**Noname manuscript No.**
(will be inserted by the editor)

# Gaussian Processes for Object Categorization

**Ashish Kapoor · Kristen Grauman · Raquel Urtasun · Trevor Darrell**

Received: date / Accepted: date

**Abstract** Discriminative methods for visual object category recognition are typically non-probabilistic, predicting class labels but not directly providing an estimate of uncertainty. Gaussian Processes (GPs) provide a framework for deriving regression techniques with explicit uncertainty models; we show here how Gaussian Processes with covariance functions defined based on a Pyramid Match Kernel (PMK) can be used for probabilistic object category recognition. Our probabilistic formulation provides a principled way to learn hyperparameters, which we utilize to learn an optimal combination of multiple covariance functions. It also offers confidence estimates at test points, and naturally allows for an active learning paradigm in which points are optimally selected for interactive labeling. We show that with an appropriate combination of kernels a significant boost in classification performance is possible. Further, our experiments indicate the utility of active learning with probabilistic predictive models, especially when the amount of training data labels that may be sought for a category is ultimately very small.

**Keywords** Object Recognition · Gaussian Process · Kernel Combination · Active Learning

Ashish Kapoor
Microsoft Research, Redmond WA 98052, USA
E-mail: akapoor@microsoft.com

Kristen Grauman
University of Texas at Austin, TX 78712, USA
E-mail: grauman@cs.utexas.edu

Raquel Urtasun
UC Berkeley EECS & ICSI, Berkeley, CA 94720, USA
E-mail: rurtasun@csail.mit.edu

Trevor Darrell
UC Berkeley EECS & ICSI, Berkeley, CA 94720, USA
E-mail: trevor@eecs.berkeley.edu

## 1 Introduction

Object categorization is a fundamental problem in image understanding. It remains a challenging learning task given both the variability of images that objects from the same class can produce, as well as the substantial expense of providing high quality image annotations needed to train accurate models. Discriminative methods for visual category learning have yielded promising results in recent years, including various approaches based on support vector machines or nearest neighbor classification [14,49,46,30,21,43,5,13,19]. However, such methods typically are not explicitly probabilistic, which makes them inadequate when estimates of uncertainty are required. At the same time, probabilistic generative methods that attempt to directly model the joint distribution of object classes and their features—though appealing for their ability to estimate uncertainty during inference—can be impractical for image recognition applications due to the complexity of representing the data's underlying density.

In this work we provide a probabilistic discriminative approach to object categorization, with the goal of exercising the advantages of both types of methods. We introduce a new Gaussian Process (GP) regression method for object category recognition using a local feature correspondence kernel. Local feature-based object recognition has several important advantages, including invariance to various translational, rotational, affine and photometric transformations and robustness to partial occlusions. Our method is based on a GP with a covariance function derived from a Pyramid Match Kernel [14], which offers an efficient approximation to a partial-match distance function and can therefore handle outliers and occlusions. Our model offers some of the known benefits of probabilistic techniques, while still maintaining the power of a discriminative learner. In particular, we show how it enables both *active* visual category learning, as well as learning from multiple image feature sources with an optimal combination of covariance functions.

Collecting training data for large-scale image category models is a potentially expensive process. While certain categories may have a large number of training images available, many more will have relatively few. A number of ingenious schemes have been developed to obtain labeled data from people performing other tasks (e.g., [45,44]), or directly labeling objects in images [1]. To make the most of scarce human labeling resources it is imperative to carefully select points for user labeling. The paradigm of active learning has been introduced in the machine learning community to address this issue [12,39,25,29,50]; with an active learning method, generally new test points are selected so as to minimize the model entropy.

GPs have received limited attention in the computer vision literature to date perhaps due to the fact that they are conventionally limited to modest amounts of training data: the learning complexity is $O(n^3)$, cubic in the number of training examples. While recent advances in sparse GPs are promising (e.g., [20,34,37,40]), we focus here on the case of active learning with relatively small numbers of labeled examples (10-100), which is feasible with existing implementations. In this realm, we show that active learning provides significantly more accurate estimates per labeled point than does a conventional random selection of training points.

Specific choices made regarding image representations and kernel parameters can greatly influence a classifier's potential. Even within the domain of local image features and matching kernels, a variety of alternative interest point detectors, descriptors, match criteria, and feature space quantization strategies are available. Rather than require a user to decide *a priori* which particular items will define the GP's covariance function, we show how to automatically optimize the combination of kernels for the recognition task using the GP marginal likelihood function. As a result, one can compute a set of potential kernels using a variety of local feature types, and then directly learn a weight for each such that the final combination is highly discriminative. While recent work has considered multiple kernel learning [43,19] and cross-validation approaches [5] to combine image feature types within SVM classifiers, to our knowledge our approach is the first to consider kernel combinations in a probabilistic setting.

The three main contributions of this paper are (1) a probabilistic discriminative category recognition scheme based on a Gaussian Process prior with a covariance function defined using the Pyramid Match Kernel, (2) the introduction of an active learning paradigm for object category learning which optimally selects unlabeled test points for interactive labeling, and (3) a probabilistic approach to learn discriminative kernel combinations for multiple local feature types within a GP framework. We show that with active learning small amounts of interactively labeled data can provide very accurate category recognition performance, while with covariance functions that optimally combine multiple matching kernels our method obtains state-of-the-art results with benchmark datasets.

## 2 Previous Work

Object category recognition has been a topic of active interest in the computer vision literature. Methods based on local feature descriptors (c.f. [23,26]) have been shown to offer invariance across a range of geometric and photometric conditions. Early models captured appearance and shape variation in a generative probabilistic framework [11], but more recent techniques have typically exploited methods based on SVMs or Nearest Neighbors in a bag-of-visual-words feature space [36,30,49,9].

Several authors have explored correspondence-based kernels [49,46], where the distance between a set of local feature descriptors—potentially including appearance and shape / position—is computed based on associating pairs of descriptors. However, the polynomial-time computational cost of correspondence-based distance measures makes them unsuitable for domains where there are large databases or large numbers of features per image. In [14] the authors introduced the Pyramid Match Kernel (PMK), an efficient linear-time approximation to a partial match correspondence, and in [21] it was demonstrated that a spatial variant—which efficiently represents the distinction between appearance and image location features—outperformed many competing m-ethods.

Semi-supervised or unsupervised visual category learning methods are related to active learning, in that they also leverage unlabeled examples to learn more accurately when limited labeled examples are available. Generative models which model visual words as arising from a set of underlying objects or "topics" based on recently introduced methods for Latent Dirichlet Allocation have been developed [35,38] but as yet have not been applied to active learning nor evaluated on purely supervised tasks. A semi-supervised method using normalized cuts to cluster a graph defined by Pyramid Match distances between examples was presented in [16], but this method is not probabilistic nor does it provide for an active learning formalism.

In the machine learning literature active learning has been a topic of recent interest, and numerous schemes have been proposed for choosing unlabeled points for tagging. For example, in [12] the authors propose using the disagreement among the committee of classifiers as a criterion for active learning, and show an application to image classification [2]. In [39], unlabeled examples to query are selected based on minimizing the version space within the SVM formulation, while in [7] an SVM-based active learner is applied for image retrieval using color and texture features.

Within the Gaussian Process framework, the method of choice has been to look at the expected informativeness of an unlabeled data point [20,24]. Specifically, the idea is to choose to query cases that are expected to maximally influence the posterior distribution over the set of possible classifiers. Additional studies have sought to combine active learning with semi-supervised learning [25,29,50]. Our work is significantly different as we focus on local feature approaches for the task of object categorization. We explore the GP models, which provide estimates for uncertainty in prediction and can be easily extended to active learning.

Recent work has shown the value of combining multiple local image feature types into a single kernel matrix, either by using cross-validation with a held-out set of labeled images to adjust the weight attached to each [5], or by optimizing the weights to align the combined kernel with the ideal kernel matrix reflecting the labels on the training data [43, 19]. Both tactics have yielded impressive results in practice. Our proposed method to optimize kernel weights fits directly within our GP learning framework, and is distinct in that rather than target the labels of training examples, it maximizes the evidence of the probabilistic model.

Gaussian Processes have been recently introduced to the computer vision literature. While they have been used in [41, 42] for human motion modeling and in [48] for stereo segmentation, we are unaware of any prior work on visual object recognition in a Gaussian Process framework.[1]

## 3 Approach Overview

The main idea of our approach is to construct probabilistic discriminative classifiers for object recognition using Gaussian Process priors, with covariance functions defined by the Pyramid Match Kernel (GP-PMK). In addition to offering a novel approach to supervised visual category learning, we show how this framework also allows both the learning of optimal combinations of covariance functions, as well as an active learning strategy—which is especially preferable when minimal labeling effort is available. Figure 1 shows the proposed framework for active image categorization. Given a pool of images of which few are labeled, the system aims to actively seek labels for unlabeled images by considering information from both the labeled and unlabeled sets of images. With the uncertainty estimates the GP classifier provides, we are able to designate an active learning criterion to focus labeling efforts on the most ambiguous unlabeled examples.

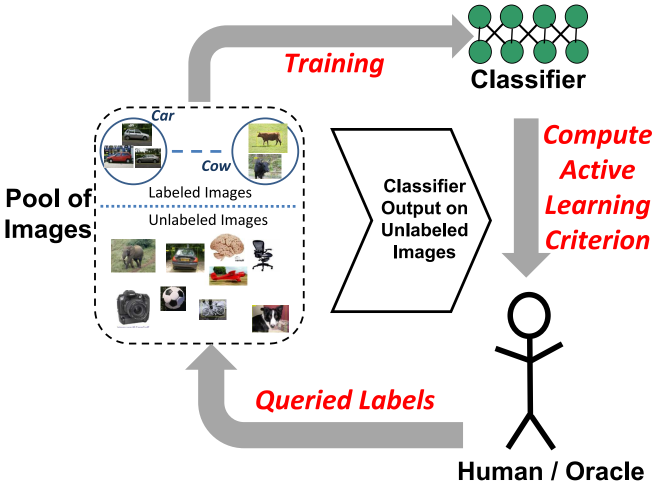

**Fig. 1** The active learning framework. The goal of the system is to query labels for images that are most useful in training.

