Learning Search Based Inference for Object Detection

Peter Gehler PGEHLER@TUE.MPG.DE
MPI for Intelligent Systems

Alain Lehmann LEHMANN@VISION.EE.ETHZ.CH
ETH Zurich

Abstract

We study the problem of visual object detection and propose a method that learns the inference procedure during training time. We propose a best-first search based inference system that is already optimized for during training. This overcomes the inherent limitation of branch&bound whose applicability relies on the availability of *tight* bounding functions. The optimization problem is implemented using a structured maximum margin formulation, an efficient test time inference is already "learned" during training time. Based on this technique we show how to perform object detection using non-linear SVM only, without the need of cascade-like approximations. We demonstrate the algorithmic properties using the VOC'07 dataset.

1. Introduction

Object class detection in images is challenging because of two problems. First, objects exhibit large variations due to intra-class variability, illumination changes, *etc*. Second, objects may appear anywhere in an image with unknown scale, and need to be localised. Much progress has been reported lately, manifesting in increasing evaluation scores of the VOC benchmark (Everingham et al.). In this paper we are studying an algorithmic approach that focuses on *detection efficiency*. Our approach is designed with two demands in mind: detectors must cope with the appearance variations and must handle the large search space efficiently.

This workshop paper is a shorter version of a conference contribution of this work (Lehmann et al., 2011). Please refer to the conference version for more details including a full discussion of the related literature.[1]

In current object detection systems, appearance variations of images are best captured by non-linear SVM classifiers that make use of diverse image features (Vedaldi et al., 2009; Gehler & Nowozin, 2009). Unfortunately these classifiers are computationally demanding and yield a challenging inference problem. This is because of the large search space of typically $>10k$ bounding boxes per image. This leaves two possible options to handle the search space: (A) reducing the cost of a single classifier evaluation (Viola & Jones, 2004; Vedaldi et al., 2009) or (B) reducing the number of classifier calls (Lampert et al., 2009; Lehmann et al., 2010). Let us stress this point: we distinguish between the cost of the classifier and the number of times it is executed. These two factors are orthogonal and their product yields the total runtime. In this work we focus primarily on (B), noting that combination of both options are also possible (Lampert, 2010; Weiss et al., 2010).

Cascade classifiers (Viola & Jones, 2004; Vedaldi et al., 2009; Felzenszwalb et al., 2010) are prominent examples to approach (A). They reject many hypothesis with a simple criterion and thereby avoid many computations. However, they do not per-se reduce the number of calls and the total runtime still scales linearly in the number of detection sites. We consider these approaches to be fast but not efficient as they do not *scale* well (*e.g.* to multi-class). Branch&bound methods (Lampert et al., 2009) fall into the category (B). They reduce the number of calls by avoiding exhaustive search. This is possible by operating on *sets* of hypotheses. The detector adaptively partitions the search space and focuses on the most promising set. This best-first search allows for impressive runtime given a *tight* bound on the classifier function. Tight bounds are however a severe limitation as they are often unavailable expect for simple function classes *e.g.* linear SVMs.

[1] see www.vision.ee.ethz.ch/~lehmanal/publications.html for material.

Presented at the International Conference on Machine Learning (ICML) workshop on *Inferning: Interactions between Inference and Learning*, Edinburgh, Scotland, UK, 2012.

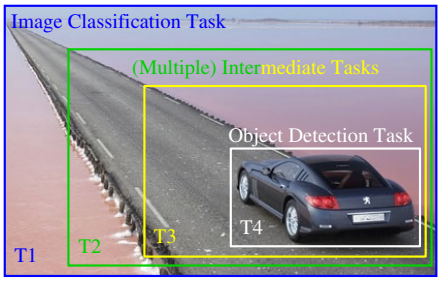

Figure 1. Detecting an object in an image is decomposed into different tasks. The approach smoothly blends from image classification (T1) to object detection (T4).

