A Multiple Kernel Learning Approach to Joint Multi-class Object Detection

Christoph H. Lampert and Matthew B. Blaschko

Max Planck Institute for Biological Cybernetics,
Department for Empirical Inference
72076 Tübingen, Germany
{chl,blaschko}@tuebingen.mpg.de

Abstract. Most current methods for multi-class object classification and localization work as independent 1-vs-rest classifiers. They decide whether and where an object is visible in an image purely on a per-class basis. Joint learning of more than one object class would generally be preferable, since this would allow the use of contextual information such as co-occurrence between classes. However, this approach is usually not employed because of its computational cost.

In this paper we propose a method to combine the efficiency of single class localization with a subsequent decision process that works jointly for all given object classes. By following a multiple kernel learning (MKL) approach, we automatically obtain a sparse dependency graph of relevant object classes on which to base the decision. Experiments on the PASCAL VOC 2006 and 2007 datasets show that the subsequent joint decision step clearly improves the accuracy compared to single class detection.

1 Introduction

Object detection in natural images is inherently a multi-class problem. Already in 1987, Biederman estimated that humans distinguish between at least 30,000 visual object categories [3]. Even earlier, he showed that the natural arrangement and co-occurrence of objects in scenes strongly influences how easy it is to detect objects [4]. Recently, Torralba and Olivia obtained similar results for automatic systems [27]. However, most algorithms that are currently developed for object detection predict the location of each object class independently from all others. The main reason for this is that it allows the algorithms to scale only linearly in the number of classes. By disregarding other object classes in their decision, such systems are not able to make use of dependencies between object classes. Dependencies are typically caused by functional relations between objects (an image showing a computer keyboard has a high chance of also showing a computer screen), or by location and size (a mosquito would not be visible in an image of an elephant).

In this paper, we propose a method that automatically makes use of dependencies between objects and their background as well as between different object

G. Rigoll (Ed.): DAGM 2008, LNCS 5096, pp. 31–40, 2008.

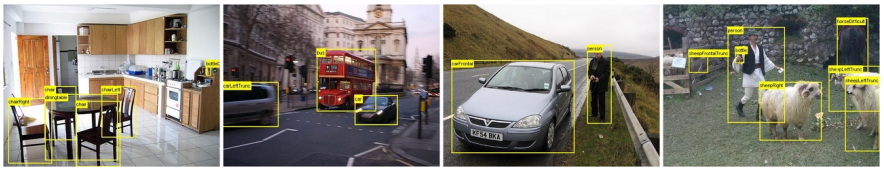

Fig. 1. Example images from the PASCAL VOC 2007 dataset [8]. Objects are marked by their bounding boxes. Some object classes like chairs and tables or cars and buses tend to occur together.

classes. It relies on first performing an overcomplete per-class detection, followed by a post-processing step on the resulting set of candidate regions. All necessary parameters, in particular the relation between the categories, are learned automatically from training data using a multiple kernel learning procedure. As a result, we obtain a sparse dependency graph of classes that are relevant to each other. At test time only these relevant classes are considered, making the algorithm efficiently applicable for problems with many object categories.

2 Related Work

Early approaches to object localization were mainly targeted at the detection of frontal faces and of pedestrians in street scenes. The influential work by Viola and Jones [28] might be the most well known publication in this area. Viola and Jones propose to detect faces by applying a cascade of weak classifiers at every location of the image. Regions that do not look face-like are rejected early on, whereas promising regions are kept until a final decision is made. The authors also mention the possibility of using a different classifier as last element of the cascade, which then acts as a strong post-filter of the cascade's output. Such two-step procedures, in which a first stage predicts candidate regions and a second stage accepts or rejects them, have been used frequently, especially when real-time performance is required. Variants include the of use of artificial neural networks [22], linear SVMs [7], reduced set SVMs [14], tree-structures instead of linear cascades [17], or fusion of different data modalities [19].