In the next section we review classification using GP priors and discuss the distributions and parameters we employ for our model. Then in Section 5 we present our GP-PMK model, which is directly suitable for supervised learning with or without active learning. Then in Section 6 we describe how to optimize the weights to combine multiple matching kernels computed from different feature sets. Finally, we derive an active learning variant that can optimally select points for interactive labeling in Section 7.

Note that throughout we assume that there is one primary object of interest in an image. Handling multiple objects in the same image is also an interesting and challenging problem, and will be the focus of future work.

## 4 Categorization with Gaussian Processes

Gaussian Process (GP) classification is related to kernel machines such as Support Vector Machines (SVMs) [8] and Regularized Least Square Classification (RLSC) and has been well-explored in machine learning. In contrast to these methods, GPs provide probabilistic prediction estimates and thus are well- suited for active learning. In this section we briefly review regression and classification with Gaussian Process priors and describe our model choices.

Given a set of labeled data points $\mathbf{X}_L = \{\mathbf{x}_1, .., \mathbf{x}_n\}$, with class labels $\mathbf{t}_L = \{t_1, .., t_n\}$, we are interested in classifying the unlabeled data $\mathbf{x}_u$. Under the Bayesian paradigm, we are interested in the distribution $p(t_u|\mathbf{X}, \mathbf{t}_L)$. Here $\mathbf{X} = \{\mathbf{X}_L, \mathbf{x}_u\}$, and $t_u$ is the random variable denoting the class label for the unlabeled point $\mathbf{x}_u$. For sake of simplicity in discussion we limit ourselves to two-way classification, hence, the labels are $t_i \in \{-1, 1\}$, but this can be extended to multi-label classification; see [32] for a detailed discussion.

[1] This paper expands on our previous conference publication [18]; here we provide further explanation of our Gaussian Process model, extend it to allow combinations of multiple kernel functions, and report and discuss a number of additional experiments.

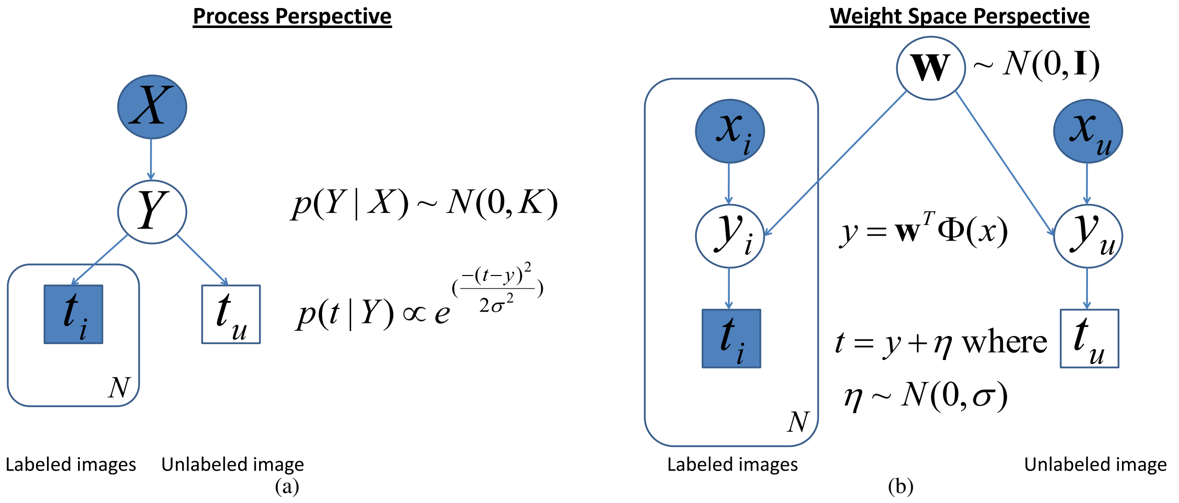

**Fig. 2** Graphical models in plate notation for classification via Gaussian Processes. The rounds and squares represent continuous and discrete random variables, respectively. A filled (unfilled) round/square denotes that the random variable is fully observed (unobserved). $\mathbf{X} = \{\mathbf{x}_1, .., \mathbf{x}_n, \mathbf{x}_u\}$ is the set of all images and is observed for both labeled and unlabeled data points. The corresponding $\mathbf{Y} = \{y_1, ..y_n, y_u\}$ is completely unobserved and the labels $\{t_1, .., t_n\}$ are observed only for the training images $\{\mathbf{x}_i, ..\mathbf{x}_n\}$ and unobserved for the unlabeled image $\mathbf{x}_u$.

With GP models, a discrete label $t$ for a data point $\mathbf{x}$ can be considered to be generated via a continuous hidden random variable $y$. The soft-hidden label arises due to a Gaussian Process, which in turn imposes a smoothness constraint on the possible solutions. A likelihood model $p(t|y)$ characterizes the relationship between the soft label $y$ and the observed annotation $t$. Thus, when we infer the label $t_u$ for the unlabeled data point $\mathbf{x}_u$, we probabilistically combine the smoothness constraint and the information obtained by observing the annotations $\mathbf{t}_L$.

## 4.1 Smoothness Constraints via the GP Prior

There exist two different perspectives for regression and classification with Gaussian Process: the process perspective and the weight perspective. We overview both in the following in order to provide background on the basic concepts underlying the GP model.

*The process perspective:* The smoothness constraint is imposed using a Gaussian Process prior that defines the probabilistic relationship between the images $\mathbf{X}$ and the soft labels $\mathbf{Y}$. The distribution $p(\mathbf{Y}|\mathbf{X})$ gives higher probability to the labelings that respect the similarity between the data points. Intuitively, the assumption is that similar data points should have the same class assignments / regression values; the similarity between two points $\mathbf{x}_i$ and $\mathbf{x}_j$ is defined via a kernel $k(\mathbf{x}_i, \mathbf{x}_j)$. Probabilistic constraints are imposed on the collection of soft labels $\mathbf{Y} = \{y_1, ..., y_n, y_u\}$. In particular, the soft labels are assumed to be jointly Gaussian and the covariance between two outputs $y_i$ and $y_j$ is typically specified using a kernel function[2] applied to $\mathbf{x}_i$ and $\mathbf{x}_j$. Formally, $p(\mathbf{Y}|\mathbf{X}) \sim \mathcal{N}(0, \mathbf{K})$ where $\mathbf{K}$ is a $(n+1)$-by-$(n+1)$ kernel matrix with $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$, and $n + 1$ reflects the $n$ labeled examples and one unlabeled example.

*The weight perspective:* What we have described above is the *process* perspective for regression and classification with the GP priors. An alternate but mathematically equivalent interpretation is based on the *weight* perspective. In this perspective the hidden soft-label $y$ arises due to application of a function $f(\cdot)$ directly on the input data point (i.e. $y = f(\mathbf{x})$), which takes the form of a linear combination of orthonormal basis functions:

$$f(\mathbf{x}) = \sum_k w_k \nu_k^{1/2} \phi_k(\mathbf{x}) = \mathbf{w}^T \Phi(\mathbf{x}), \tag{1}$$

where $\phi_k$ are the eigenfunctions of the operator induced by $k$ in the Reproducing Kernel Hilbert Space [8], $\nu_k$ are the corresponding eigenvalues, $w_k$ are the weights, and $\Phi(\mathbf{x}) = [\nu_1^{1/2}\phi_1(\mathbf{x}), \nu_2^{1/2}\phi_2(\mathbf{x}), ..]^T$. Note that the dimensionality of the basis can be infinite. Assuming a spherical Gaussian prior over the weights, that is $\mathbf{w} = [w_1, w_2, ..]^T \sim \mathcal{N}(0, \mathbf{I})$, it can be shown that the hidden soft labels $\mathbf{Y}$ (which result from evaluation of the function $f(\cdot)$ on the input data points $\mathbf{X}$) are jointly Gaussian with zero mean and with the covariance given by the kernel matrix $\mathbf{K}$.

These two different but equivalent perspectives for regression and classification with the GP priors are illustrated in Figure 2. Both views lead to different implementations,

[2] One can use a non-parametric covariance function, but the number of parameters to estimate grows exponentially with the amount of training data.

but are conceptually equivalent. In this work, we follow the process perspective. For details, please see [33].

### 4.2 The Likelihood Model

The likelihood models the probabilistic relationship between the observed label $t$ and the hidden label $y$. The majority of the likelihood models proposed for GP classification use additional latent "squashing" variables that transform unconstrained variables into labels. A wide range of squashing functions have been developed in the literature [32] and examples include the logisitic and the probit functions. To make predictions based on the training set for a test set in these models (i.e., probit and logit) one has to integrate out the prediction over the posterior. Since the likelihood is not Gaussian, neither the posterior, the marginal likelihood, nor the predictions can be computed analytically. Instead, one has to rely on numerical methods, such as MCMC [47], or approximations of the posterior, e.g. Laplace and Expectation Propagation [28].

In contrast to GP classification, GP regression leads to efficient analytic solutions for prediction. For Gaussian Process regression using a Gaussian noise model, the relation between $t$ and $y$ is given by

$$p(t|y) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(t-y)^2}{2\sigma^2}}, \tag{2}$$

where $\sigma$ is the noise model variance. Since this likelihood model is Gaussian, it leads to a closed form solution for inference. Although originally developed for regression, the Gaussian noise model has also proven effective for classification,[3] and its performance typically matches the more complex probit and logit likelihood models noted above. Due to its simplicity and good performance, in our experiments we use regression (i.e., the Gaussian noise model) to label variables. Non-Gaussian noise models could also be applied within the proposed framework, and exploring them is a topic of interest for future work.

