. Each scene class contains 200 to 400 images. The average size of the images is around 300 $\times$ 250 pixels. Example images of different scene categories of this database are shown in Figure 5.1.

In the experiments, a random subset with $N_{train} = \{1, 5, 20, 50, 100\}$ images per class is taken out to form the training set. For efficiency, only 50 images are drawn (without overlapping with the training set) for each class in testing. For each given N_{train}, we average the results over 10 individual runs.

5.2 Feature Extraction and Classifier

The 128-dimensional SIFT descriptors [1] are extracted densely for each image within 20 $\times$ 20 patches over a grid with spacing of 5 pixels. The dimensionality of the SIFT descriptors is reduced from 128 to 64 by principal component analysis

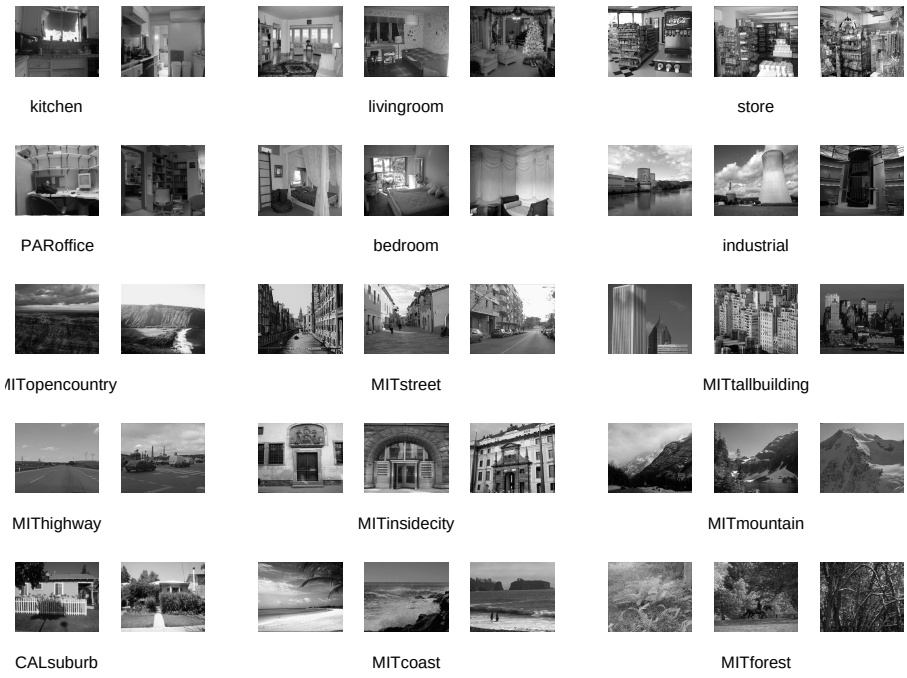

Figure 5.1: Example images from the scene category database.

(PCA). We perform classification based on the Bayesian decision theorem [21]:

$$\begin{aligned} \hat{c}_j &= \arg\max_{c_j} p(c_j|X) \\ &= \arg\max_{c_j} p(X|c_j)P(c_j), \end{aligned} \tag{5.1}$$

where c_j denotes the j-th class. In many cases including here, different classes are assumed to be equally likely, namely, the prior term $p(c_j)$ can be dropped from (5.1). Therefore, the Bayesian decision theorem simply assigns the class label that maximizes the likelihood function $p(X|c_j)$.

5.3 Results

Figure 5.2 presents a performance comparison for a scene recognition task. Here the forest is comprised of 10 trees and each of them is grown to have at most 9 levels (512 leaf nodes in theory). A histograms and localized Gaussians are estimated on the leaf nodes of each tree. For fair comparison, a 512-mixture GMM is trained under the same settings.

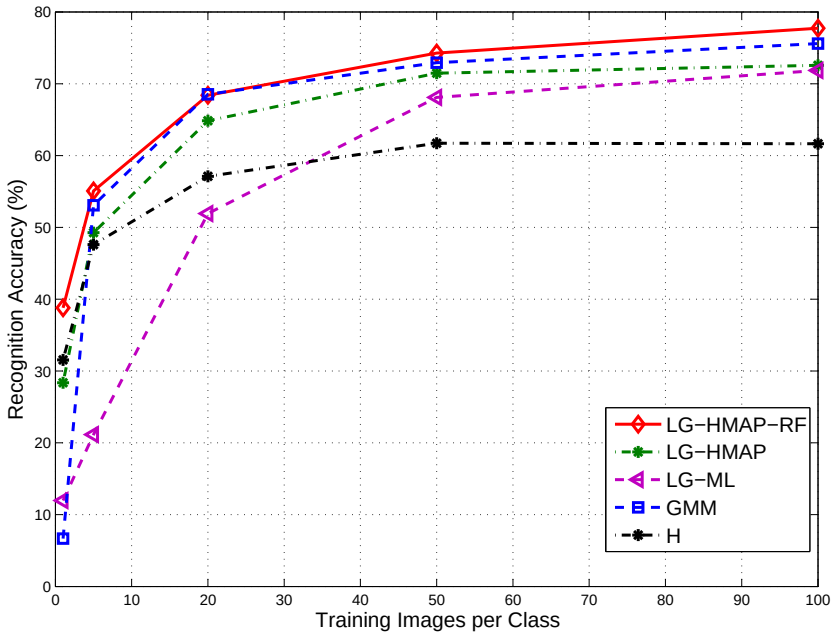

Figure 5.2: Performance on scene recognition. H: histogram; GMM: Gaussian mixture model; LG-ML: localized Gaussians by ML; LG-HMAP: localized Gaussians by hierarchical MAP; LG-HMAP-RF: localized Gaussians by hierarchical MAP in random forest.