Recently, more methods to also detect multiple and more general object classes have been developed. In this area, sliding window approaches of single layer quality functions are more popular than hierarchical cascades to generate the regions of interest [11,15]. Alternatively, the *Implicit Shape Model* has been used [12], or heuristic techniques based on keypoint voting [5,6]. There is also a variety of methods to estimate the location and pose of object by probabilistic or geometric part models [2,10,13,18,24]. However, all these methods have in common that they only consider one object class at a time and cannot make use of class dependencies.

Attempts to take context into account using probabilistic appearance models, *e.g.* by Torralba [25], were restricted to object–background interaction and do not

capture relations between objects. Aiming at simultaneous multi-class detection, Torralba et al. [26] have proposed to share features between different classes, but this applies only to the class representations and does not allow one to base the final decision on between-class dependencies.

To our knowledge, the only published work making use of inter-class dependencies to improve object detection is by Rabinovich et al. [20]. Their method segments the image and classifies all segments jointly based on a conditional random field. However, this requires the object dependencies to be specified *a-priori*, whereas our method learns the dependencies during the training process to best reflect the *a-posteriori* beliefs.

3 Joint Multi-class Object Detection

The proposed method for joint object detection is applicable as a post-processing operation to any of the single-class methods mentioned in the previous section. We will therefore concentrate on this aspect and assume that routines to identify candidate locations for K object classes $\omega_1, \ldots, \omega_K$ are given. We do not, however, assume that these routines are able to reliably judge if an object is present at all or not, so in theory, even random sampling of locations or an exhaustive search would be possible.

For all candidate regions, it is predicted whether they are correct or incorrect hypotheses for the presence of their particular object class. In this way we reduce the problem to a collection of binary classifications, but in contrast to existing detection methods, the decision is based jointly on all object hypotheses in the image, not only on each of them separately.

Following a machine learning approach, the system learns its parameters from a set of training images I^i, $i = 1, \ldots, N$, with known locations $l^i_1, \ldots, l^i_{n_i}$ and class labels for the n_i objects present in I^i. For simplicity, we assume that there is exactly one candidate region per class per image, writing $x^i := (I^i, l^i_1, \ldots, l^i_K)$ for $i = 1, \ldots, N$. This is not a significant restriction, since for missing classes we can insert random or empty regions, and for classes with more than one candidate region, we can create multiple training examples, one per object instance.

For every test image I we first predict class hypotheses and then, if necessary, we use the same construction as above to bring the data into the form $x = (I, l_1, \ldots, l_K)$. The class decisions are given by a vector valued function

$$\boldsymbol{f} : \mathcal{I} \times \overbrace{\mathcal{L} \times \cdots \times \mathcal{L}}^{K \text{ times}} \to \mathbb{R}^K \tag{1}$$

where $\mathcal{I}$ denotes the space of images and $\mathcal{L}$ denotes the set of representations for objects, *e.g.* by their location and appearance. Each component f_k of $\boldsymbol{f}$ corresponds to a score how confident we are that the object l_k is a correct detection of an object of class ω_k. If a binary decision is required, we use only the sign of f_k.

3.1 Discriminative Linear Model

In practical applications, the training set will rarely be larger than a couple of hundred or a few thousand examples. This is a relatively low number taking into account that the space $\mathcal{I} \times \mathcal{L}^K$ grows exponentially with the number of classes. We therefore make use of a discriminative approach to classification, which has shown to be robust against the curse of dimensionality. Following the path of statistical learning theory, we assume $\boldsymbol{f}$ to be a vector-valued function that is linear in a high-dimensional feature space $\mathcal{H}$. Using the common notation of reproducing kernel Hilbert spaces, see *e.g.* Schölkopf and Smola [21], each component function f_k of $\boldsymbol{f}$ can be written as

$$f_k(x) \;=\; \langle\, w_k, \phi_k(x) \,\rangle_{\mathcal{H}} + b_k \tag{2}$$

where $x = (I, l_1, \ldots, l_K)$. The feature map $\phi_k : \mathcal{I} \times \mathcal{L}^K \to \mathcal{H}$ is defined implicitly by the relation $k_k(x, x') = \langle \phi_k(x), \phi_k(x') \rangle_{\mathcal{H}}$ for a positive definite kernel function k_k. The projection directions $w_k \in \mathcal{H}$ and the bias terms $b_k \in \mathbb{R}$ parametrize $\boldsymbol{f}$. Note that we do not compromise our objective of learning a joint decision for all classes by writing separate equations for the components of $\boldsymbol{f}$, because each f_k is still defined over the full input space $I \times \mathcal{L}^K$.