### 4.3 Inference

Given the labeled and unlabeled data points, our goal is then to infer $p(t_u|\mathbf{X}, \mathbf{t}_L)$. Specifically:

$$p(t_u|\mathbf{X}, \mathbf{t}_L) \propto \int_{\mathbf{Y}} p(t_u|\mathbf{Y}) p(\mathbf{Y}|\mathbf{X}, \mathbf{t}_L). \tag{3}$$

For a Gaussian noise model we can compute this integral using closed form expressions. Note that the key quantity to compute is the posterior $p(\mathbf{Y}|\mathbf{X}, \mathbf{t}_L)$, which can be written as:

$$p(\mathbf{Y}|\mathbf{X}, \mathbf{t}_L) \propto p(\mathbf{Y}|\mathbf{X}) p(\mathbf{t}_L|\mathbf{Y}) = p(\mathbf{Y}|\mathbf{X}) \prod_{i=1}^{n} p(t_i|y_i). \tag{4}$$

This equation probabilistically combines the smoothness constraints $p(\mathbf{Y}|\mathbf{X})$ imposed via the GP prior and the information provided in the labels $(p(\mathbf{t}_L|\mathbf{Y}))$. The posterior as shown in Equation 4 is simply a product of Gaussians, and the posterior over the soft label $y_u$ has a particularly simple form. Specifically, $p(y_u|\mathbf{X}, \mathbf{t}_L) \sim \mathcal{N}(\bar{y}_u, \sigma_u^2)$, where:

$$\bar{y}_u = \mathbf{k}(\mathbf{x}_u)^T (\sigma^2 \mathbf{I} + \mathbf{K}_{LL})^{-1} \mathbf{t}_L \tag{5}$$

$$\Sigma_u = k(\mathbf{x}_u, \mathbf{x}_u) - \mathbf{k}(\mathbf{x}_u)^T (\sigma^2 \mathbf{I} + \mathbf{K}_{LL})^{-1} \mathbf{k}(\mathbf{x}_u). \tag{6}$$

Here, $\mathbf{k}(\mathbf{x}_u)$ is the vector of kernel function evaluations with $n$ training points, and $\mathbf{K}_{LL} = \{k(\mathbf{x}_i, \mathbf{x}_j)\}$, is the training covariance, with $\mathbf{x}_i, \mathbf{x}_j \in \mathbf{X}_u$. Further, due to the Gaussian noise model that links $t_u$ to $y_u$, the predictive distribution over the unknown label $t_u$ is also a Gaussian: $p(t_u|\mathbf{X}, \mathbf{t}_L) \sim \mathcal{N}(\bar{y}_u, \Sigma_u + \sigma^2)$.

Note that the posterior mean for both $t_u$ and $y_u$ is the same; thus, the unlabeled point $\mathbf{x}_u$ can be classified according to the sign of $y_u$. Unlike the Regularized Least Square Classification (RLSC) methods, we also get the variance in prediction. As we will show in Section 7, we can exploit these measures of uncertainty to guide an active learning procedure. The computationally costly operation in GP inference is the inversion of $(\sigma^2 \mathbf{I} + \mathbf{K}_{LL})$ which has a time complexity of $O(n^3)$ for $n$ training examples. In addition to reducing manual labeling effort, an active learning formulation can help us reduce the computational overhead in inference by reducing the number of needed training points.

### 4.4 Training with the Gaussian Process Models

The performance of Gaussian Process-based classification depends upon the chosen kernel used to capture the similarity between examples, as well as the kernel's hyperparameters, such as the length-scale, the noise variance, and other parameters determining local feature-based image similarity. Finding the right set of all these parameters can be a challenge. Many discriminative models (including SVMs) often use cross-validation, which is a robust measure but can be prohibitively expensive and problematic when we have few labeled data points. Learning in a Gaussian Process framework is equivalent to choosing the kernel hyperparameters of the covariance function. Ideally we would like to marginalize over these hyperparameters. While approaches based on Hybrid Monte Carlo have been explored to perform this marginalization [47], such techniques are relatively expensive.

[3] This method is referred to as least-squares classification in the literature (see Section 6.5 of [32]) and often demonstrates performance competitive with more expensive Gaussian Process classification methods that require approximate inference.

Empirical Bayes is a more computationally efficient alternative where the idea is to maximize the marginal likelihood or the evidence, which is nothing but the constant $p(\mathbf{t}_L|\mathbf{X})$ that normalizes the posterior. This methodology of tuning the hyperparameter is often called *evidence maximization*, and has been one of the favorite tools for performing model selection. Evidence is a numerical quantity and signifies how well a model fits the given data. By comparing the evidence corresponding to the different models (or hyperparameters that determine the model), we can choose the model and the hyperparameters suitable for the task.

The idea is to choose a set of hyperparameters $\Theta$ that maximize the evidence: $\hat{\Theta} = \arg\max_{\Theta} \log[p(\mathbf{t}_L|\mathbf{X}, \Theta)]$. Note that the log evidence $\log(p(\mathbf{t}_L|\mathbf{X}, \Theta))$ can be written as a closed form equation for the Gaussian noise model (GP regression):

$$\log p(\mathbf{t}_L|\mathbf{X}, \Theta) = -\frac{1}{2}\mathbf{t}_L^T(\sigma^2\mathbf{I} + \mathbf{K}_{LL})^{-1}\mathbf{t}_L - \\ \frac{1}{2}\log|\sigma^2\mathbf{I} + \mathbf{K}_{LL}| - Const.$$

This objective can be maximized using non-linear optimization techniques, such as gradient descent. In this work, we use gradient-descent to maximize evidence. The optimization procedure can perform multiple searches with different initializations to deal with the fact that the evidence will have multiple local optima.

This scheme of learning hyperparameters by maximizing evidence lets us find the correct parameters without the need of cross-validation. Further, this procedure can also be used to learn an ideal linear combination of covariance functions, which is a useful tool in practice to combine various local feature object categorization schemes. We show this combination strategy in Section 6.

## 5 Pyramid Match Kernel Gaussian Processes (GP-PMK)

To use GPs for object categorization, we need to define a suitable covariance function. We would like to exploit local feature methods for object and image representations. However, GP priors require covariance functions which are positive semi-definite (a Mercer kernel) and traditional covariance functions (e.g., RBF) are not suitable for representations that are comprised of sets of features.

We wish to define a GP with a covariance function based on a partial match distance function. The idea is to first represent an image as an unordered set of local features, and then use a matching over these sets of features to compute a smoothness prior between images. The optimal least-cost partial matching takes two sets of features, possibly of varying sizes, and pairs each point in the smaller set to a unique point in the larger one, such that the sum of the distances between the matched points is minimized. The cubic cost of the optimal matching makes it prohibitive for recognition with a large number of local image features, yet rich image descriptions comprised of densely sampled local features are known to often yield better recognition accuracy [31].

Therefore, rather than adopt a full partial match kernel for the GP prior, we use the Pyramid Match [14]. The Pyramid Match is a linear-time kernel function over unordered feature sets that approximates the similarity measured by the optimal partial matching, and it forms a Mercer kernel. A multi-resolution partition (pyramid) carves the feature space into increasingly larger regions. At the finest resolution level in the pyramid, the partitions are very small; at successive levels they continue to grow in size until a single partition encompasses the entire feature space. The insight of the Pyramid Match algorithm is to treat points which share a bin in this pyramid as being matched, and to use the histograms to read off the number of possible matches without explicitly searching for correspondences. Histogram intersection (the sum of the minimum number of points in a given histogram bin) is used to count the number of new matches that occur at each resolution level.

The input space $S$ contains sets of feature vectors drawn from feature space $\mathcal{F}$: $S = \{\mathbf{F}|\mathbf{F} = \{\mathbf{f}_1, \ldots, \mathbf{f}_m\}\}$, where each feature $\mathbf{f}_i \in \mathcal{F} \subseteq \Re^d$, and $m = |\mathbf{F}|$. For example, $\mathcal{F}$ might be the space of SIFT [23] descriptors ($d = 128$), or image coordinate positions ($d = 2$), etc.; a set $\mathbf{F}$ contains a collection of these descriptors extracted from a single image or object. An $L$-level histogram pyramid for input example $\mathbf{F} \in S$ is defined as: $\Psi(\mathbf{F}) = [H_0(\mathbf{F}), \ldots, H_{L-1}(\mathbf{F})]$, where $H_i(\mathbf{F})$ is a histogram vector formed over points in $\mathbf{F}$ using multi-dimensional bins.

The Pyramid Match Kernel (PMK) value between two input sets $\mathbf{F}_1$, $\mathbf{F}_2 \in S$ is defined as the weighted sum of the number of feature matches found at each level of their pyramids [14]:

$$\mathbf{K}_{\Delta}\left(\Psi(\mathbf{F}_1), \Psi(\mathbf{F}_2)\right) = \\ \sum_{i=0}^{L-1} w_i \Big(\mathcal{I}\left(H_i(\mathbf{F}_1), H_i(\mathbf{F}_2)\right) - \mathcal{I}(H_{i-1}(\mathbf{F}_1), H_{i-1}(\mathbf{F}_2))\Big),$$

where $\mathcal{I}$ denotes histogram intersection, and the difference in intersections across levels serves to count the number of new matches formed at level $i$ that were not already counted at any finer resolution level. Note that bins at level $i$ are always larger than those at level $i-1$. The weights are set to be inversely proportional to the size of the bins, in order to reflect the maximal distance two matched points could be from one another. As long as $w_i \geq w_{i+1}$, the kernel is Mercer.

