Fast Sparse Gaussian Processes Learning for Man-Made Structure Classification

Hang Zhou
Institute for Vision Systems Engineering, Dept Elec. & Comp. Syst. Eng.
PO Box 35, Monash University, Clayton, VIC 3800, Australia
hang.zhou@eng.monash.edu.au

David Suter
Institute for Vision Systems Engineering, Dept Elec. & Comp. Syst. Eng.
PO Box 35, Monash University, Clayton, VIC 3800, Australia
d.suter@eng.monash.edu.au

Abstract

Informative Vector Machine (IVM) is an efficient fast sparse Gaussian processs (GP) method previously suggested for active learning. It greatly reduces the computational cost of GP classification and makes the GP learning close to real time.

We apply IVM for man-made structure classification (a two class problem). Our work includes the investigation of the performance of IVM with varied active data points as well as the effects of different choices of GP kernels. Satisfactory results have been obtained, showing that the approach keeps full GP classification performance and yet is significantly faster (by virtue if using a subset of the whole training data points).

1 Introduction

We aim to develop an efficient way of classifying man-made structures from natural scenes by applying the fast GP approximation as an online learning method.

Gaussian Process (GP) classification models the posterior directly, thus relaxing the strong assumption of conditional independence of the observed data (generally used in a generative model). However, GP has $O(N^3)$ computational complexity with respect to the number of training data points N .

Different approaches have been proposed to deal with this problem. Csato and Opper developed a sparse representation by minimization of KL divergences between the approximate posterior and a sparse representation [1]. Snelson and Ghahramani presented a sparse approximation method with M pseudo-input points which are learnt by a gradient based optimization [2]. Lawrence, Seeger and Herbrich proposed a sparse GP method that employs greedy selections to choose the subset of the training data points maximizing a differential entropy score [3] (which is more simple and efficient to implement compared with other similar methods). Using this enables us to tackle the issue of kernel selection and to begin to tackle the questions of construction of on-line learning support (such as active data selection).

For man-made structure classification on 2D image data, typical approaches are based on Bayesian generative models as proposed in [4] and [5]. The generative model [4, 5] models the joint probability of the observed data and the related labels. Data is conditionally independent given the class labels [4] which is not true for man-made structures with obvious neighbouring dependencies. The TSBN generative model described in [4] is more for outdoor general scene segmentation rather than for man-made structure specifically. Hebert and Kumar [5] proposed a generative Multi-Scale Random Field (MSRF) model which extracts image block features that capture the general properties of the man-made structures. Observed data dependency is modelled by a pseudo-likelihood approximation. It yields better results compared with most other approaches.

We adopt a similar feature extraction procedure as in [5] but we replace the generative model approach with a discriminative GP model approach, to capture the dependencies between the block features by directly modelling the posterior over labels. Moreover, its kernel based non-parametric nature makes GP more flexible compared with parametric models.

The paper is structured as follows. GP Classification is introduced in Section 2 and a description of IVM is given in Section 3. In Section 4, experiment details and results are presented. Section 5 gives out the main conclusions of the work.

2 Gaussian Processes for classification

2.1 GP

1-4244-1180-7/07/$25.00 ©2007 IEEE

A GP is a collection of random variables, any finite number of which has a joint Gaussian distribution [6]. It is fully specified by its mean function m(x) and covariance function k(x, x') , expressed as:

$$f \sim GP(m,k) \qquad (2.1)$$

which defines a distribution over it covariance functions. The inference can be cast directly into the GP framework by learning a covariance function from training data.

2.2 GP regression

GP regression aims to recover the underlying process from the observed training data. Following the exposition in [7]: we have a dataset D with n observations $D = \{(x_i, y_i) \mid i = 1,...,n\}$, where x is the input vector of dimension d and y is the scalar output. Input data are put in d x n matrix X and the targets/output in vector y, $D = (X, y)$.

Typically, given noisy observations $D = (X, y)$ where $y = f + \varepsilon$ and additive noise $\varepsilon \sim \mathrm{N}(0, \sigma^2 I)$, the condition GP mean predictive distribution can be expressed as

$$\bar{f}_* = K(X_*, X)[K(X,X) + \sigma_n^2 I]^{-1} y \qquad (2.2)$$

where $K(X_*, X)$ denotes the covariance matrix of training and test points and $K(X,X)$ is the training data covariance.

