Robust Classification and Semi-Supervised Object Localization with Gaussian Processes

Alexander Lütz

Chair for Computer Vision
Friedrich Schiller University of Jena
Alexander.Luetz@uni-jena.de
http://www.inf-cv.uni-jena.de

Abstract. Traditionally, object recognition systems are trained with images that may contain a large amount of background clutter. One way to train the classifier more robustly is to limit training images to their object regions. For this purpose we present a semi-supervised approach that determines object regions in a completely automatic manner and only requires global labels of training images. We formulate the problem as a kernel hyperparameter optimization task and utilize the Gaussian process framework. To perform the computations efficiently we present techniques reducing the necessary time effort from cubically to quadratically for essential parts of the computations. The presented approach is evaluated and compared on two well-known and publicly available datasets showing the benefit of our approach.

1 Introduction and Related Work

Image categorization became a well studied problem in the area of image understanding during the last years. Traditionally, one represents already labeled training images by certain features and trains a classifier based on features and labels. In a second step labels of unknown images can be estimated by evaluating the response of the classifier for each image. The main assumption is the presence of only one single dominant object per training image with only few clutter and occlusion. Otherwise, the extracted features would not be representative for the category given by the image label. Going one step further, researchers attempted to overcome this limitation by using more complex classifiers [11] or by extracting a large set of features [12, 3]. Nevertheless, this leads to higher computation times as well as higher memory demand in many cases. For this reason, we introduce a new method to determine object regions in training images only given the category label. Therefore, we interprete the object region in an image as a kernel function hyperparameter and optimize the model likelihood with respect to these hyperparameters. This allows obtaining convenient training images for a robust training of a classification system. To reduce the computational effort we apply two lemmata that allow computing inverse and determinant of a matrix in quadratically time in contrast to cubically effort with standard approaches.

recommended for submission to YRF2011 by Prof. Dr.-Ing. Joachim Denzler

Many publications directly deal with the detection or localization of objects in images [5, 8]. Many of these approaches use sliding window techniques to collect hundreds of possible object regions, classify each region and return the one classified with lowest uncertainty or best score. Obviously, this is not possible, if the classifier was trained on images rather than on regions. An alternative are generic object detectors, as proposed by Alexe et al. [1]. They perform detection of arbitrary objects by defining object cues for the presence of an object — like strong color contrast or high edge density.

To our knowledge, just a few publications directly address the determination of object regions in training images by using class labels only. Chum et al. [4] select the region in an image which achieves the highest similarity score to all other images of its class, measured by similarities of visual words and edge densities. Bosch et al. [2] present a method similar to [4] that also obtains object regions in images by maximizing a similarity score, but evaluates the similarity function only on a subset of the training images, instead of considering every training example. In contrast to these approaches, we select the image region, which gives highest probability to explain the class labels by considering only the part of the image covered by the region.

The remainder of the paper is organized as follows. In Sect. 2 we will briefly review classification with Gaussian processes, present our approach for object localization with hyperparameter optimization and show techniques for efficient computations. Experimental results are given in Sect. 3 that show the benefit of our approach. A summary of our findings and a discussion of future research directions conclude the paper.

2 Object Localization with Hyperparameter Optimization in a Gaussian Processes Framework

Brief review of Gaussian Process Classification Assume a given set of training images $\left(\mathcal{I}_1, \ldots, \mathcal{I}_n\right)$ represented by certain features $\boldsymbol{X} = \left(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\right)$ and a vector $\boldsymbol{t}_L \in \{-1, 1\}^n$ containing the labels of the images. Then we are interested in estimating the general relationship between unseen examples $\boldsymbol{x}_* \in \mathcal{X}$ and their class labels t^*. If we use a kernel function $\kappa : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ that maps each pair of features to a similarity score we can model the relation in a probabilistic way using Gaussian processes (GP) [11]. The main assumption is that every label t_i is created by a continous latent variable y_i. Then every two labels y_i, y_j are expected to be jointly Gaussian and their covariance is specified by applying the kernel function $\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)$ to their inputs. As in [11] we assume the y_i to have a zero mean, which leads to $\mathrm{P}(\boldsymbol{y}|\boldsymbol{X}) \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{K})$ with $\boldsymbol{K}_{i,j} = \kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)$. The choise of κ is crucial for the performance of the classification system, because it defines how strong the estimated label differs given a change in the feature vector. Therefore, to adjust the chosen kernel function to the training data one possibility is to use a parameterized kernel function and to optimize its hyperparameters with respect to the training data. In the Gaussian process framework, optimization can be done by maximizing