This paper supersedes the notion of bounding and thereby allows for using arbitrary classifiers. We adopt the best-first search and "branch" but do not "bound". Instead, we explicitly integrate the idea of scoring sets into the training problem. Intuitively speaking we aim to "learn the bound". More precisely, we learn a ranking function that prioritises hypothesis sets that do *contain an* object over those that do not. We deliberately choose to work with non-linear SVMs and RBF-χ^2 kernels. These classifiers have been shown to perform well (Lazebnik et al., 2006; Vedaldi et al., 2009), but are generally perceived as being too slow to be directly applicable; we here show that using *only* evaluations of these non-linear SVMs is feasible. We train them in a multi-task setup which accounts for the size of hypothesis sets; we thereby separate image classification from object recognition, yet combine them in a joint objective (*c.f.* Fig. 1).

2. Branch and Rank

The test time inference of object detection is to predict a tight bounding box around the object in an image. We parametrise a bounding box by its center (x, y), scale s, and aspect-ration r. Sets of bounding boxes $\Lambda = \left(I, \underline{x}, \overline{x}, \underline{y}, \overline{y}, \underline{s}, \overline{s}, \underline{r}, \overline{r}\right)$ comprise all bounding boxes in image I with center $(x, y) \in [\underline{x}, \overline{x}] \times [\underline{y}, \overline{y}]$, and scale/aspect-ratio in the intervals $[\underline{s}, \overline{s}]$ and $[\underline{r}, \overline{r}]$.

The test-time inference problem is implemented as a adaptive, best-first search strategy that is governed by a priority queue. Initially, the whole search space is entered into the queue as a single element. At each step the highest ranking element in the priority queue is split in two smaller sets which are subsequently scored and inserted into the priority queue. In case the highest ranking hypotheses set is a single bounding box (or a sufficiently small set) a detection is reported. This results in a kD-tree partitioning of the entire search space. See (Lehmann et al., 2010) for more details.

Before we turn to learning, imagine we had available a function f^{oracle} that at any point during inference sorts the elements in the priority queue in the best possible way. In that case, the algorithm will first examine all sets that do contain an object and the size of these sets decreases exponentially fast (as we always split them into two halves). That implies that objects of interest are detected in logarithmic time (aka binary search). Of course, this cannot be expected in practice, but it suggests: the better a ranking function, the better *and faster* the detector.

2.1. Learning Problem

Let $\mathbb{L}$ be the set of all hypotheses sets while $\mathbb{L}^+$ represents only those that *contain* at least one object. When using a priority queue during inference we would like a function f that keeps it sorted depending on whether the object is present in the set, namely

$$f(\Lambda^+) > f(\bar{\Lambda}) \qquad \forall \Lambda^+ \in \mathbb{L}^+, \quad \forall \bar{\Lambda} \in \mathbb{L} \setminus \mathbb{L}^+, \quad (1)$$

where the set Λ^+ contains at least one object and $\bar{\Lambda}$ contains no object (or only with partial overlap).

We implement learning using a max-margin formulation with margin-rescaling

$$\min_{f, \xi \geq 0} \quad \|f\|^2 + C \textstyle\sum_{j=1}^n \xi_j \quad (2)$$

$$\text{sb.t.} \quad f(\Lambda_j^+) - f(\Lambda) \geq \Delta(\Lambda_j^+, \Lambda) - \xi_j, \quad \forall \Lambda_j^+ \in \mathbb{L}^+, \forall \Lambda \in \mathbb{L} \setminus \mathbb{L}^+ \quad (3)$$

with slack variables ξ_j for every positively annotated example $\{\Lambda_j^+\}_{j=1}^n$ and regularisation parameter C, that trades data fit with model complexity. The loss $\Delta(\Lambda_1, \Lambda_2) \mapsto \mathbb{R}$ encodes the cost of predicting Λ_2 if Λ_1 were correct. As we tackle object (not instance) detection, we need to handle the case of multiple objects in an image. Therefore the standard VOC loss is extended to sets of bounding boxes, noting that it depends on the entire training annotations Y:[2]

$$\Delta(\Lambda) := \Delta(\Lambda_j^+, \Lambda) = 1 - \max_{\substack{\lambda_i \in Y \\ \lambda \in \Lambda}} \frac{area(B(\lambda) \cap B(\lambda_i))}{area(B(\lambda) \cup B(\lambda_i))}$$

with bounding box $B(\lambda)$. This definition exploits the fact that during training the first argument is always a positive example.