It is apparent that RF-based hierarchical MAP estimation achieves the best results under all training conditions. Especially in the extreme case of only one training image per class, our method is still able to give an accuracy of about 39% (by chance is only about 7%). Under this condition, GMM performs significantly worse. It should be noted that, although GMM gives a comparable result with a large training set, the classification process of the proposed approach is much faster than that of GMM. For each tree, the likelihood of a feature vector is calculated against a single Gaussian. The searching process of this particular Gaussian only involves a few comparison operations, and is completed almost instantaneously (compared to the subsequent much slower Gaussian evaluation process). Therefore, the total number of Gaussian calculations in the proposed approach is equivalent to the number of trees in the forest (e.g. 10), which is more than an order of magnitude less than the number of mixtures in GMM (e.g. 512). We also observe that hierarchical MAP estimation consistently improves the performance of localized Gaussians, especially when there are only a few training samples. As

can be expected, the performance of ML and hierarchical MAP gradually merge as the size of the training set gets sufficiently large.

We also investigate the hierarchial weighting given by (3.10), as shown in Figure 5.3. When the training set contains only one image per class, the ML estimation obtained at leaf nodes can hardly be relied on, and information from upper levels is borrowed to constitute the final estimation. As more training images are added, the weighting at leaf nodes consistently increases, whereas that at upper levels turns less significant. Finally, when the training set is sufficiently large (100 images per class), the average weighting at leaf nodes approaches 1, and MAP estimation degenerates to ML. This is consistent with the results of LG-HMAP and LG with 100 training images in Figure 5.2.

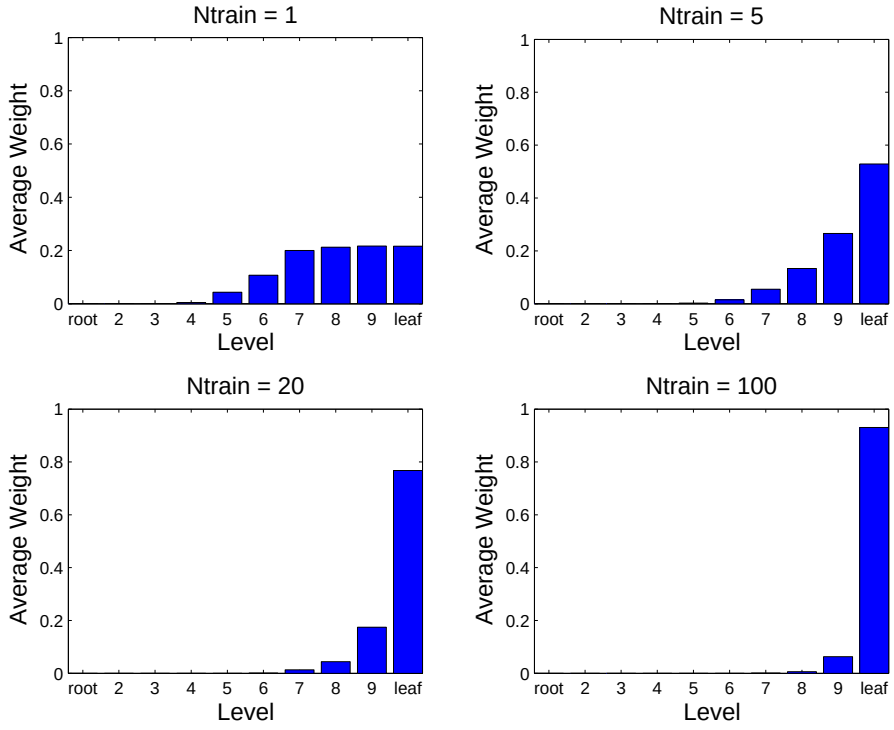

Figure 5.3: Hierarchial weights for each level in the tree.

Finally, we compare the proposed approach with several recently reported algorithms, as shown in Table 5.1. All results are reported with 100 training images per class. It should be noted that our approach outperforms the SVM based method in [6], even though the authors introduced an enhanced kernel metric to boost the performance of SVM.

Table 5.1: Performance comparison with previous methods.

Algorithm	Average accuracy (%)
Histogram-BH [5]	65.2
Histogram-SVM [6]	74.8
LG-HMAP-RF	77.7

CHAPTER 6

CONCLUSION

In the final chapter of thesis, we briefly summarize our contributions and discuss the potential directions for future work.