the model likelihood $\mathrm{P}\big(\boldsymbol{t}_\mathrm{L}|\boldsymbol{X},\boldsymbol{\beta}\big)$, which states how well the class labels can be explained given the training data under the chosen model.

Object localization with hyperparameter optimization If the object region in an image is interpreted as a hyperparameter of the kernel function, object localization becomes equivalent to optimization of hyperparameters. Let $\boldsymbol{\beta} = \big(\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_n\big)$ be the vector of hyperparameters with $\boldsymbol{\beta}_i$ as a representation of the object region for the ith image, such as upper left and lower right corner of a rectangle. Then the determination of the object regions can be done by

$$\boldsymbol{\beta}^* = \underset{\boldsymbol{\beta}}{\operatorname{argmax}}\, \mathrm{P}\big(\boldsymbol{t}_\mathrm{L} \mid \boldsymbol{X}, \boldsymbol{\beta}\big). \tag{1}$$

If we expect only additional Gaussian noise in the labels, the logarithmic likelihood in the GP regression framework can be written in closed form [14]

$$\log \mathrm{P}\big(\boldsymbol{t}_L \mid \boldsymbol{X}, \boldsymbol{\beta}\big) = -\frac{1}{2}\log\det(\boldsymbol{K}_{\boldsymbol{\beta}} + \sigma^2\boldsymbol{I}) - \frac{1}{2}\boldsymbol{t}_L^T\big(\boldsymbol{K}_{\boldsymbol{\beta}} + \sigma^2\boldsymbol{I}\big)^{-1}\boldsymbol{t}_L + \text{const}\ . \tag{2}$$

In (2), $\boldsymbol{K}_{\boldsymbol{\beta}}$ denotes the GP covariance matrix computed with the parameterized kernel function, which in our case is equal to restricting the training images to the regions specified by $\boldsymbol{\beta}$.

If we have a multi-class classification task that is $\boldsymbol{t}_L \in \{1, \ldots, m\}^n$, m one-vs-all-classifiers can be used. Assuming independent outputs of the m classifiers, we can again compute the joint likelihood [11]

$$\log \mathrm{P}\big(\boldsymbol{t}_L \mid \boldsymbol{X}, \boldsymbol{\beta}\big) = \sum_{j=1}^{m} \log \mathrm{P}\big(\boldsymbol{t}_L^{(j)} \mid \boldsymbol{X}, \boldsymbol{\beta}\big), \tag{3}$$

with binary label vectors $\boldsymbol{t}_L^{(j)}$ whose entries are equal to one if the corresponding entry of $\boldsymbol{t}_L$ is j and -1 otherwise.

To perform the optimization of (2) and (3) one typically uses non-linear optimization techniques like gradient descent. Caused by the descrete parameter space this is not possible in our case. Therefore and due to the combinatorial complexity, we use a greedy strategy as an approximation. In detail, we fix every dimension of $\boldsymbol{\beta}$ except one and perform likelihood optimization according to this dimension. This is done for every dimension and repeated for several times, which is known as cyclic coordinate search [13]. In practice this corresponds to fixing every image region except for one and choosing the region for this specific image that maximizes the likelihood with respect to the already computed regions of all other images.

Methods for efficient computations To reduce the computational effort we draw advantage of our greedy approximation scheme. While performing the optimization of one single dimension, the resulting kernel matrix changes only in one row and one column. This is equal to a rank-2-update of $\boldsymbol{K}$. Therefore we can apply Woodbury's formula [9] to compute the inverse of the slightly changed covariance matrix $\boldsymbol{K}'$ by utilizing the already computed inverse of $\boldsymbol{K}$.