2.2. Multi-Task Decomposition

The previous section's formulation suggests *one* ranking function for all possible sets of bounding boxes. Let us consider the two extremes of such sets. At one

[2] see (Lehmann et al., 2011) for details

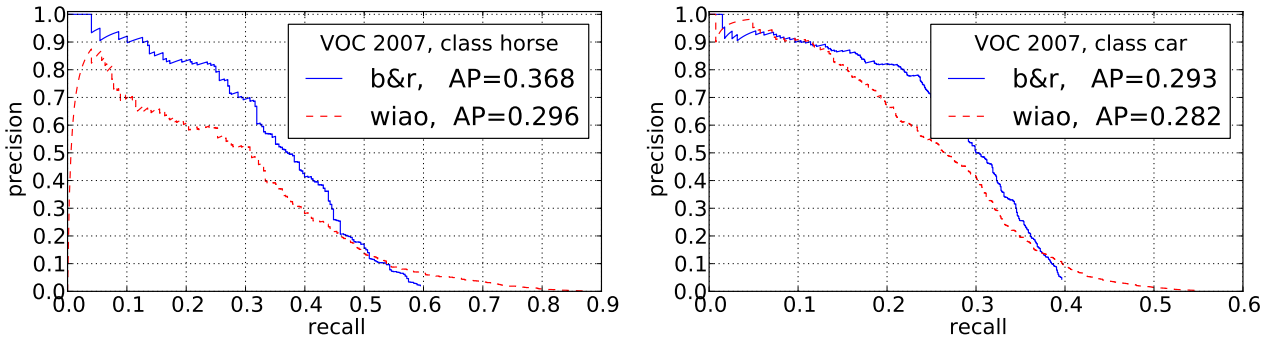

Figure 2. Precision-recall curves for our method (b&r) with 400 proposals of (Alexe et al., 2010) (wiao) scored with task T classifier of our method (on classes *horse,car*). (wiao) provides high recall (right end of red curve) which makes it a good approximation to sliding window. Our method compares well and even improves over the baseline in terms of AP. Drop in AP is mainly due to missing recall (*e.g.*, dog).

end, the initial set represent the entire image and all possible sub-windows. Scoring this set is the task of image classification. The other extreme is a hypotheses set with only one instance, corresponding to scoring a single bounding box. This is an object recognition problem. Both of course are related, but note the difference in the tasks: the first set should have a high score if it *contains an* object, the latter if it *is centered* on the object. This suggest that these tasks are better solved separately *but* combined in a joint objective. For example the first task could benefit from different image features such as the gist of a scene (Oliva & Torralba, 2001), while the latter could make use of object specific features (Oliva & Torralba, 2001; Dalal & Triggs, 2005; Lowe, 2004).

2.3. Multitask Learning Problem

We implement the aforementioned intuition in the problem by decomposing the to be learned function into different regimes, depending on the size of the bounding box set to be scored. We define the size of a hypothesis set as

$$|\Lambda| = \frac{\overline{x}-\underline{x}}{\overline{s}} \times \frac{\overline{y}-\underline{y}}{\overline{s}} \times (\log \overline{s} - \log \underline{s}) \times (\log \overline{r} - \log \underline{r}) \quad (4)$$