The GP mean prediction in equation (2.2) can either be regarded as a linear combination of the observations y or the linear combination of kernel functions, each centred on a training point.

2.3 GP classification

In our application, we need a binary classifier to discriminate between man-made structure and non-structure so our dataset is $D = (X, y)$, where X are input training image features and y the class labels -1/+1. GP binary classification is done through a latent function. After calculating the distribution over latent function: the output of regression is 'squashed' through a sigmoid transformation to guarantee the valid probabilistic value within the range of [0,1]. Since class labels are discrete in binary classification the Gaussian likelihood is no longer valid, and so approximation is needed for calculation. EP approximation is generally used (see Algorithms (3.5) and (3.6) in [7]).

2.4 GP kernels

The GP kernel is the crucial part of GP learning, since it incorporates the prior smoothness assumption.

The typical covariance functions, studied in this paper, include [7]:

1) Radial Basis Function (RBF), also called as Squared Exponential (SE) function or Gaussian function

$$k_{RBF}(r) = \exp(-\frac{r^2}{2l^2}) \qquad (2.3)$$

where $r = x - x'$, x and x' are input pairs, l is the characteristic length-scale.

2) Matern class of covariance functions

$$k_{v=5/2}(r) = (1 + \frac{\sqrt{5}r}{l} + \frac{5r^2}{3l^2})\exp(-\frac{\sqrt{5}r}{l}) \qquad (2.4)$$

3) Linear kernel

$$k(x,x') = \sigma_0^2 + x \bullet x' \qquad (2.5)$$

where x and x' are input pairs.

3 Fast sparse Gaussian Process – the Informative Vector Machine (IVM)

IVM [8, 9] selects only a small subset of the dataset: the most informative d points out of the total N training points, thus reduce the computation complexity from $O(N^3)$ to $O(d^2 N)$. IVM greedily minimise the entropy of the posterior by including only the most informative data points that most reduce the entropy in a sequential manner. The selected d points form the so called 'active set' [3].

Following Eq (A.20) in [7], the entropy of a Gaussian $N(\mu, \Sigma)$ in D dimensions can be expressed as:

$$H[N(\mu,\Sigma)] = \tfrac{1}{2}\log|\Sigma| + \tfrac{D}{2}(\log 2\pi e) \qquad (3.1)$$

For the greedy algorithm deciding which points are taken into the active set I, Lawrence [3] proposed to choose the next point for inclusion into the active set being the one that maximizes the differential entropy score $H[Q_i^{new}] - H[Q_i]$, where Q_i is the Gaussian approximation of the posterior $p(f \mid X, y)$ at site i as described in Section 2.3, $H[Q_i]$ is the entropy at site i and $H[Q_i^{new}]$ being the entropy at site i once the observation at this site is included. By involving Eq (3.1), the differential entropy score can be written as:

$$H[Q_i^{new}] - H[Q_i] = \tfrac{1}{2}\log|\Sigma_i^{new}| - \tfrac{1}{2}\log|\Sigma_i|$$

$$= \frac{1}{2} \log \frac{\left|\Sigma_i^{new}\right|}{\left|\Sigma_i\right|} \quad (3.2)$$

Thus, it is proportional to the log ratio between the variances of the Q_i^{new} and Q_i. The change of the entropy after including a point is equivalent to the reduction in the level of uncertainty.

Choosing the inclusions (d of them) forces the resulting model to be sparse. Moreover, IVM uses an EP style approximation of the posterior and, as shown in Eq (3.51) in [7], the likelihood term can be ignored if its site values are very small. In this way a sparse model is obtained and computation efficiency is gained. Details of IVM implementation can be found in [3, 10].

4 Experiments and results

4.1 Orientogram features

A feature vector is computed at each 16×16 block.These features are designed to capture the lines and edges patterns in man-made structure [5] [11].

As described in [11] , a 14 component feature vector is generated at different scales: 1×1, 2×2, and 4×4 blocks. These features are derived from "orientograms": histograms of gradient orientations in a region weighted by gradient magnitudes. The 14 features include:

1) The first heaved central-shift moments (three scales)
2) The third heaved central-shift moments (three scales)
3) The absolute location of the highest bin (three scales)
4) The relationship of two most dominant orientations at the three scales expressed as

$$\text{rd\_intra} = \left|\sin(\delta_1 - \delta_2)\right| \quad (4.1)$$

where δ_1 and δ_2 are the two dominant orientations.