Table 1. Recognition rates averaged over all categories of Caltech-101. Entry x/y denotes restricting training images to x and test images to y

# training examples per category	global / global	ROI est. / GT ROI	GT ROI / GT ROI
5	39.12	40.63	49.11
10	44.89	45.55	55.17
15	48.76	49.42	58.59

This results in a computational effort from only $\mathcal{O}(n^2)$ compared to $\mathcal{O}(n^3)$ with standard approaches like Cholesky decomposition. With our implementation, this leads to a time effort of just 0.04 s for inverting a 2000×2000 -Matrix on a standard PC in contrast to 12.04 s with a complete Cholesky decomposition. Apart from that, we also benefit from using the determinant lemma (see chapter 18 of [10]). With the Schur-Complement of $\boldsymbol{K}$ on hand — which we already needed for the efficient determination of the inverse — we are able to compute the determinant in constant time for rank-2-modifications of $\boldsymbol{K}$.

3 Experimental Results

To demonstrate the benefit of our approach, we performed experiments on Caltech-101 [7] and Pascal VOC 2008 [6]. We extracted PHOG-features [2] and BoF-features (identical setup as presented in [15]) for every image to use both structure and color information. The results were combined with uniform weights. As supposed in [11] we also tested weight optimization but this decreased the results slightly. We want to point out that we did not focus on choosing the most promising features or optimize their extraction. To generate region hypotheses for the greedy optimization scheme we performed a sliding window approach. Therefore, we scaled the initial image region by a factor ranging from 1.0 to 0.6 with step size of 0.1. To perform the optimization of (2) or (3) in Sect. 2, we initialized the bounding boxes with the whole image regions and repeated the iterations over all training images for 10 times. For the multi-class classification task we measured recognition rates averaged over all classes whereas we chose the average precision measure for the binary case.

Evaluation Although Caltech-101 is not the most convenient dataset for evaluating the performance of an object localization system, it is one of the standard datasets for classification tasks. Therefore, we present the results achieved with our approach on this dataset.

As we can see in Table 1, our approach improves the quality of the training step slightly, although there is still some space left for improvement compared to the results based on ground truth regions for training. This is due to the fact, that many images of Caltech-101 show only one dominant object. Nevertheless, the automatically determined object regions are visually meaningful as shown in Fig. 1.

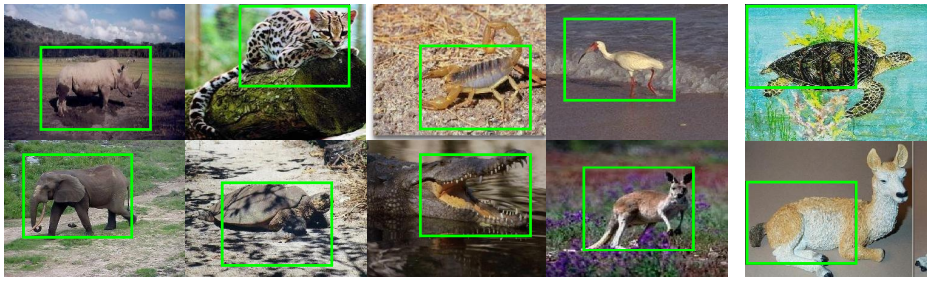

Fig. 1. Good (left) and bad (right) results achieved with our approach on Caltech-101 with 15 training images per category (best viewed in color)

Classifying images from Pascal VOC 2008 is a more challenging task. On this difficult dataset our method showed superior performance compared to the standard approach, which can clearly be seen in Table 2. Although the results obtained with our approach are a little lower than the ground truth results, the improvement is up to a factor of six for our simple feature set. This clearly points out the advantages of our approach for a robust training especially in difficult classification tasks. The results confirm the fact that images restricted to their object regions give an essential benefit for building classifiers more robustly. Fig. 2 shows some exemplary results on Pascal VOC 2008 bicycle achieved by our approach. Note that the bad examples are cases where the bicycle regions are too small compared to the minimum scaling factor or are not highly representative for the bicycle category.