and discretize the input set sizes $\log(|\Lambda|)$ uniformly into T different tasks to account for the exponential decay (due to splitting scheme). With $q(\Lambda) \mapsto \{1, 2, \ldots, T\}$ we denote the task mapping that assigns the input to the corresponding function. For each task separately we have a function of the form $f(\Lambda) = \langle w_{q(\Lambda)}, \Phi(\Lambda) \rangle + b_{q(\Lambda)}$ with per-task weight vectors w_t and bias terms b_t. In the experiment we will use kernelized version of the function. It turns out that with fixed function q the original problem (2) can be equivalently decomposed into the following T convex optimization problems

$$\begin{aligned} \min_{w_t, b_t, \xi \geq 0} \quad & \|w_t\|^2 + C \textstyle\sum_j \xi_j \\ \langle w_t, \Phi(\Lambda_j^+) \rangle + b_t \quad & \geq -\xi_j \quad \forall \Lambda_j^+ \in \mathbb{L}^+,\ q(\Lambda_j^+) = t \\ \langle w_t, \Phi(\Lambda) \rangle + b_t \quad & \leq -\Delta(\Lambda) \quad \forall \Lambda \in \mathbb{L} \setminus \mathbb{L}^+,\ q(\Lambda) = t \end{aligned} \quad (5)$$

that uses only examples from one given task t.We solve the problem (6) in sequential order using delayed constraint generation and a modified version of SVM^{struct} (Tsochantaridis et al., 2005). Both for test-time as well as for loss-augmented inference the search procedure is applied. During learning "positive" hypothesis sets Λ_j^* are created using a "ground-truth" detector making use of the annotated bounding boxes for the training set.

3. Experiments

To show the benefit of the search based inference we chose to use the kernelized version only, since non-linear kernel classifiers have been perceived as being to costly to directly be applicable for object detection. We use a RBF-χ^2-kernel $k(u, v) = \exp(-\gamma \sum_l \frac{(u_l - v_l)^2}{u_l + v_l})$.

Dataset and Experimental Setup We perform experiments on the VOC'07 dataset (Everingham et al.) consisting of 20 classes and about $10k$ images. Performance is measured using the official average precision (AP) score function. The hyper-parameter C is chosen by cross-validation on the train (val) data splits. As features we resort to rgb-SIFT descriptors (using the code of (van de Sande et al., 2010)) being extracted on a dense grid at multiple scales. Using a standard pyramid histogram representation this yields a 2100D feature vector. For hypothesis sets we use the

	aerop	bicyc	bird	boat	bottle	bus	car	cat	chair	cow	dtable	dog	horse	mbike	person	plant	sheep	sofa	train	tv
T1	9.6	12.8	2.3	3.1	1.1	11.4	16.3	11.7	9.1	9.5	5.0	9.3	18.3	15.7	10.0	0.1	2.3	5.5	15.4	10.7
T6	21.8	23.2	2.9	9.8	9.1	20.3	23.0	18.1	9.4	10.8	10.3	9.2	30.0	28.9	11.6	1.5	10.3	13.6	24.9	15.6
wiao(Alexe et al., 2010)	17.6	18.6	0.7	1.2	9.1	28.6	28.2	13.7	1.3	12.8	6.1	14.8	29.6	25.5	13.7	1.5	11.4	14.3	22.6	16.3
b&r	17.3	22.4	0.2	0.6	9.1	27.6	29.3	17.6	3.3	10.8	12.2	13.2	36.8	27.9	14.2	1.7	13.0	15.3	22.5	18.4
dt(Felzenszwalb et al., 2008)	18.0	41.1	9.2	9.8	24.9	34.9	39.6	11.0	15.5	16.5	11.0	6.2	30.1	33.7	26.7	14.0	14.1	15.6	20.6	33.6
v7(Everingham et al.)	26.2	40.9	9.8	9.4	21.4	39.3	43.2	24.0	12.8	14.0	9.8	16.2	33.5	37.5	22.1	12.0	17.5	14.7	33.4	28.9
$\#f$	80	70	160	160	160	80	70	60	160	80	80	60	60	70	100	180	80	60	60	100

Table 1. Average precision results on VOC'07. T1,T6: the influence of using 1 or 6 tasks, evaluated on split *val*. The second group shows the final evaluation on *trainval*/*test*-split. Branch&rank (b&r) uses 6 tasks and the C value from T6. The performance is compared to 400 "what is an object" proposals (wiao), the state-of-the-art detector (dt) and the best result in the challenge (v7). The average number of classifier calls ($\#f$) at prec=recall shows the efficiency of (b&r).