We thus define a Pyramid Match Gaussian Process model (GP-PMK) using the prior

$$p(\mathbf{Y}|\mathbf{X}) \sim \mathcal{N}(0, \mathbf{K}_{\Delta}). \tag{7}$$

In contrast to previous GP priors, this prior is well-suited for visual category recognition as it naturally handles representations based on sets of local image features.

A variety of Pyramid Match Kernels (and thus GP priors) are possible, given that we have flexibility in choosing the interest operator used to sample local image regions, the type of descriptor used to describe each region, and the partitioning strategy used to form the pyramid histogram bins. To extract local features, we can exploit a wealth of interest operators designed to detect a sparse set of salient regions (e.g., [23,27,17]), or simply sample densely at regular intervals and at multiple scales. To describe each region or patch, we can choose from an array of descriptors designed to capture local texture while maintaining some invariance to small shifts and rotations, such as SIFT [23], shape context [3], or geometric blur [4].

For low-dimensional feature spaces, the partitions within each histogram $H_i$ may be placed at uniform intervals to divide the feature space into equally sized grid cells, as in [14, 21]. For higher-dimensional feature spaces it is better to place the partitions non-uniformly in a data-dependent manner, as described in [15]. To encode spatial position together with the region appearance, each feature $\mathbf{f}_i$ can be expanded to include both the image descriptor concatenated with the normalized image coordinate at which it occurred; however, doing so requires standardizing the dimensions carefully. An efficient way to incorporate both feature channels is to use the spatial pyramid match [21], a variant of the PMK that first quantizes the appearance feature descriptors to form a bag-of-words representation, and then sums over the PMK values for each word in the space of image coordinates. Depending on the image data, such choices are likely to influence the accuracy of the GP-PMK model. In the next section, we describe a technically sound strategy to combine all of these different kernels such that the resulting kernel is highly discriminatory.

## 6 Combining Multiple Covariance Functions

Given multiple kernels $\mathbf{K}^{(1)}, \cdots, \mathbf{K}^{(k)}$ we seek a linear combination of the base kernels such that the resulting kernel $\mathbf{K}$ has a good discriminatory power. Formally, we have

$$\mathbf{K} = \sum_{i=1}^{k} \alpha_i \mathbf{K}^{(i)}, \tag{8}$$

where $\boldsymbol{\alpha} = \{\alpha_1, .., \alpha_k\}$ are the weight parameters that we wish to optimize. We can take an evidence maximization approach as described in Section 4.4 to solve for these weights. In this case, the objective function $\mathcal{L}(\boldsymbol{\alpha})$ that we wish to minimize is the negative log evidence (that is, the marginal likelihood) of the model:

$$\arg\min_{\boldsymbol{\alpha}} - \log p(\mathbf{t}_L | \mathbf{X}, \boldsymbol{\alpha})$$

subject to: $\alpha_i \geq 0$ for $i \in \{0, .., k\}$.

The non-negativity constraints on $\boldsymbol{\alpha}$ ensure that the resulting $\mathbf{K}$ is positive-semidefinite and can be used in an GP formulation (or other kernel-based methods).

The proposed objective is a non-linear program and can be solved using any gradient-descent based procedure. In our implementation we use a gradient descent procedure based on the projected BFGS method using a simple line search. The gradients of the objective are efficient to compute and can be written as:

$$\frac{\delta \mathcal{L}(\boldsymbol{\alpha})}{\delta \alpha_i} = -\frac{1}{2} \mathbf{t}_L^T \mathbf{A}^{-1} \mathbf{K}_{LL}^{(i)} \mathbf{A}^{-1} \mathbf{t}_L + \frac{1}{2} \mathrm{Tr}(\mathbf{A}^{-1} \mathbf{K}_{LL}^{(i)}),$$

where $\mathbf{A} = \sigma^2 \mathbf{I} + \mathbf{K}_{LL}$.

Once the parameters $\boldsymbol{\alpha}$ are found, then the resulting linear combination of kernels ($\mathbf{K}$) can be used for classification. By selecting the kernel weights within the GP framework, we allow a user to provide several feature choices and PMK kernel variants that seem plausible, with the system itself selecting the most discriminative combination.

## 7 Active Learning for Object Categorization

In this section we consider the scenario where our visual category learner has access to a pool of unlabeled data $\mathbf{X}_U = \{\mathbf{x}_{n+1}, .., \mathbf{x}_{n+m}\}$. The task in active learning is to seek the label for one of these examples and then update the classification model by incorporating it into the existing training set. The goal is to select the sample that would maximize the benefit in terms of the discriminatory capability of the system.

With non-probabilistic classification schemes, a popular heuristic for establishing the confidence of estimates and identifying points for active learning is to simply use the distance from the classification boundary (margin). This approach can also be used with GP classification models, by inspecting the magnitude of the posterior mean $\bar{y}_u$: one would then choose the next point $\mathbf{x}^*$ as $\arg\max_{\mathbf{x}_u \in \mathbf{X}_U} \bar{y}_u$.

However, GP classification provides us with both the posterior mean as well as the posterior variance for the unknown label $t_u$. An alternative criteria could be to look at the variances and select the point that has the maximum variance, i.e. $\mathbf{x}^* = \arg\max_{\mathbf{x}_u \in \mathbf{X}_U} \Sigma_u$. However such an approach does not consider the mean $\bar{y}_u$ at all! Further, the expression for $\Sigma_u$ does not consider labels from the annotated training data; this scheme uses only a very limited amount of information.

**Table 1** Active Learning Criteria

| Method | Criteria |
|---|---|
| Distance<br>from Boundary (SVM) | $\mathbf{x}^* = \arg\min_{\mathbf{x}_u \in \mathbf{X}_U} \lvert\bar{y}_u\rvert$ |
| Variance | $\mathbf{x}^* = \arg\max_{\mathbf{x}_u \in \mathbf{X}_U} \mathit{\Sigma}_u$ |
| Uncertainty (GP) | $\mathbf{x}^* = \arg\min_{\mathbf{x}_u \in \mathbf{X}_U} \frac{\lvert\bar{y}_u\rvert}{\sqrt{\mathit{\Sigma}_u + \sigma^2}}$ |

We therefore propose an approach which considers both the posterior mean as well as the posterior variance. Specifically, we select the next point according to:

$$\mathbf{x}^* = \arg \min_{\mathbf{x}_u \in \mathbf{X}_U} \frac{|\bar{y}_u|}{\sqrt{\mathit{\Sigma}_u + \sigma^2}}, \tag{9}$$

where $\sigma^2$ is the noise model variance. This formulation considers uncertainty in the labeling $\mathbf{x}_u$ as $\pm 1$. Note that the predictive distribution for $t_u$ is a Gaussian; however, we are interested in the binary label decided according to the sign of $t_u$. To this end we should consider the value $p(t_u \geq 0) = \phi(\frac{\bar{y}_u}{\sqrt{\mathit{\Sigma}_u + \sigma^2}})$, where $\phi(\cdot)$ denotes the cdf of a standard normal distribution, to provide the hard label $\pm 1$. Further, we are interested in selecting those samples where the uncertainty is maximum. The points where the classification model is most uncertain should have a value for $p(t_u \geq 0)$ that is close to 0.5—equivalently, a value of $\frac{|\bar{y}_u|}{\sqrt{\mathit{\Sigma}_u + \sigma^2}}$ that is very close to zero. Thus, the criterion in Equation 9 chooses the unlabeled point where the classification is the most uncertain.

We summarize the methods for identifying points to be labeled in Table 1, with our strategy given in the third row. Our active learning approach looks at all the points before choosing the points to actively label; thus it considers the whole dataset instead of just looking at individual points. Further, this scheme considers both the distance from the boundary as well as the variance in selecting the points; this is only possible due to the availability of the predictive distribution in GP regression. In results below we show that in practice we can effectively choose useful examples to label, allowing our active GP approach to fare much better with minimal labeled data than a "passive" random selection scheme.

## 8 Experiments and Results

**Table 2** Recognition accuracy on the Caltech-101 using 15 labeled points per class.

| Method | GP | SVM | 1-NN |
|---|---|---|---|
| | 1-vs-All | 1-vs-All | |
| Dense PMK | 52.13 ± 0.69 | 48.77 ± 0.95 | 24.20 ± 0.48 |
| Spatial PMK | 51.90 ± 0.78 | 54.26 ± 0.65 | 41.10 ± 0.78 |
| AppColour | 44.54 ± 1.05 | 44.88 ± 1.09 | 37.42 ± 0.88 |
| AppGray | 56.04 ± 1.10 | 57.87 ± 1.00 | 46.82 ± 0.58 |
| Shape 180 | 49.64 ± 0.70 | 48.86 ± 1.00 | 35.35 ± 0.79 |
| Shape 360 | 50.92 ± 0.90 | 51.80 ± 0.75 | 34.34 ± 0.84 |
| GB | 64.15 ± 0.76 | 65.87 ± 0.92 | 45.58 ± 0.79 |
| GBDist | 58.81 ± 0.58 | 65.91 ± 0.66 | 50.23 ± 0.66 |
| Combination | 88.65 ± 0.53 | -NA- | -NA- |

In this section we report results from experiments to demonstrate 1) the effectiveness of the GP-PMK classification framework, 2) the ability of the proposed framework to do good kernel combination, and 3) how active learning can guide the learning procedure to select critical examples to be labeled. We show how kernel combination and active learning with Gaussian Process priors yield classifiers which can learn object categories from relatively few examples.