4 Conclusion and Future Work

We have shown that reducing images to their object regions allows building classifiers more robustly. Our approach showed superior performance by improving classification results up to a factor of six for challenging tasks compared to classification based on whole images. To overcome computational limitations we proposed techniques for efficient computations. As future work we plan to replace the sliding window approach with a generic object detector to reduce both computation time and probability of choosing non-meaningful image regions. It could also be interesting to evaluate the utility of our approach in an active learning setup. Apart from this, we want to use our approach to localize multiple objects per image in the test step.

Table 2. Average precision rates achieved on Pascal VOC 2008 bicycle

# training examples per category	global / global	ROI est. / GT ROI	GT ROI / GT ROI
15	6.13	11.63	56.67
30	8.46	35.74	55.03
50	7.48	42.84	57.28

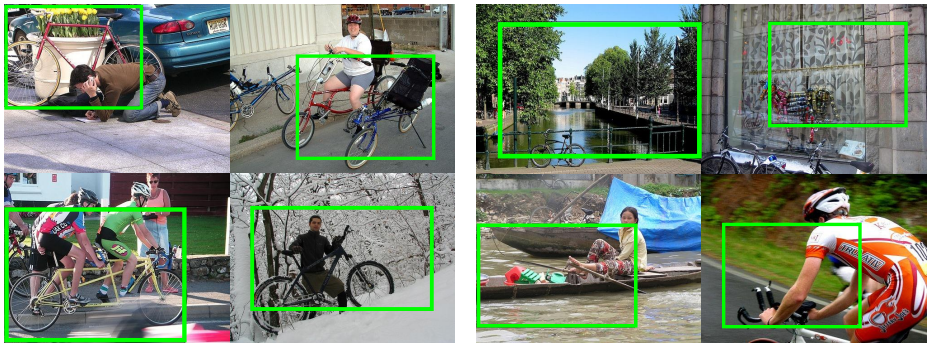

Fig. 2. Good (left) and bad (right) results achieved with our approach on Pascal VOC 2008 bicycles with 50 training images per category (best viewed in color)

Acknowledgements I am grateful for the support of my advisor Erik Rodner.

References

1. Alexe, B., Deselaers, T., Ferrari, V.: What is an object? In: Proceedings of the CVPR. pp. 73–80 (2010)
2. Bosch, A., Zisserman, A., Munoz, X.: Image classification using random forests and ferns. In: Proceedings of the ICCV. pp. 1–8 (2007)
3. Bosch, A., Zisserman, A., Munoz, X.: Representing shape with a spatial pyramid kernel. In: Proceedings of the CIVR. pp. 401–408 (2007)
4. Chum, O., Zisserman, A.: An exemplar model for learning object classes. In: Proceedings of the CVPR (2007)
5. Dalal, N., Triggs, B.: Histogram of oriented gradients for human detection. In: Proceedings of the CVPR. pp. 886–893 (2005)
6. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes (VOC) challenge. IJCV 88, 303–338 (2010)
7. Fei-Fei, L., Fergus, R., Perona, P.: Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In: Workshop on Generative-Model Based Vision (2005)
8. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part based models. PAMI 32 (2010)
9. Hager, W.W.: Updating the inverse of a matrix. Society for Industrial and Applied Mathematics (SIAM) Review 31(2), 221–239 (1989)
10. Harville, D.A.: Matrix Algebra From a Statistician's Perspective. Springer (2007)
11. Kapoor, A., Grauman, K., Urtasun, R., Darrell, T.: Gaussian processes for object categorization. IJCV 88, 169–188 (2010)
12. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: Proceedings of the CVPR. pp. 2169–2178 (2006)
13. Nocedal, J., Wright, S.J.: Numerical Optimization. Springer (1999)
14. Rasmussen, C.E., Williams, C.K.I.: Gaussian Processes for Machine Learning. Adaptive Computation and Machine Learning, The MIT Press (2006)
15. Rodner, E., Denzler, J.: One-shot learning of object categories using dependent gaussian processes. In: Proceedings of the DAGM. pp. 232–241. Springer (2010)