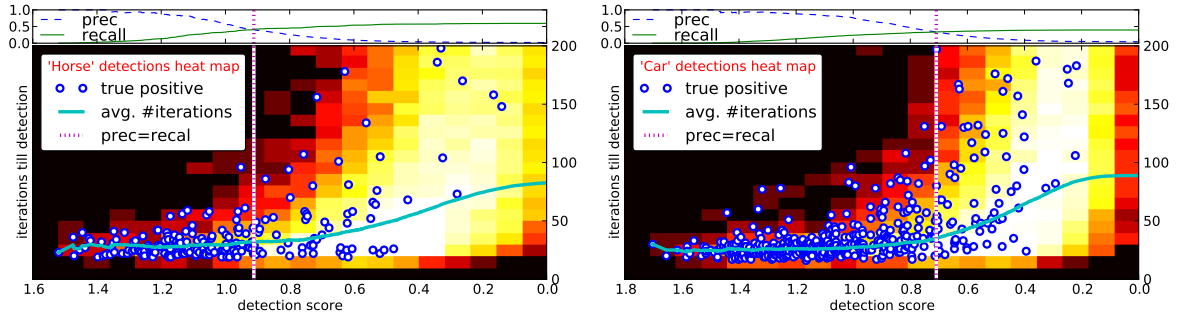

Figure 3. Iterations till detection versus object detection score for classes Horse, Car, and Dog. The heat map represents the joint (score–iteration) density (bright indicating high density) that is estimated from all detections (reported by our final system); the blue circles denote the true positive detections only. Our method uses less than 100 iterations on average (cyan), about 40 at precision=recall (magenta), and sometimes only 20.

image features computed on the union of the bounding boxes it contains. A combination of more and diverse features is likely to improve performance, we chose a simpler method to demonstrate the algorithmic properties of the proposed method.

Detection Results We first report the detection results achieved on this dataset. We compare the AP of our branch&rank (b&r) approach with a state-of-the-art detector (Felzenszwalb et al., 2008) (dt) as well as the best (per category) results reported in the challenge (Everingham et al.) (v7).

Table 1 shows that our scores are sometimes higher (dtable,horse), lower (*e.g.* bicyc,boat,bus,car), or in between (*e.g.* cat,dog). This is the ranking we have expected using only a single un-tuned image descriptor. The work by (Vedaldi et al., 2009) convincingly demonstrated that combining multiple complementary features significantly improves detection quality and we expect the same gain would be achieved here.

Quantifying the search error The next experiment aims to empirically quantify the search error that is incurred because the test time inference problem is approximate. The golden standard to compare against would be sliding windows: evaluate every possible bounding box. As this is infeasible, we use a proposal-verification algorithm (Alexe et al., 2010) that generates candidate regions to evaluate the classifier on. In accordance to (Alexe et al., 2010) we find that this is indeed a good approximation to exhaustive search and yields high recall rates (right most point of the red lines in Fig 2). We fix the computational budget to 400 function calls for both the baseline and the search-based inference. Thus we score the top 400 proposals using the learned function for task T (wiao) and compare the performance to running 200 iterations of our algorithm (b&r). The results are shown in Table 1 and comparing (wiao) and (b&r) we notice that the results are indeed very similar.

Detector Efficiency We measure the efficiency in number of classifier evaluations (since the classifier can be sped up using standard techniques such as cascades). Fig. 3 plots the number of iterations till a detection is reported as a function of the detection score and in Tab. 1 the line $\#f$ shows the average number of calls for all categories. Our system detects most objects rather quickly (in usually less than 50 iterations) especially the high scoring ones but the average

number of iterations is also quite low. This finding reinforces our conjecture from Sec.2: the better the ranking, the faster the detector. Higher values of $\#f$ are those with lower scores in Tab. 1.