3.2 Learning Class Dependencies

In an ordinary SVM, only w_k and b_k are learned from training data whereas the kernels k_k and thereby the feature maps ϕ_k are fixed. In our setup, this approach has the drawback that the relative importance of one class for another must be fixed *a priori* in order to encode it in k_k. Instead, we follow the more flexible approach of multiple kernel learning (MKL) as developed by Lanckriet et al. [16] and generalize it to vector valued output. MKL allows us to *learn the relative importance* of every object class for the decision of every other class from the data. For this, we parametrize $k_k = \sum_{j=0}^{K} \beta_k^j \kappa_j$, where the κ_j are fixed base kernels. The weights β_k^j are learned together with the other parameters during the training phase. They have characteristics of probability distributions if we constrain then by $\beta_k^j \in [0, 1]$ and $\sum_j \beta_k^j = 1$ for all k.

We assume that each base kernels κ_j reflects similarity with respect only to the object class ω_j, and that κ_0 is a similarity measure on the full image level. Note, however, that this is only a semantic choice that allows us to directly read off class dependencies. For the MKL training procedure, the choice of base kernels and also their number is arbitrary.

Because the coefficients β_k^j are learned from training data, they correspond to *a-posteriori* estimates of the conditional dependencies between object classes: the larger the value of β_k^j the more the decision for a candidate region of class ω_k depends on the region for class ω_j, where the dependency can be excitatory or inhibitory. In contrast, $\beta_k^j = 0$ will render the decision function for class ω_k independent of class ω_j. This interpretation shows that the joint-learning approach is a true generalization of image based classifiers (setting $\beta_k^0 = 1$ and $\beta_k^j = 0$ for $j \neq 0$) and of single class object detectors ($\beta_k^j = \delta_{jk}$).

3.3 Vector Valued Multiple Kernel Learning

To learn the parameters of $\boldsymbol{f}$ we apply a maximum-margin training procedure. As usual for SVMs, we formulate the criterion of maximizing the soft margin for all training examples with slack variables ξ_k^i as the minimization over the norm of the projection vector. Consequently, we have to *minimize*

$$\frac{1}{2}\sum_{k=1}^{K}\left(\sum_{j=0}^{K}\beta_j\|w_k^j\|_{\mathcal{H}_k}^2 + C\sum_{i=1}^{n}\xi_k^i\right) \tag{3}$$

with respect to $w_k^j \in \mathcal{H}_k, b_k \in \mathbb{R}, \beta_k^j \in [0,1]$ and $\xi_k^i \in \mathbb{R}^+$, subject to

$$y_k^i\left(\sum_{j=1}^{K}\beta_j^k\langle\, w_j^k, \phi_k(x^i)\,\rangle_{\mathcal{H}_k} + b_j\right) \geq 1 - \xi_k^i \quad \text{for } i = 1\ldots, N,\ k = 1,\ldots,K,$$

where the training labels $y_k^i \in \{\pm 1\}$ indicate whether the training location l_k^i in image I^i did in fact contain an object of class ω_k, or whether it was added artificially. The space $\mathcal{H}_j$ with scalar product $\langle .,. \rangle_{\mathcal{H}_j}$ is implicitly defined by κ_j, and C is the usual slack penalization constant for soft-margin SVMs. Because the constraints for different k do not influence each other, we can decompose the problem into K optimization problems. Each of these is convex, as Zien and Ong [30] have shown. We can therefore solve (3) by applying the multiple kernel learning algorithm K times. The result of the training procedure are classifiers

$$f_k(x) = \sum_{i=1}^{N}\sum_{j=0}^{K}\alpha_k^i\beta_j^k\kappa_j(x,\, x^i) + b_k \quad \text{for } k = 1,\ldots,K, \tag{4}$$

where the coefficients α_k^i are Lagrangian multiplier that occur when dualizing Equation (3). Because the coefficients α_k^i and β_k^j are penalized by L^1-norms in the optimization step, they typically become sparse. Thus, in practice most of the $N \cdot K$ terms in Equation (4) are zero and need not be calculated.