5) The relationship of the dominant orientations between adjacent scales, which is

$$\text{rd\_inter} = \left|\cos 2(\delta^i - \delta^{i+1}\right| \quad (4.2)$$

where δ^i and δ^{i+1} are dominant orientations at adjacent scales i and i+1.

We only keep the eight features in 1), 4) and 5) which cover more general properties of man-made structures.

4.2 Experiments and results

The proposed approach was trained and tested using the Corel images that Kumar [5] used[1]. To increase the variation and to test the generalization ability, we used some images, collected by the authors around our campus, for testing as well All images are cut to the size of 256x256 and divided into non-overlapping 16x16 pixels blocks which are labelled as one of the two classes, i.e. building or non-building blocks.

We used a training set of 11 Corel images, containing 407 structured blocks and 1768 non-structured blocks. Testing is implemented on 43 images, including 33 Corel images and 10 self-collected images. All test images do not appear in the training set.

For IVM GP classification, we run Lawrence's program [9] [2] . Rasmussen and Williams's GP classification program is applied for standard GP classification [7][3]. The inclusion training point number, d , is set to 660 in our test. This is a compromise between speed and performance considering the time complexity being proportional to d^2.

The RBF kernel is the most frequently used in applications of GP learning. Kernels allow for incorporation of prior knowledge, therefore it makes little sense to apply the same kernel to different applications. Thus we also investigated a variety of GP kernels, including RBF, RBF with ARD (Automatic Relevance Determination) [12], RBF with linear function, etc. The Rational Quadratic (RQ) and Neural Network (NN) functions were also tested: However these two functions yield less satisfactory results, and are not listed in the comparison figures.

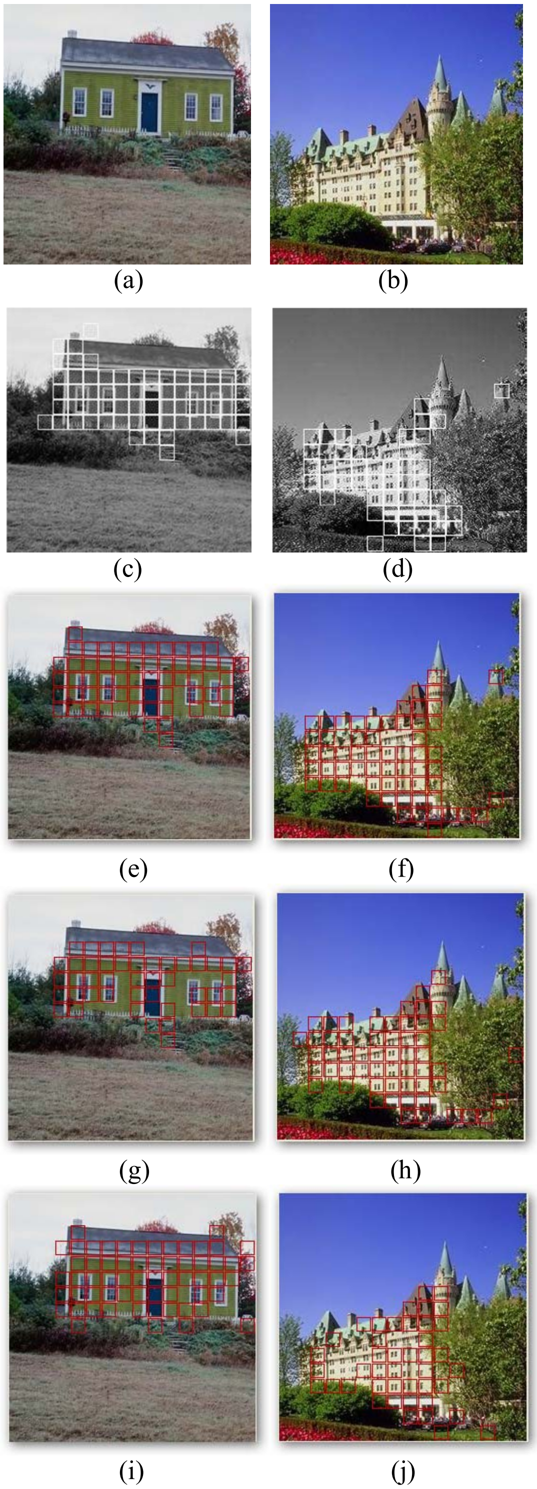
Figure 1 shows some of the test results for IVM GP classification (using the Matern kernel and RBF kernel respectively) and standard GP classification as well as Kumar's MSRF results. GP classification with Matern kernel tends to cover more building blocks and have less false detections.