*Datasets and Implementation Details*

We performed supervised and active learning experiments on two different datasets that are considered standards for the object categorization task: the Caltech-4 dataset and the Caltech-101 dataset (which is a superset of Caltech-4). We compute the similarity between all pairs of images in each database using the PMK. We used LIBSVM [6] for the SVM baseline tests. In our experiments we set the noise model variance $\sigma = 10^{-5}$ for the Gaussian Process models and fix $C = 10000$ for SVM models. These parameter values worked well; we experimented with other values but found that both SVM and GP classification schemes were fairly insensitive to the choice of these parameters.

The object categorization task is a multi-class problem ($n_{class} = 101$ and $n_{class} = 4$ for the Caltech-101 and the Caltech-4, respectively). To handle multiple classes we use the one-vs-all formulation, where we choose the label corresponding to the class with maximum value of the soft label $y$. For kernel combination under the one-vs-all classification scheme we assume a joint model by summing the log evidence over all the binary classification problems. For multiclass active learning in every round we select one example from each of the one-vs-all classifiers, thus adding $n_{class}$ examples every time.

The Caltech-4 database contains 3188 images with four object classes. There are 1155 rear views of cars, 800 images of airplanes, 435 images of frontal faces, and 798 images of motorcycles. The second database is the Caltech-101 database of 101 object categories [10]; there are 8677 images in this data set, with between 31 to 800 images for each of the 101 categories. Our experiments for kernel combination use 30 images per class (3030 images in total), and are exactly same as the ones used in [43]. We perform active learning experiments using the complete Caltech-101 dataset.

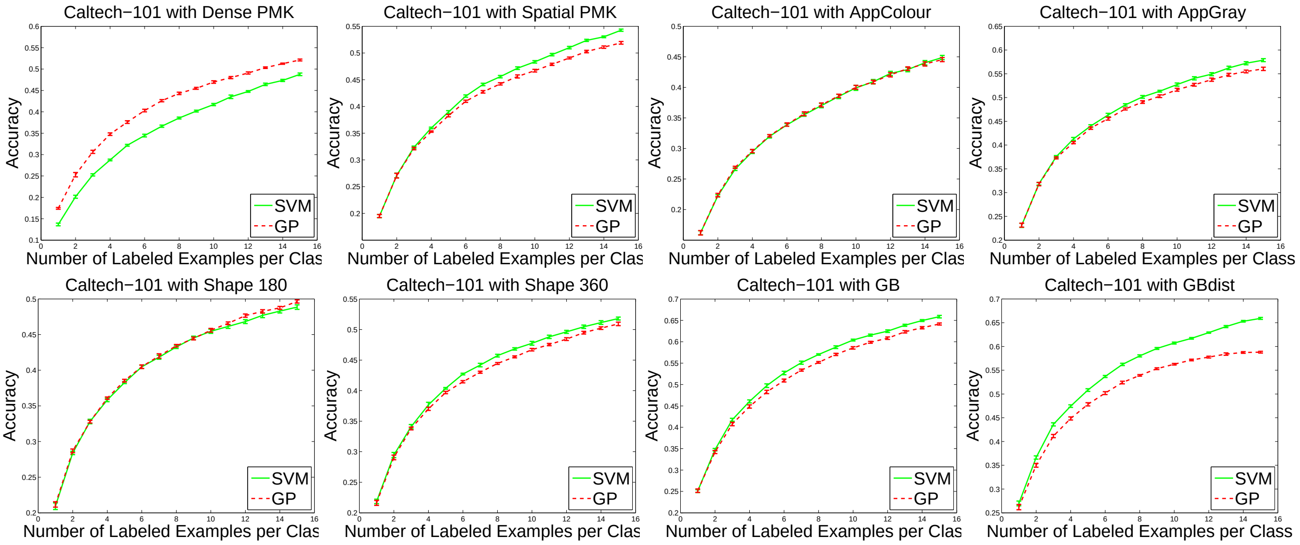

**Fig. 3** Performance comparison of GP and SVM classification on each of the eight different kernels used in this work. The figures show that using the GP framework with supervised learning achieves comparable performance to that of SVMs, but with the additional benefit of retaining a probabilistic formulation.

Further, we consider various shape and appearance features and sampling strategies, which are useful to capture the intra-class variation present in the Caltech-101 images. Specifically, we look at the following eight combinations of matching kernels and features:

- **Dense PMK:** the PMK with uniformly shaped pyramid bins, using SIFT descriptors extracted densely from the images at every 8th pixel in the image from a region of 16 pixels in diameter, with each SIFT descriptor concatenated with its normalized image position. We use PCA to reduce the dimensionality of the SIFT descriptors to 10 before adding the position, yielding features having a total of 12 dimensions. See [14] for details.
- **Spatial PMK:** The spatial variant of Dense-PMK. We take the same raw SIFT features, but quantize them into visual words, and then build one pyramid per word, each with uniform bins in the space of image coordinates. See [21] for details.
- **AppColour:** SIFT descriptors are extracted for each component of the HSV color space representation of the image, with all features sampled on a regular grid and at four fixed scales. As above, these are quantized into visual words, and the pyramid kernel is applied per word in the space of image coordinates. See [5] for details.
- **AppGray:** Same as AppColour, except features are extracted from the grayscale images.
- **Shape 180**: Histograms of oriented gradients are matched using a spatial pyramid kernel. Edges are computed using the Canny edge detector followed by Sobel filtering for gradients; the gradients are discretized into orientation histogram bins in the range [0, 180] with soft voting. See [5] for details.
- **Shape 360:** Same as Shape 180 except that the orientation bins are in the range [0, 360].
- **GB:** The Geometric Blur feature of [4] is extracted at sampled edge points. For the kernel values, the exact correspondences are computed based on the average minimum distance between points in the two input sets of features, as in [49].
- **GBdist:** Same as GB, except the feature representation has an additional geometric distortion term.

The kernel matrices for this dataset using each of these variants were provided directly by the authors. All kernels reflect variations on the PMK and feature spaces, except for the two using the geometric blur; these kernels do not approximate the matching, but rather compute an explicit correspondence between the features.

In all our experiments in which comparisons are made against other methods, we follow the standard testing protocol, where a given number of training images (say 15) are taken from each class at random, and the rest of the data is used for testing. The mean recognition rate per class is used as a metric of performance. Note that this metric ensures that the recognition accuracies are such that classes with large numbers of examples are not favored. This process is repeated 10 times and the average correctness rate is reported.

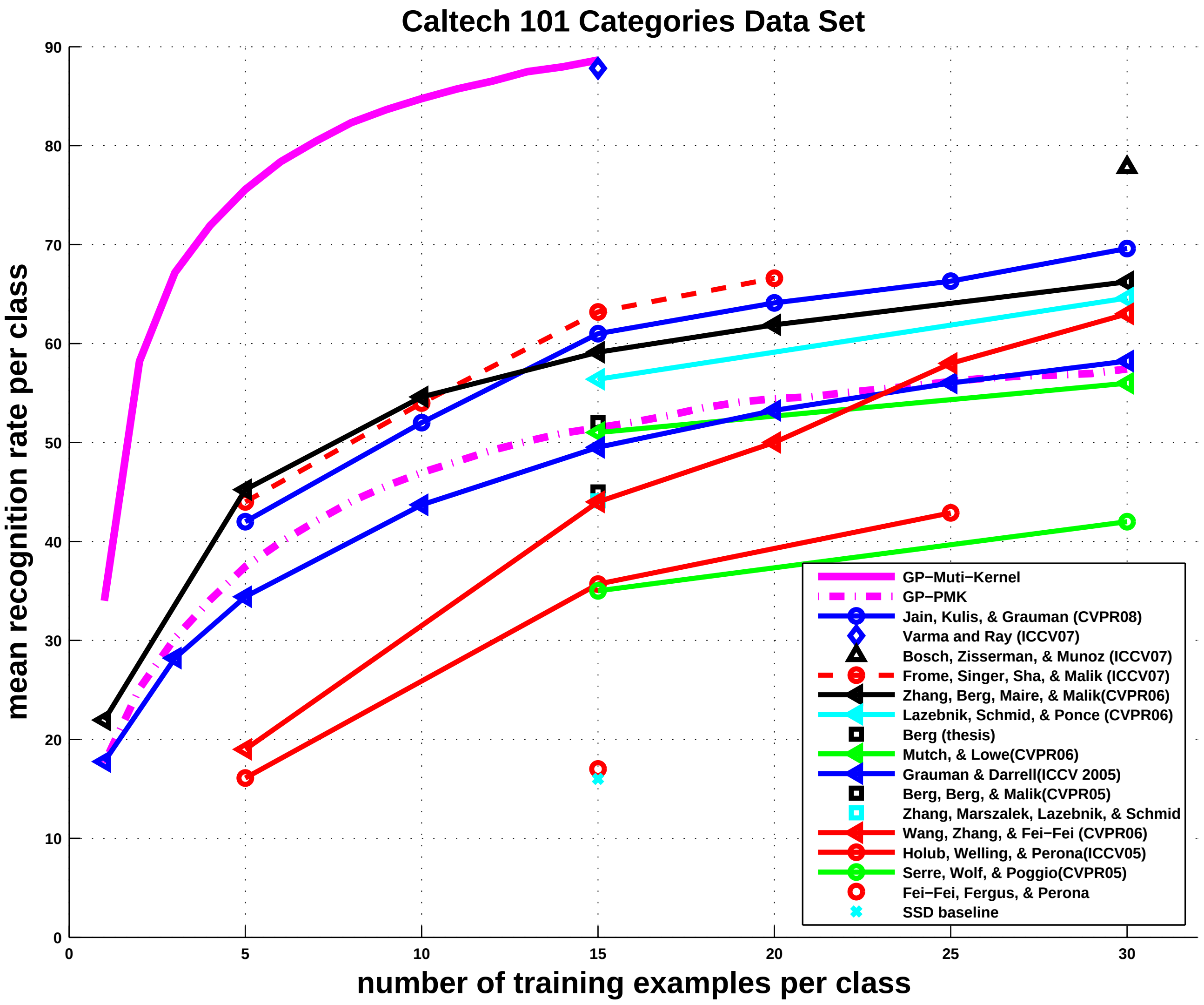

**Fig. 4** Performance comparison of GP-PMK and GP-Multi-Kernel classification with reported results from the literature. Using the same PMK kernel and features, our GP-PMK approach outperforms earlier SVM-PMK results [14]. Furthermore, with an appropriate combination of various kernels (GP-Multi-Kernel), we obtain recognition performance very competitive with the state-of-the-art. In fact, we believe our approach is yielding the highest accuracy to-date on this dataset when learning with few training examples (1 to 10 labeled examples per class).