4 Experiments

For experimental evaluation we use the recent PASCAL VOC 2006 and VOC 2007 image datasets [8,9]. They contain multiple objects per image from sets of classes that we can expect to be inherently correlated, *e.g.* tables/chairs and cars/buses, or anti-correlated, *e.g.* cats/airplanes. Some examples are shown in Figure 1. In VOC 2006, there are 5,304 images with 9,507 labeled objects from 10 classes. VOC 2007 contains 9,963 images, with a total of 24,640 objects from 20 different classes. Both datasets have pre-defined train/val/test splits and ground truth in which objects are represented by their bounding boxes.

4.1 Image Representation

We process the images following the well established *bag-of-features* processing chain. At first, we extract local SURF descriptors [1] at interest point locations as well as on a regular image grid. On average, this results in 16,000 local descriptors per image. We cluster a random subset of 50,000 descriptors using K-means to build a codebook of 3,000 entries. For every descriptor only its x, y position in the image and the cluster ID of its nearest neighbor codebook entry are stored.

To represent a full image, we calculate the histogram of cluster IDs of all feature points it contains. Similarly, we represent a region within an image by the histogram of feature points within the region. These global or local histograms are the underlying data representation for the generation of candidate regions as well as for the class-decision step.

The joint decision function takes a set of hypothesized object regions as input. To generate these for our experiments, we use a linear SVM approach similar to Lampert et al. [15]: one SVM per object class is trained using the ground truth object regions in the training set as positive training examples and randomly sampled image boxes as negative training examples. The resulting classifier functions are evaluated over all rectangular regions in the images, and for each image and class, the region of maximal value is used as hypothesis.

4.2 Base Kernels and Multiple Kernel Learning

In the area of object classification, the χ^2-distance has proved to be a powerful measure of similarity between bag-of-features histograms, see *e.g.* [29]. We use χ^2 base kernels in the following way: for a sample $x = (I, l_1, \ldots, l_K)$, let $h_0(x)$ be the cluster histogram of the image I and let $h_j(x)$ be the histogram for the regions l_j within I. We set

$$\kappa_k(x, x') = \exp\left(-\tfrac{1}{2\gamma_k}\chi^2(h_k(x), h_k(x'))\right) \quad \text{with} \quad \chi^2(h, h') = \sum\nolimits_{c=1}^{3000} \frac{(h^c - h'^c)^2}{h^c + h'^c},$$

where h^c denotes the c-th component of a histogram h. The normalization constants γ_k are set to the mean of the corresponding χ^2-distances between all training pairs. With these kernels, we perform MKL training using the SHOGUN toolbox. It allows efficient training up to tens of thousands of examples and tens of kernels [23]. At test time, $\boldsymbol{f}$ is applied to each test sample $x = (I, l_1, \ldots, l_K)$, and every candidate region l_k is assigned $f_k(x)$ as a confidence score.

5 Results

The VOC 2006 and VOC 2007 datasets provide software to evaluate localization performance as a ranking task. For each class, precision and recall are calculated as follows: at any confidence level ν, the *recall* is the number of correctly predicted object locations with confidence at least ν divided by the total number of objects in this class. The *precision* is the same number of correctly detected objects, divided by the total number of boxes with a confidence of ν or more.