Multi-Task comparison Finally we evaluate the influence of the multi-task ranking we proposed. To this end we train two detectors, one making use of $T = 6$ different tasks (T6) and one that is not divided into different tasks, setting $T = 1$ (T1). Hyperparameters are optimized individually. The results are reported in Table 1 (lines T1,T6). We observe a consistent improvement of the multi-task decomposition (T6) over the holistic classifier (T1). From this we conclude that the task dependent decomposition is indeed a crucial component for good performance.

4. Conclusion

This work combines a best-first search based inference with a multi-task decomposition for the task of object detection. This strategy enables us to use non-linear classifiers throughout the system. This is a crucial step towards efficient object detection, since it allows to model the intra-class variations with stronger, but potentially more costly classifiers. It operates using a priority queue and thus avoids search space pruning.

The system could be improved by using complementary image descriptors, factoring in feature computation time, or learning the task-decomposition. In addition to using the structure of the search space, one could also use cascades to speed the classifier calls themselves.

References

Alexe, B., Deselaers, T., and Ferrari, V. What is an object? In *CVPR*, 2010.

Dalal, N. and Triggs, B. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results.

Felzenszwalb, P., McAllester, D., and Ramanan, D. A discriminatively trained, multiscale, deformable part model. In *CVPR*, 2008.

Felzenszwalb, P., Girshick, R., and McAllester, D. Cascade object detection with deformable part models. In *CVPR*, 2010.

Gehler, Peter and Nowozin, Sebastian. On feature combination for multiclass object classification. In *ICCV*, 2009.

Lampert, Christoph H. An efficient divide-and-conquer cascade for nonlinear object detection. In *CVPR*, 2010.

Lampert, Christoph H., Blaschko, Matthew B., and Hofmann, Thomas. Efficient subwindow search: A branch and bound framework for object localization. *PAMI*, 99(1), July 2009. ISSN 0162-8828. doi: http://doi.ieeecomputersociety.org/10.1109/TPAMI.2009.144.

Lazebnik, Svetlana, Schmid, Cordelia, and Ponce, Jean. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006.

Lehmann, Alain, Leibe, Bastian, and Van Gool, Luc. Fast prism: Branch and bound hough transform for object class detection. *IJCV*, April 2010.

Lehmann, Alain, Gehler, Peter, and Gool, Luc Van. Branch&rank: Non-linear object detection. In *BMVC*, 2011. www.vision.ee.ethz.ch/ lehmanal/publications.html.

Lowe, David G. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004. ISSN 0920-5691. doi: http://dx.doi.org/10.1023/B:VISI.0000029664.99615.94.

Oliva, Aude and Torralba, Antonio. Modeling the shape of the scene: a holistic representation of the spatial envelope. *IJCV*, 42(3):145–175, 2001.

Tsochantaridis, Ioannis, Hofmann, Thomas, Joachims, Thorsten, and Altun, Yasemin. Large margin methods for structured and interdependent output variables. *JMLR*, 6:1453–1484, 2005.

van de Sande, K. E. A., Gevers, T., and Snoek, C. G. M. Evaluating color descriptors for object and scene recognition. *PAMI*, 32(9):1582–1596, 2010. URL http://www.science.uva.nl/research/publications/2010/vandeSandeTPAMI2010.

Vedaldi, Andrea, Gulshan, Varun, Varma, Manik, and Zisserman, Andrew. Multiple kernels for object detection. In *ICCV*, 2009.

Viola, Paul A. and Jones, Michael J. Robust real-time face detection. *IJCV*, 57(2):137–154, 2004.

Weiss, David, Sapp, Benjamin, and Taskar, Ben. Sidestepping intractable inferece with structured ensemble cascades. In *NIPS*, 2010.