*Classification and Kernel Combination*

First, we explore classification performance on individual kernels using different classification strategies. Figure 3 graphically shows the performance of classification with Gaussian Process as compared to an SVM classifier. From the eight graphs we can observe that overall the performance obtained using either the GP or the SVM is very similar. However, we note some deviations in performance: for example GP is significantly better on the Dense-PMK, whereas SVM performs very well with GBdist. We also compare performance of these methods with 1-Nearest Neighbor as a baseline; Table 2 summarizes the accuracy obtained with 15 labeled points per class. Both GP and SVM perform significantly better than the 1-Nearest Neighbor classifier. We find these experiments encouraging, since they indicates that we need not give up the accuracy of other well-used discriminative methods (like the SVM) in order to gain the other benefits of having the probabilistic GP model.

In general, the superior performance of a particular classification algorithm with a specific kernel might be due to several reasons. For any classification strategy to work well, the underlying data must support the assumptions made by the model; whenever the data is favorable to the assumptions of a method, then we can hope that the algorithm will perform well. The point we wish to make here is that GP classification can often provide comparable or slightly improved classification performance when compared to SVMs; we do

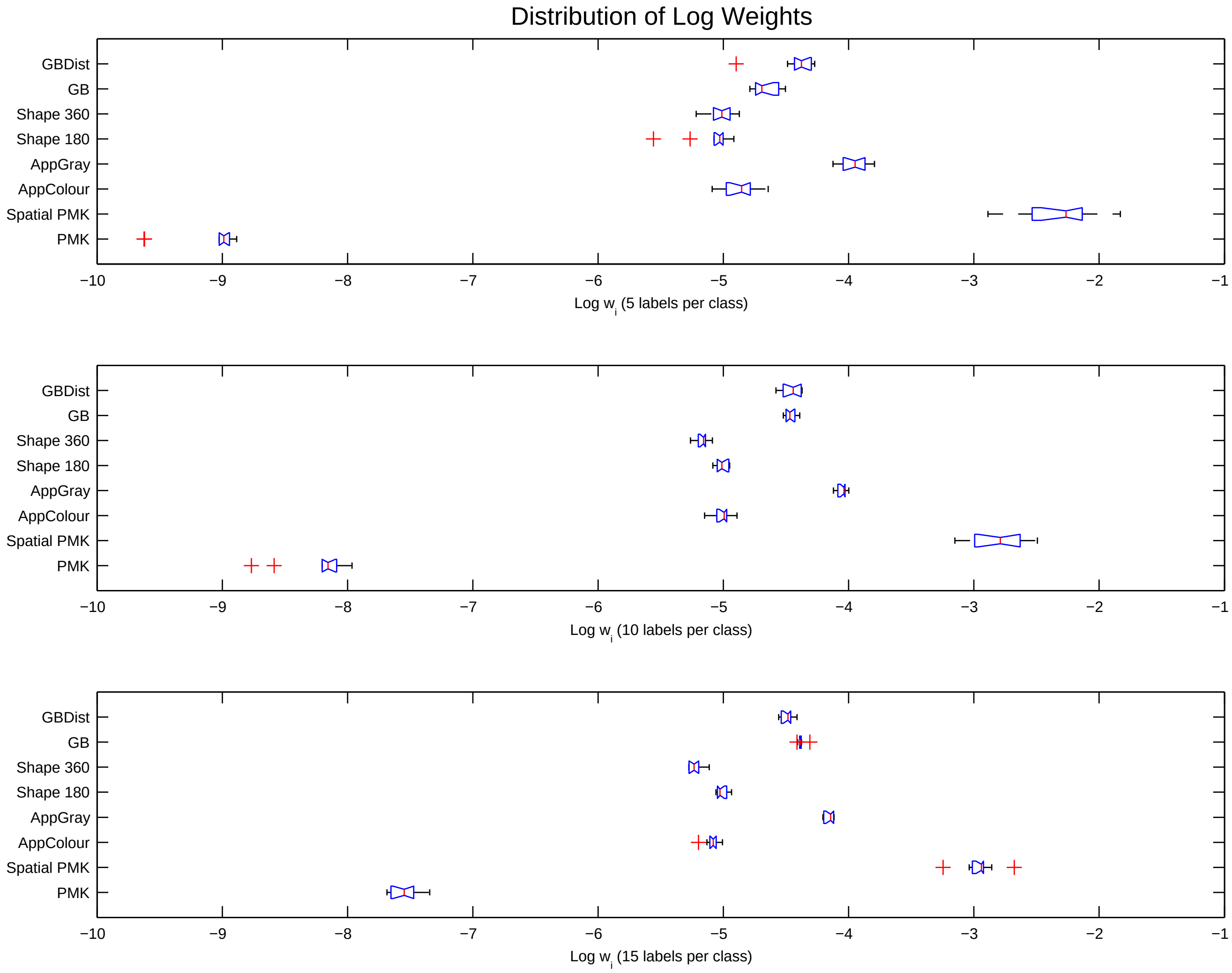

**Fig. 5** Distribution of weights plotted with MATLAB's *boxplot* command, showing the kernel combinations for various sizes of training set. All the different kernels are adding discriminatory power to the classification task. From these plots we see that the weights can be learned effectively even with a small number of training examples, as the relative weights are fairly consistent across the plots from top (5 training examples per class) to bottom (15 examples per class).

**Table 3** Accuracy reported with kernel combination on the Caltech-101.

| Method | Accuracy |
|---|---|
| **Ours** | **88.65 ± 0.53** |
| Varma and Ray [43] | 87.82 ± 1.00 |
| Bosch et al. [5] | 71.40 ± 0.80 |
| Frome et al. [13] | 60.30 ± 0.70 |
| Lin et al. [22] | 59.80 ± NA |
| Zhang et al. [49] | 59.08 ± 0.37 |
| Kumar and Sminchisescu [19] | 57.83 ± NA |

not have to lose accuracy to gain the predictive uncertainty offered by probabilistic recognition models.

Next, we compare GP-PMK classification with state-of-the-art supervised visual category learning methods that have been tested on the Caltech-101. Figure 4 shows the performance of an SVM and the classification with GP priors using the PMK along with other recent methods using the same evaluation methodology. The PMK was also earlier used in [14] with SVMs. Similar to our findings in figure 3, we show in figure 4 that classification with GP-PMK and an identical kernel (Dense-PMK) actually slightly outperforms the SVM, thus demonstrating the value in the proposed approach.

We also plot the recognition results obtained with the combination of kernels learned via evidence maximization (magenta curve). At 15 points per class we achieve 88.65% accuracy, which shows the power of combining different correspondence kernels in the probabilistic framework. In fact with just six labeled examples per class the combined kernel achieves an accuracy of 78.37% beating recognition performance by all of the other methods trained with an in-

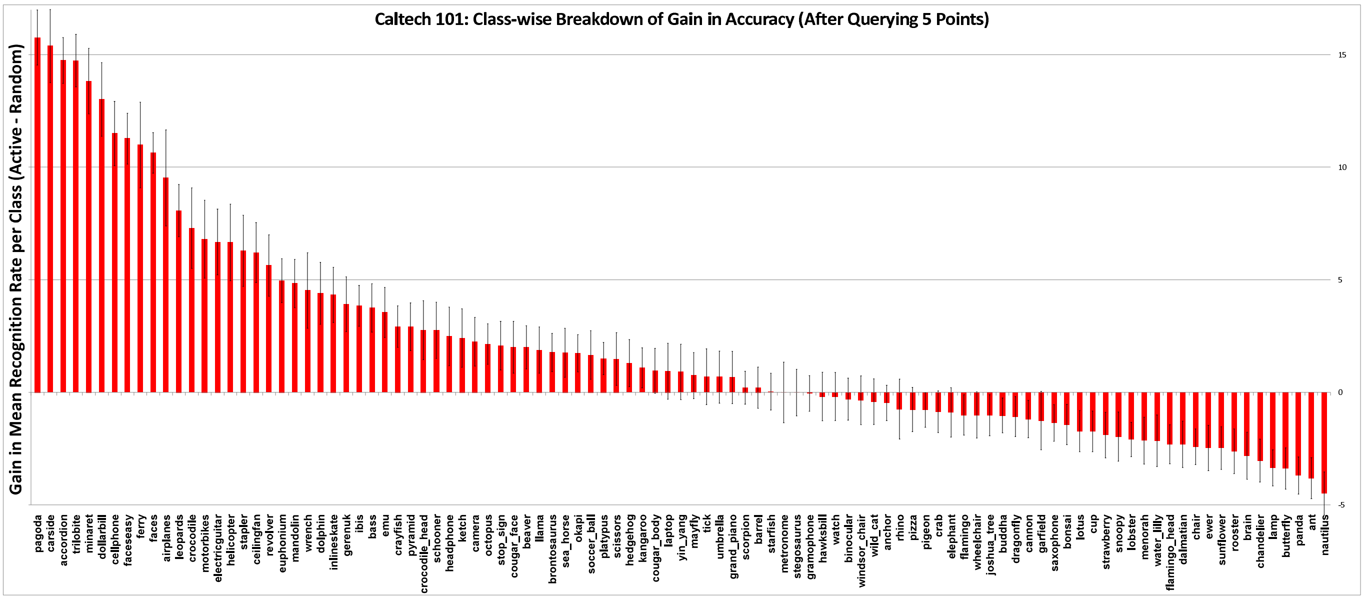

**Fig. 6** Average gain in performance over random selection when choosing five points using active learning. The graph shows the mean and the standard error for 100 runs for all the object classes in Caltech-101 database using GP-PMK.

dividual kernel with any size of the training set. Table 3 provides a direct comparison with results previously reported for other approaches that include kernel combination. Closest to our accuracy of 88.65% is the technique of [43], who report an accuracy of 87.82% on the Caltech-101 data using SVMs with multiple kernel learning.