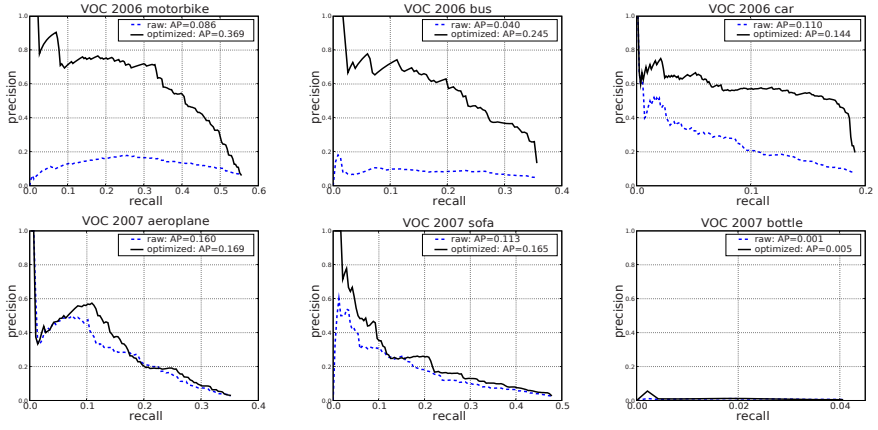

Fig. 2. Typical Precision–Recall Curves for VOC 2006 (top) and VOC 2007 (bottom). The blue (dashed) curve corresponds to the raw scores of the single-class candidate prediction, the black (dark) curve to the filtered results of the jointly optimized system.

A predicted bounding box B is counted as correct, if its area overlap $\frac{area(B \cap G)}{area(B \cup G)}$ with a ground truth box G of that class is at least 50%. From the precision–recall curves, an *average precision* score (AP) can be calculated by determining the maximal recall in 11 subintervals of the recall axis and averaging them, see [9] for details. Note, however, that AP scores are unreliable when they fall below 0.1 and should not be used to draw relative comparison between methods in this case.

Figure 2 shows results for three classes each of the VOC 2006 and of the VOC 2007 dataset. The plots contain the precision–recall curves of using either the score that the single-class candidate search returns, or the output of the learned joint-classifier as confidence values.

Table 1 lists the AP scores for all 30 classes in VOC 2006 and VOC 2007. In addition to the single-class scores and the joint-learning score, the results of the corresponding winners in the VOC 2006 and VOC 2007 challenge are included, illustrating the performance of the best state-of-the-art systems.

5.1 Discussion

The plots in Figure 2 and the list of scores in Table 1 show that the joint learning of confidence scores improves the detection results in the majority of cases over the single-class hypothesis prediction, in particular in the range of reliable values $AP > 0.1$. The increase in performance is more prominent in the VOC 2006 dataset than in 2007 (*e.g.* Figure 2, left column). For several classes, the system achieves results which are comparable to the participants of the VOC challenges and it even achieves better scores in the three categories bus-2006, sofa-2007 (Figure 2, center column) and dog-2007. The score for diningtable-2007 is higher than the previous one as well, but it is unreliable. There are

Table 1. Average Precision (AP) scores for VOC 2006 (top) and VOC 2007 (bottom)

VOC2006	bicycle	bus	car	cat	cow	dog	horse	motorbike	person	sheep
single-class	0.351	0.040	0.110	0.079	0.032	0.038	0.019	0.086	0.005	0.108
jointly learned	0.411	0.245	0.144	0.099	0.098	0.089	0.045	0.369	0.091	0.091
VOC 2006 best	0.440	0.169	0.444	0.160	0.252	0.118	0.140	0.390	0.164	0.251

VOC2007	aeroplane	bicycle	bird	boat	bottle	bus	car	cat	chair	cow
single-class	0.160	0.144	0.097	0.020	0.001	0.174	0.120	0.228	0.006	0.053
jointly learned	0.169	0.162	0.052	0.019	0.005	0.168	0.126	0.188	0.009	0.055
VOC 2007 best	0.262	0.409	0.098	0.094	0.214	0.393	0.432	0.240	0.128	0.140

	table	dog	horse	motorbike	person	plant	sheep	sofa	train	tv
single-class	0.049	0.150	0.032	0.207	0.116	0.004	0.092	0.113	0.101	0.055
jointly learned	0.101	0.165	0.048	0.219	0.089	0.023	0.092	0.165	0.118	0.042
VOC 2007 best	0.098	0.162	0.335	0.375	0.221	0.120	0.175	0.147	0.334	0.289

some cases, in which both stages of the system fail to achieve a good detection rate compared to the state-of-the-art, *e.g.* car-2006 or bottle-2007. Analyzing the precision–recall curves shows this is typically due to a bad set of candidate region. The maximum recall level in the examples is below 20% and 5% (Figure 2, right column). One cannot hope to achieve better scores in these cases, because the post-processing only assign a confidence to the object regions but cannot create new ones. We expect that a better hypothesis generation step and a test procedure predicting several candidate boxes per image would improve on this.