The motivation of the study is to seek a distribution estimation method for patch-based image classification that 1) has good modeling capability, 2) can be reliably learned, 3) is very efficient in testing. The proposed random forest based hierarchical density estimation approach meets all the aforementioned goals. First, the use of localized Gaussians inherently guarantees a superior modeling capability. Second, by using the hierarchical *maximum a posteriori* (MAP) method, the class-conditional feature distributions are robustly estimated in a tree structure. Third, random forest provides a way of incorporating the strength of multiple trees: the final estimation by forest is naturally smoothed and over-fitting can be avoided. Fourth, the efficiency of the testing process is achieved by confining the testing sample to several particular Gaussians. More specifically, the computational cost is linear in the number of trees (e.g. 10) rather than the number of Gaussians (e.g. 512).

Future work might include employing discriminant classifiers such as *support vector machine* (SVM) to further improve the performance of the proposed approach. As indicated from [6], the classification accuracy of the SVM based method could be significantly increased by choosing a carefully designed distance metric. We shall also consider incorporating the spatial correspondence among patches (by modeling the joint distribution of patches from different locations). Although it is disregarded in our current framework, the spatial information proved to be helpful in many visual recognition problems [24],[8].

REFERENCES

[1] D. Lowe, "Object recognition from local scale-invariant features," in *The 7th IEEE International Conference on Computer Vision*, vol. 2, 1999, pp. 1150–1157.

[2] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1615–1630, 2005.

[3] A. Bosch, A. Zisserman, and X. Munoz, "Representing shape with a spatial pyramid kernel," in *The 6th ACM International Conference on Image and Video Retrieval*, 2007, pp. 401–408.

[4] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, 2002.

[5] L. Fei-Fei and P. Perona, "A Bayesian hierarchical model for learning natural scene categories," in *The 18th IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, 2005, pp. 524–531.

[6] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *The 19th IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, 2006, pp. 2169–2178.

[7] H. Zhang, A. Berg, M. Maire, and J. Malik, "SVM-KNN: Discriminative nearest neighbor classification for visual category recognition," in *The 19th IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, 2006, pp. 2126–2136.

[8] J. Shotton, J. Winn, C. Rother, and A. Criminisi, "Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context," *International Journal of Computer Vision*, vol. 81, no. 1, pp. 2–23, 2009.

[9] F. Moosmann, B. Triggs, and F. Jurie, "Fast discriminative visual codebooks using randomized clustering forests," in *Advances in Neural Information Processing Systems*. Cambridge, MA: MIT Press, 2007, pp. 985–992.

[10] F. Perronnin, C. Dance, G. Csurka, and M. Bressan, "Adapted vocabularies for generic visual categorization," in *The 9th European Conference on Computer Vision*, 2006, pp. 464–475.

[11] J. van Gemert, J. Geusebroek, C. Veenman, and A. Smeulders, "Kernel codebooks for scene categorization," in *The 10th European Conference on Computer Vision*, 2008, pp. 696–709.

[12] L. Yang, R. Jin, R. Sukthankar, and F. Jurie, "Unifying discriminative visual codebook generation with classifier training for object category recognition," in *The 21st IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.

[13] T. Tuytelaars and C. Schmid, "Vector quantizing feature space with a regular lattice," in *The 11th IEEE International Conference on Computer Vision*, 2007, pp. 1–8.

[14] Y. Liu and F. Perronnin, "A similarity measure between unordered vector sets with application to image categorization," in *The 21st IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.

[15] S. Yan, X. Zhou, M. Hasegawa-Johnson, and T. Huang, "Regression from patch-kernel," in *The 21st IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.

[16] X. Zhou, X. Zhuang, S. Yan, S. Chang, M. Hasegawa-Johnson, and T. Huang, "Sift-bag kernel for video event analysis," in *The 16th ACM International Conference on Multimedia*, 2008, pp. 229–238.

[17] X. Zhou, X. Zhuang, H. Tang, M. Hasegawa-Johnson, and T. Huang, "A novel Gaussianized vector representation for natural scene categorization," in *The 19th International Conference on Pattern Recognition*, 2008, pp. 1–4.

[18] L. Breiman, *Classification and Regression Trees*. Chapman & Hall/CRC, 1984.

[19] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[20] O. Siohan, T. Myrvoll, and C. Lee, "Structural maximum a posteriori linear regression for fast HMM adaptation," *Computer Speech and Language*, vol. 16, no. 1, pp. 5–24, 2002.

[21] R. Duda, P. Hart, and D. Stork, *Pattern Classification*, 2nd ed. New York, NY: John Wiley and Sons, Inc., 2001.

[22] R. Caruana, N. Karampatziakis, and A. Yessenalina, "An empirical evaluation of supervised learning in high dimensions," in *The 21st International Conference on Machine Learning*, 2008, pp. 96–103.

[23] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Machine Learning*, vol. 63, no. 1, pp. 3–42, 2006.

[24] G. Qi, X. Hua, Y. Rui, J. Tang, Z. Zha, and H. Zhang, "A joint appearance-spatial distance for kernel-based image categorization," in *The 21st IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.