Specifically, we have compared our results with that of Kumar's[5] on group of test images– Table 1. Despite using only 1/20 number of training data compared with his (and only 8 of the 14 feature types he used), the results on Corel images are almost equivalent to his results. Moreover, we do not impose spatial coherence in image space (unlike the MSRF of Kumar). The results on the 10 images added by the authors have a relatively lower detection rate and similar false positives which implies that our campus building may not well represented by the buildings in the Corel data set. Nevertheless, clearly there is significant generalisation to different architectural types. The overall results on all 43 test images: with a detection rate of 70.65%, the false positive rate is 1.49 block/image. One can increase the detection rate at the cost of more false positives: The false positives go up to 2.53 with a higher detection rate of 78.59%.

[1] http://www.cs.cmu.edu/~skumar/manMadeData.tar

[2] http://www.cs.man.ac.uk/~neill/ivm/downloadFiles/

[3] http://www.gaussianprocess.org/gpml/code/matlab/doc /classification.html


(a) (b)

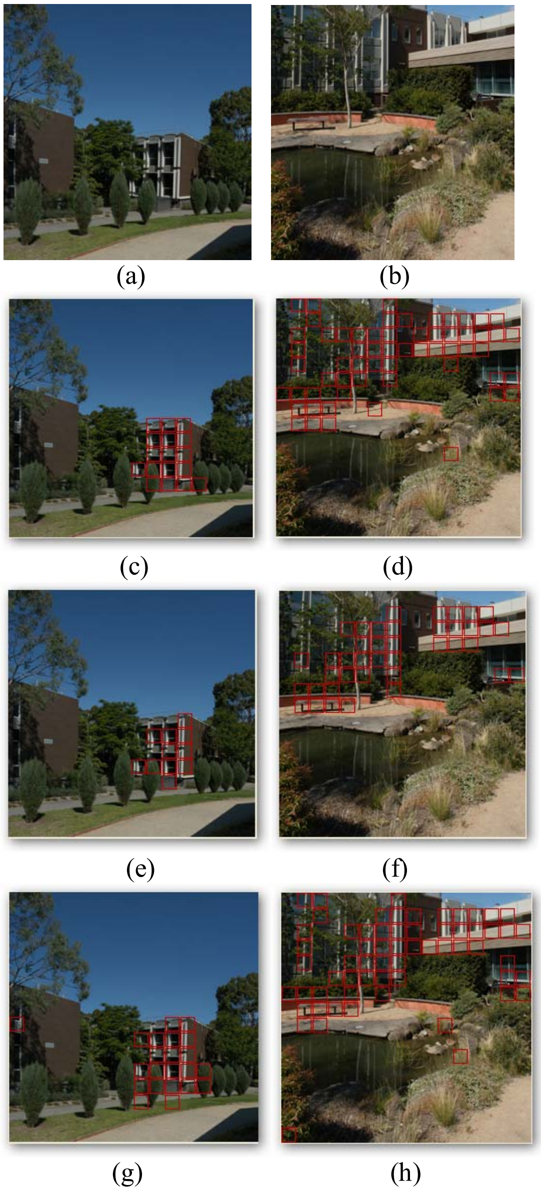
(c) (d)

(e) (f)

(g) (h)

(i) (j)

Figure 1. Classification results. (a)(b) original images (c)(d) Kumar’s results (e)(f) IVM Matern kernel results (g)(h) IVM RBF kernel results (i)(j) GP RBF kernel results.