We chose to work with this dataset due to the variety of categories it contains, as well as the large number of existing results published using it. The accuracy of our GP framework is a compelling result, with what appear to be some of the best performance numbers obtained to-date. We would like to point out that we have certainly benefitted from progress in recent years due to other work that has, for example, determined good features that are applicable for this data. Nonetheless, it is an encouraging result that our method can automatically learn a good combination from a variety of existing matching kernels, and greatly improve the state-of-the-art with quite small labeled training sets.

As a final experiment with supervised GP-PMK, we investigate the distribution of learned weights attributed to the different matching kernels. Figure 5 displays the range of weights over the 10 different runs of the algorithm using MATLAB's *boxplot*, which is a graphical representation of the statistics of the log weights [4]. Each row corresponds to a kernel, and the red line in each row denotes the median over the 10 different runs. The end lines are at the lower and upper quartile values, and the outliers are data with values beyond the ends of the whiskers. We show boxplots for runs with five, 10 and 15 training examples per class. From the figure we see that this distribution is fairly similar for different sizes of training sets. This highlights that we can hope to learn the kernel weights even with very few data points. Further, we also notice that the weight corresponding to the spatial PMK is often fairly high. However, we cannot interpret the weights directly as a measure of importance. This is due to the fact that the scale of each kernel is different; thus, the weight encompasses both the discriminatory power as well as automatic scale adjustment.

### *Active Learning for Object Categorization*

In this section, we show the value of active learning in selecting examples to annotate. For these experiments, we test the classification performance on a validation set that includes 10 examples from each class. We first consider the *binary* problem of detecting an object class. Starting with one labeled example per class, the procedure chooses the next image to query from the set of images not in the validation set. We compare the active version of the GP classification with a version that selects the points to query randomly. We again use the mean classification rate per class to compare the methods. We repeat this procedure for 100 different validation sets.

Figure 6 shows the gain in performance on all the 101 binary problems, averaged over the 100 runs, made by the active learning scheme on the validation set after 5 examples are chosen. We can clearly see that for most of the categories there is a significant positive gain showing the benefit of the active learning scheme. Further, figure 7 shows the performance on various binary problems as we increase the size of the training set. The figure depicts that the active learning scheme quickly exploits the uncertainty in its estimates to select appropriate examples to seek the annotation for. The random policy on the other hand performs poorly. The fact

[4] We use log weights for clarity in plotting.

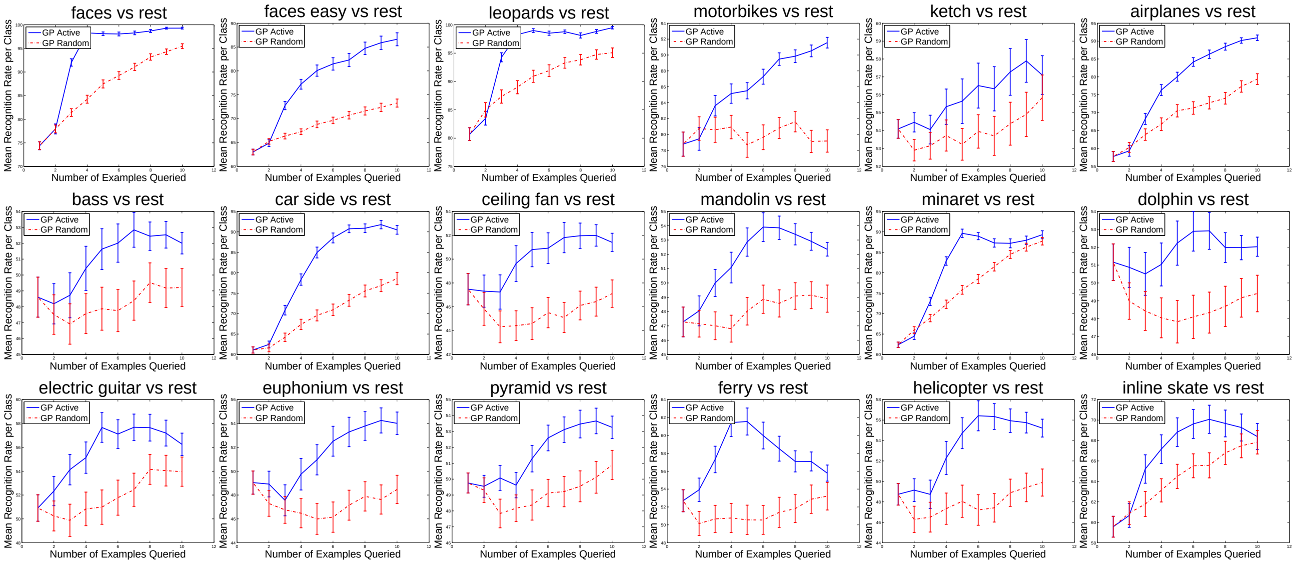

**Fig. 7** Performance comparison of GP classification with active learning and GP with random supervision for various object detection problems (binary) in the Caltech-101 database.

that the Caltech-101 dataset has unbalanced numbers of examples per category affects the random sampling policy; it does not work well in these unbalanced scenarios because the training set will usually be skewed towards one class, resulting in poor accuracy. However, selecting points via active learning focuses on points with maximum uncertainty, irrespective of their label, making the procedure highly effective.

Next we describe active learning experiments with the Caltech-4 dataset using multiple feature sampling and pyramid partitioning strategies. The goal here was to investigate the benefits of the proposed scheme across the spectrum of kernels available for the task of object categorization. For this experiment we again experimented with three different flavors of the Pyramid Match Kernel. Besides Dense PMK, we also used PMK computed using only sparse interest points where salient points in the images are detected with a Harris-Affine interest operator (Harris PMK). The third PMK variant was vocabulary guided (Vocabulary Guided PMK) where the features were binned non-uniformly in a data-dependent manner, as in [15].

Figure 8 compares different classification approaches on the Caltech-4 database for different kinds of kernels. Essentially, the plot shows mean classification accuracy per class as we vary the total number of examples in the training data. The images not in the training set are considered as the test set to compute the classification performance. We plot the performance of the SVM and the GP classification with and without active learning. We start with zero labeled points. For the SVM and supervised GP without active learning, we randomly select points as we increase the size of the training set, whereas for the active learning GP classification we use uncertainty to guide its sample selection process. This process was repeated 40 times and figure 8 shows the mean performance. The errorbars denote the standard error and non-overlapping errorbars signify difference in performance levels with 95% confidence.

From figure 8 we observe that GP classification again performs competitively with the SVM, and using active learning further improves the performance. In fact we can see that a mean accuracy per class close to 90% can be obtained with just 20 labeled examples, whereas the non-active learners achieve around 85% accuracy for the same amount of labeled data. This demonstrates that active learning can provide a significant boost in accuracy across different flavors of kernels and feature types used in object categorization. Further, the scheme also makes it possible for the learning algorithm to learn the object classes even with very few labeled examples.

Table 4 shows the confusion matrix resulting after incorporating only 120 examples in the training set using the active learning methodology with Dense PMK. We obtain an overall accuracy of 98.48%, which demonstrates the effectiveness of the framework. The completely supervised GP classification and SVM achieved a mean classification accuracy per class of 95.6% and 95.19%, respectively. This shows that our active learning strategy allows us to learn object categories much more effectively than traditional supervised classification.

## 9 Discussion

The experiments in this paper indicate that classification using GP priors performs competitively with SVM on the ob-

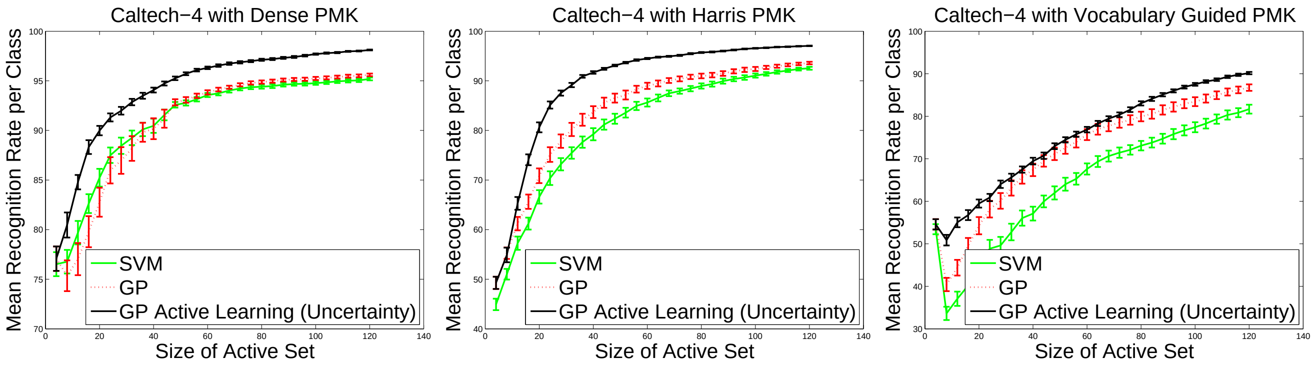

**Fig. 8** Active learning on the Caltech-4 database using different kinds of Pyramid Match Kernels and feature types. In each case, our active learning approach provides significant gains over traditional passive approaches that label points at random, while the GP classification even shows some gains over the SVM.