5.2 Dependency Graphs

Besides improving the localization performance, the multiple kernel learning also predicts class-specific dependency coefficients β_j^k that allow us to form a sparse

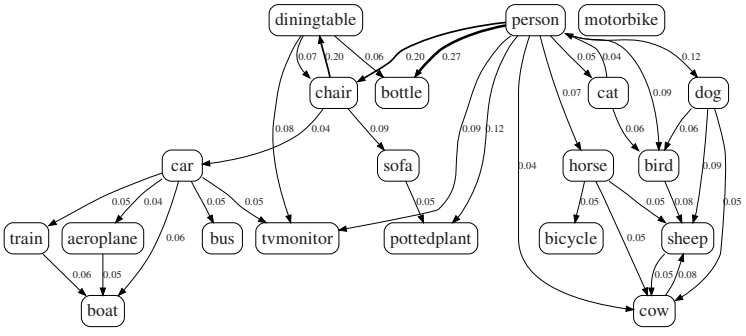

Fig. 3. Automatically learned dependency graph between the classes in VOC 2007. An arrow $\omega_j \rightarrow \omega_k$ means that ω_j helps to predict ω_k. This effect can be excitatory or inhibitory. All classes additionally depend on themselves and on the full image (not shown). The score and width of the arrow indicates the relative weight $\beta_k^j / \sum_{l=1}^{K} \beta_k^l$ without the image component. Connections with a score below 0.04 have been omitted.

dependency graph. These dependencies are non-symmetric, in contrast to generative measures like co-occurrence frequencies or cross-correlation. Figure 3 shows the automatically generated graph for VOC 2007. One can see that semantically meaningful groups have formed (*vehicles, indoors, animals*), although no such information was provided at training time.

6 Conclusions

We have demonstrated how to perform joint object-class prediction as a post-processing step to arbitrary single-class localization systems. This allows the use of class dependencies while remaining computationally feasible. The method is based on a maximum margin classifier using a linear combination of kernels for the different object classes. We gave an efficient training procedure based on formulating the problems as a collection of convex optimization problems. For each class, the training procedure automatically identifies the subset of object classes relevant for the prediction. This provides a further speedup at test time and allows the formation of an *a posteriori* dependency graph.

Experiments on the VOC 2006 and 2007 datasets show that the joint decision is almost always able to improve on the scores that the single-class localization system provided, resulting in state-of-the-art detection rates, if the set of candidate regions allows so. The resulting dependency graph has a semantically meaningful structure. Therefore, we expect that the learned dependency coefficients will be useful for other purposes as well, *e.g.* to generate class hierarchies.

Acknowledgements. This work was funded in part by the EC project CLASS, IST 027978. The second author is supported by a Marie Curie fellowship under the EC project PerAct, EST 504321.

References

1. Bay, H., Tuytelaars, T., Gool, L.J.V.: SURF: Speeded up robust features. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3954, pp. 404–417. Springer, Heidelberg (2006)
2. Bergtholdt, M., Kappes, J.H., Schnörr, C.: Learning of graphical models and efficient inference for object class recognition. In: Franke, K., Müller, K.-R., Nickolay, B., Schäfer, R. (eds.) DAGM 2006. LNCS, vol. 4174, pp. 273–283. Springer, Heidelberg (2006)
3. Biederman, I.: Recognition by components - a theory of human image understanding. Psychological Review 94(2), 115–147 (1987)
4. Biederman, I., Mezzanotte, R.J., Rabinowitz, J.C.: Scene perception: Detecting and judging objects undergoing relational violations. Cognitive Psychology 14, 143–177 (1982)
5. Bosch, A., Zisserman, A., Muñoz, X.: Representing shape with a spatial pyramid kernel. In: CIVR, pp. 401–408 (2007)
6. Chum, O., Zisserman, A.: An exemplar model for learning object classes. In: CVPR, pp. 1–8 (2007)

7. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR, pp. 886–893 (2005)
8. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2007 Results (2007), http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html
9. Everingham, M., Zisserman, A., Williams, C.K.I., Van Gool, L.: The PASCAL Visual Object Classes Challenge 2006 Results (2006), http://www.pascal-network.org/challenges/VOC/voc2006/results.pdf
10. Fergus, R., Perona, P., Zisserman, A.: A sparse object category model for efficient learning and exhaustive recognition. In: CVPR, pp. 380–387 (2005)
11. Ferrari, V., Fevrier, L., Jurie, F., Schmid, C.: Groups of adjacent contour segments for object detection. PAMI 30, 36–51 (2008)
12. Fritz, M., Leibe, B., Caputo, B., Schiele, B.: Integrating representative and discriminative models for object category detection. In: ICCV, pp. 1363–1370 (2005)
13. Keysers, D., Deselaers, T., Breuel, T.M.: Optimal geometric matching for patch-based object detection. ELCVIA 6(1), 44–54 (2007)
14. Kienzle, W., Bakır, G.H., Franz, M.O., Schölkopf, B.: Face detection - efficient and rank deficient. In: NIPS (2004)
15. Lampert, C.H., Blaschko, M.B., Hofmann, T.: Beyond sliding windows: Object localization by efficient subwindow search. In: CVPR (2008)
16. Lanckriet, G.R.G., Cristianini, N., Bartlett, P., Ghaoui, L.E., Jordan, M.I.: Learning the kernel matrix with semidefinite programming. JMLR 5, 27–72 (2004)
17. Lienhart, R., Liang, L., Kuranov, A.: A detector tree of boosted classifiers for real-time object detection and tracking. ICME 2, 277–280 (2003)
18. Ommer, B., Buhmann, J.M.: Learning the compositional nature of visual objects. In: CVPR (2007)
19. Opelt, A., Pinz, A., Fussenegger, M., Auer, P.: Generic object recognition with boosting. PAMI 28(3), 416–431 (2006)
20. Rabinovich, A., Vedaldi, A., Galleguillos, C., Wiewiora, E., Belongie, S.: Objects in context. In: ICCV (2007)
21. Schölkopf, B., Smola, A.J.: Learning with Kernels. MIT Press, Cambridge (2002)
22. Schulz, W., Enzweiler, M., Ehlgen, T.: Pedestrian recognition from a moving catadioptric camera. In: Hamprecht, F.A., Schnörr, C., Jähne, B. (eds.) DAGM 2007. LNCS, vol. 4713, pp. 456–465. Springer, Heidelberg (2007)
23. Sonnenburg, S., Rätsch, G., Schäfer, C., Schölkopf, B.: Large scale multiple kernel learning. JMLR 7, 1531–1565 (2006)
24. Teynor, A., Burkhardt, H.: Patch based localization of visual object class instances. In: MVA (2007)
25. Torralba, A.: Contextual priming for object detection. IJCV 53(2), 169–191 (2003)
26. Torralba, A., Murphy, K.P., Freeman, W.T.: Shared features for multiclass object detection. In: Toward Category-Level Object Recognition, pp. 345–361 (2006)
27. Torralba, A., Oliva, A.: Statistics of natural image categories. Network: Computation in Neural Systems 14(3), 391–412 (2003)
28. Viola, P.A., Jones, M.J.: Robust real-time face detection. IJCV 57(2), 137–154 (2004)
29. Zhang, J., Marszalek, M., Lazebnik, S., Schmid, C.: Local features and kernels for classification of texture and object categories: A comprehensive study. IJCV 73(2), 213–238 (2007)
30. Zien, A., Ong, C.S.: Multiclass multiple kernel learning. In: ICML, pp. 1191–1198 (2007)