(a) (b)

(c) (d)

(e) (f)

(g) (h)

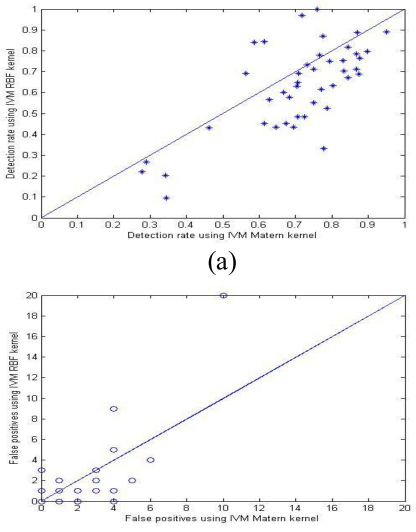
Figure 2. Classification results. (a)(b) original images (c)(d) IVM Matern kernel results (e)(f) IVM RBF kernel results (g)(h) GP RBF kernel results.

In Figure 2, results on our campus images again shows that IVM GP classification with Matern kernel is better although no campus images have been included in our training set yet.

We focus on a comparison between Matern kernel and RBF kernel, since these have the best performance, compared with other kernels, in our application. The Matern kernel with IVM is compared with RBF with IVM (shown in Figure 3). The RBF kernel used in a standard GP is compared to the IVM Matern (shown in Figure 4). Results in Figure 3 shows a clear advantage of Matern kernel over RBF kernel on detection rate (and a similar rate of false positives). Compared with the RBF kernel, the Matern kernel with IVM has a similar detection rate with less false positives as shown in Figure 4. In Figure 5 (a), the Matern kernel is compared with several kernels in an IVM implementation. It has better performance in that either the false positives are low under similar detection rate or the detection rate is higher with similar false positives. In Figure 5 (b), performance is compared on the Corel images only. Overall, the Matern kernel seems to yield the best performance.

Tests have also been done in extending the IVM inclusion points from 660 to 1060 as well as enlarging the training data sets up to 8000 points. Results are all similar as to that of 2000 training points with 660 included. This implies that the IVM approach is not only efficient in terms of computation time but also can capture the information well with limited inclusion points.

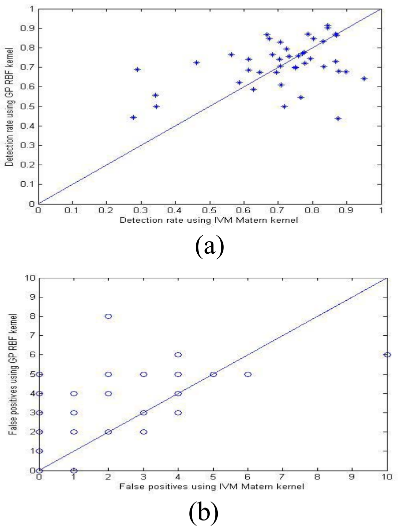
Results in Figure 6 are obtained from the computer with Intel 1.66GHz+980MHz CPU. It can be seen that the computational time of GP increases drastically with the growth in the number of training data points. In case of 8000 training points, the standard GP is almost prohibitive. IVM times are consistent with $O(N \cdot d^2)$.



(a)

(b)

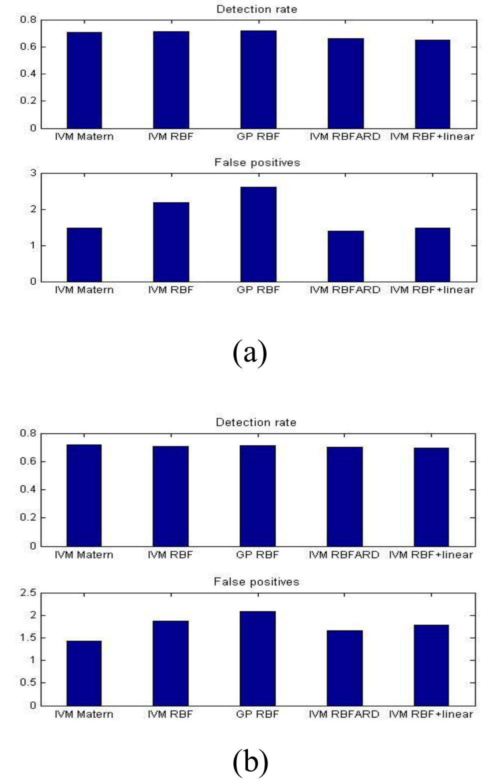
Figure 3. Comparison of IVM Matern kernel and IVM RBF kernel on test data. (a) Detection rate of IVM Matern vs IVM RBF. (b) False positives of IVM Matern vs IVM RBF.