**Table 4** Confusion matrix obtained for Caltech-4 database using active learning with the Pyramid Match Kernel (Dense PMK). (120 labeled images, mean accuracy over all the classes = 98.48%).

| | Recognized Class | | | |
|---|---|---|---|---|
| True Class | **Cars** | **Faces** | **Airplanes** | **Motorbikes** |
| **Cars** | 1121 | 0 | 0 | 1 |
| **Faces** | 0 | 416 | 0 | 2 |
| **Airplanes** | 0 | 2 | 753 | 20 |
| **Motorbikes** | 10 | 0 | 10 | 733 |

ject categorization task with Caltech-101 and Caltech-4 data. Of course, these experiments cannot serve as conclusive proof that classification using a GP prior is inherently superior than other classification techniques or vice-versa. Yet for this object categorization task and data, the underlying data density is favorable to the assumptions of the classification model we are using. The experiments in this paper strongly suggest that there is a value in looking at GP classification models for object categorization.

Another important aspect of our framework lies in its seamless extension to kernel combination and active learning. The probabilistic paradigm allows us to exploit the evidence maximization framework to principally combine different correspondence kernels. Furthermore, the Bayesian formulation lets us incorporate measures such as uncertainty, variance, and expected information gain that could be highly valuable in guiding a supervised learning procedure. One of the challenges in computer vision is the ability to learn object categories with a low number of examples. Humans are able to learn object categories and generalize from a very small number of examples. However, current machine vision systems are far from achieving performance akin to humans. One of the principal differences among humans and existing object classification systems is that humans have the ability to actively seek supervision from the environment and other sources of information. We believe that active learning might enable us to move towards vision systems that require few examples to learn successfully.

## 10 Conclusion and Future Work

We have presented a discriminative probabilistic framework based on Gaussian Process priors and the local feature-based correspondence kernels, and have shown its utility for visual category recognition. Gaussian Process regression provides a principled framework to combine different correspondence kernels, which results in performance superior to individual kernels. Further, the modeling with Gaussian Process priors provides direct estimates of prediction uncertainty using a smoothness prior that captures a correspondence-based notion of similarity between sets of local image features. We introduced an active learning method for visual category recognition based on the GP-PMK uncertainty estimates, and showed empirically that active learning can be used to achieve very good recognition results using far fewer training images than standard supervised learning approaches.

We plan to extend the framework to adopt non-Gaussian noise models, and investigate other active learning formulations such as value of information and/or criteria previously developed for sparsifying GPs [20]. By incorporating decision-theoretic formulations we should be able to learn object categories within a given budget. We also plan to extend the model to handle multiple objects in the same image, incorporate semi-supervised learning, and explore sparse GP techniques for large training sets.

**Acknowledgements** We thank the following for providing kernel matrices: Alex Berg, Anna Bosch, Jitendra Malik and Andrew Zisserman. We also wish to thank Manik Varma for his help in obtaining the required data and for many helpful discussions.

## References

1. *http ://labelme.csail.mit.edu/*.
2. Y. Abramson and Y. Freund. Active learning for visual object recognition. Technical report, UCSD, 2004.
3. S. Belongie, J. Malik, and J. Puzicha. Matching shapes. In *ICCV*, 2001.

4. A. Berg and J. Malik. Geometric blur for template matching. In *CVPR*, 2001.
5. A. Bosch, A. Zisserman, and X. Munoz. Representing shape with a spatial pyramid kernel. In *CIVR*, 2007.
6. C. Chang and C. Lin. *LIBSVM: a library for SVMs*, 2001.
7. E. Y. Chang, S. Tong, K. Goh, and C. Chang. Support vector machine concept-dependent active learning for image retrieval. *IEEE Transactions on Multimedia*, 2005.
8. T. Evgeniou, M. Pontil, and T. Poggio. Regularization networks and support vector machines. *Advances in Computational Mathematics*, 13(1), 2000.
9. B. T. F. Moosmann and F. Jurie. Fast discriminative visual codebooks using randomized clustering forests. In *NIPS*, 2007.
10. L. Fei-Fei, R. Fergus, and P. Perona. One-shot learning of object categories. *IEEE Transaction on Pattern Recognition and Machine Intelligence*, 2006.
11. R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *CVPR*, 2003.
12. Y. Freund, H. S. Seung, E. Shamir, and N. Tishby. Selective sampling using the query by committee algorithm. *Machine Learning*, 28(2-3), 1997.
13. A. Frome, Y. Singer, F. Sha, and J. Malik. Learning globally-consistent local distance functions for shape-based image retrieval and classification. In *ICCV*, 2007.
14. K. Grauman and T. Darrell. The pyramid match kernel: Discriminative classification with sets of image features. In *ICCV*, 2005.
15. K. Grauman and T. Darrell. Approximate correspondences in high dimensions. In *NIPS*, 2006.
16. K. Grauman and T. Darrell. Unsupervised learning of categories from sets of partially matching image features. In *CVPR*, 2006.
17. T. Kadir and M. Brady. Scale saliency : A novel approach to salient feature and scale selection. In *International Conference Visual Information Engineering*, 2003.
18. A. Kapoor, K. Grauman, R. Urtasun, and T. Darrell. Active learning with Gaussian Processes for object categorization. In *ICCV*, 2007.
19. A. Kumar and C. Sminchisescu. Support kernel machines for object recognition. In *ICCV*, 2007.
20. N. Lawrence, M. Seeger, and R. Herbrich. Fast sparse Gaussian Process method: Informative vector machines. *NIPS*, 2002.
21. S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006.
22. Y. Y. Lin, T. Y. Liu, and C. S. Fuh. Local ensemble kern el learning for object category recognition. In *CVPR*, 2007.
23. D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2), 2004.
24. D. MacKay. Information-based objective functions for active data selection. *Neural Computation*, 4(4), 1992.
25. A. K. McCallum and K. Nigam. Employing EM in pool-based active learning for text classification. In *ICML*, 1998.
26. K. Mikolajczyk and C. Schmid. Indexing Based on Scale Invariant Interest Points. In *ICCV*, 2001.
27. K. Mikolajczyk and C. Schmid. Scale and Affine Invariant Interest Point Detectors. *IJCV*, 1(60):63–86, October 2004.
28. T. P. Minka. *A Family of Algorithms for Approximate Bayesian Inference*. PhD thesis, MIT, 2001.
29. I. Muslea, S. Minton, and C. A. Knoblock. Active + semi-supervised learning = robust multi-view learning. In *ICML*, 2002.
30. D. Nister and H. Stewenius. Scalable Recognition with a Vocabulary Tree. In *CVPR*, 2006.
31. E. Nowak, F. Jurie, and B. Triggs. Sampling strategies for bag-of-features image classification. In *ECCV*, 2006.
32. C. E. Rasmusen and C. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.
33. M. Seeger. Gaussian Processes for machine learning. *International Journal of Neural Systems*, 14(2), 2004.
34. Y. Shen, A. Ng, and M. Seeger. Fast Gaussian Process regression using kd-trees. In *NIPS*, 2006.
35. J. Sivic, B. Russell, A. Efros, A. Zisserman, and W. Freeman. Discovering object categories in image collections. In *ICCV*, 2005.
36. J. Sivic and A. Zisserman. Video Google: a text retrieval approach to object matching in videos. In *ICCV*, 2003.
37. E. Snelson and Z. Ghahramani. Sparse Gaussian Processes using pseudo-inputs. In *NIPS*, 2006.
38. E. Sudderth, A. Torralba, W. Freeman, and A. Willsky. Describing visual scenes using transformed dirichlet processes. In *NIPS*, 2005.
39. S. Tong and D. Koller. Support vector machine active learning with applications to text classification. In *ICML*, 2000.
40. R. Urtasun and T. Darrell. Local probabilistic regression for activity-independent human pose inference. In *CVPR*, 2008.
41. R. Urtasun, D. J. Fleet, and P. Fua. Gaussian Process dynamical models for 3d people tracking. In *CVPR*, 2006.
42. R. Urtasun, D. J. Fleet, A. Hertzman, and P. Fua. Priors for people tracking from small training sets. In *ICCV*, 2005.
43. M. Varma and D. Ray. Learning the discriminative power-invariance trade-off. In *ICCV*, 2007.
44. L. von Ahn and L. Dabbish. Labeling images with a computer game. In *ACM CHI*, 2004.
45. L. von Ahn, R. Liu, and M. Blum. Peekaboom: A game for locating objects in images. In *ACM CHI*, 2006.
46. C. Wallraven, B. Caputo, and A. Graf. Recognition with local features: the kernel recipe. In *ICCV*, 2003.
47. C. Williams and D. Barber. Bayesian classification with Gaussian Processes. *IEEE Transaction on Pattern Recognition and Machine Intelligence*, 20(12):1342–1351, 1998.
48. O. Williams. A switched Gaussian Process for estimating disparity and segmentation in binocular stereo. In *NIPS*, 2006.
49. H. Zhang, A. Berg, M. Maire, and J. Malik. SVM-KNN: Discriminative nearest neighbor classification for visual category recognition. In *CVPR*, 2006.
50. X. Zhu, J. Lafferty, and Z. Ghahramani. Combining active learning and semi-supervised learning using Gaussian fields and harmonic functions. In *Workshop on The Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining at ICML*, 2003.