(a)

(b)

Figure 4. Comparison of IVM Matern kernel and GP RBF kernel on test data. (a) Detection rate of IVM Matern vs GP RBF. (b) False positives of IVM Matern vs GP RBF.



(a)

(b)

Figure 5. (a) Detection rate and false positives comparison of different kernels on all test images. (b) Detection rate and false positives comparison of different kernels on Corel test images only.

5 Conclusions

We have described the application of IVM (which is an efficient sparse approximation of GP classification) to man-made structure classification. With IVM GP classification, performance is maintained with only a fraction of the training data. Moreover, since this affords experimental kernel tuning, the resulting structure can be more accurately trained. Future work will involve the investigation of active data selection (seeking parts of the images to improve the classification in regions where the GP indicates most uncertainty, and asking the user to verify the classification, for example) for semi-supervised learning and other facets that will facilitate on-line learning of building detection in image data.

6 References

[1] L. Csato and M. Opper, "Sparse representation for Gaussian process models," *Advances in Neural Information Processing Systems,* vol. 13, pp. 444-450, 2001.

[2] E. Snelson and Z. Ghahramani, "Sparse Gaussian Processes using Pseudo-inputs," in *Neural Information Processing Systems*, 2005.

[3] N. Lawrence, M. Seeger, and R. Herbrich, "Fast Sparse Gaussian Process Methods: The Informative Vector Machine," *Advances in Neural Information Processing Systems,* 2003.

[4] X. Feng, C. K. I. Williams, and S. N. Felderhof, "Combining Belief Networks and Neural Networks for Scene Segmentation," *IEEE Transaction on Pattern Analysis and Machine Intelligence,* vol. 24, pp. 467-483, 2002.

[5] S. Kumar and M. Hebert, "Man-Made Structure Detection in Natural Images using a Causal Multiscale Random Field," in *CVPR2003*, 2003, p. 119.

[6] C. E. Rasmussen, "Gaussian Processes in Machine Learning," *Advanced Lectures on Machine Learning,* 2003.

[7] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*: The MIT Press, 2006.

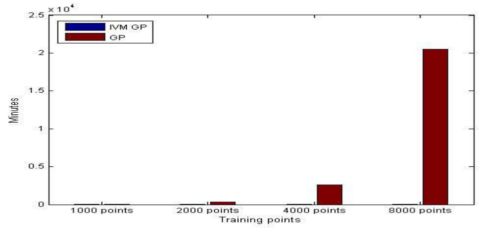
[8] N. D. Lawrence and J. C. Platt, "Learning to Learn with the Informative Vector Machine," in *International Conference on Machine Learning*, 2004.

[9] N. D. Lawrence, J. C. Platt, and M. I. Jordan, "Extensions of the Informative Vector Machine," in *Deterministic and Statistical Methods in Machine Learning*, 2004.

[10] M. Seeger, "Bayesian Gaussian Process Models: PAC-Bayesian Generalisation Error Bounds and Sparse Approximations," in *Institute for Adaptive and Neural Computation, Division of Informatics.* vol. Doctor of Philosophy Edinburgh: University of Edinburgh, 2003.

[11] C. Pantofaru, R. Unnikrishnan, and M. Hebert, "Toward Generating Labeled Maps from Color and Range Data for Robot Navigation," in *2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2003.

[12] R. M. Neal, *Bayesian Learning for Neural Networks*: Springer-Verlag New York, Inc., 1996.



	1000 pts	2000pts	4000pts	8000pts
IVM GP	8min	14min	22min	35min
GP	40min	320min	2560min	20480min*

Figure 6. Comparison of computational time between IVM GP and GP.
* Estimation only.

	Kumar's MSRF model	Our IVM GP model with Matern kernel		
Training set	Corel images	Corel images		
Training data scale	108 images (3004 structured blocks + 36269 non-structured blocks)	11 Corel images with 407 structured blocks and 1768 non-structured blocks		
Testing set	Corel images	Images from Corel Photo Stock as well as varied pictures collected by the authors in the campus		
Data scale	129 images	43 images (33 Corel images + 10 random collected images)		
Detection rate	72.13%	33 Corel images	10 collected images	All 43 images
		71.69%	61.11%	70.65%/78.59%
False positives	1.46	33 Corel images	10 collected images	All 43 images
		1.42	1.54	1.49/2.53

Table 1. Comparison of our IVM GP approach with Kumar's MSRF model.