UNIVERSITÄT AUGSBURG
Universitätsbibliothek

OPEN ACCESS
OPUS AUGSBURG

Automatic object annotation from weakly labeled data with latent structured SVM

Christian X. Ries, Fabian Richter, Stefan Romberg, Rainer Lienhart

Angaben zur Veröffentlichung / Publication details:

Ries, Christian X., Fabian Richter, Stefan Romberg, and Rainer Lienhart. 2014. "Automatic object annotation from weakly labeled data with latent structured SVM." In *12th International Workshop on Content-Based Multimedia Indexing (CBMI), 18-20 June 2014, Klagenfurt, Austria*, 1–4. Piscataway, NJ: IEEE. https://doi.org/10.1109/cbmi.2014.6849838.

Nutzungsbedingungen / Terms of use:

licgercopyright

Dieses Dokument wird unter folgenden Bedingungen zur Verfügung gestellt: / This document is made available under these conditions:
Deutsches Urheberrecht
Weitere Informationen finden Sie unter: / For more information see:
https://www.uni-augsburg.de/de/organisation/bibliothek/publizieren-zitieren-archivieren/publiz/

Automatic Object Annotation From Weakly Labeled Data With Latent Structured SVM

Christian X. Ries Fabian Richter Stefan Romberg Rainer Lienhart
Augsburg University
86159 Augsburg, Germany
Email: {ries, richter, romberg, lienhart}@informatik.uni-augsburg.de

***Abstract*—In this paper we present an approach to automatic object annotation. We are given a set of positive images which all contain a certain object and our goal is to automatically determine the position of said object in each image. Our approach first applies a heuristic to identify initial bounding boxes based on color and gradient features. This heuristic is based on image and feature statistics. Then, the initial boxes are refined by a latent structured SVM training algorithm which is based on the CCCP training algorithm. We show that our approach outperforms previous work on multiple datasets.**

I. Introduction

In this paper, we address the issue of automatically annotating objects of interest by determining their bounding boxes in images. We only exploit "weak labels", i.e. binary image labels which indicate if a desired object is present in a given training image. In the context of our work, a desired object is an instance from an object class with relatively low intra-class variance with regards to a given set of features. Our goal is to reduce the annotation process from manually drawing bounding boxes to providing a set of positive images. Our work builds on previous work of Ries et al. [1] which we thus use as our baseline. In comparison to this work we have a number of novel contributions: First, we unify parameter selection for different feature types in the heuristic for initial bounding boxes. Second, we enhance the method by adding the capability to identify (i.e. annotate) multiple object per image. Third, we propose using a latent structured Support Vector Machine (SVM) training algorithm for refining bounding boxes. We require that the following assumptions hold with regard to the features we use: (1) a negative image set is provided which is relatively representative for background of the positive features, and (2) the background of the positive images is more diverse than the wanted objects themselves. Random photos usually fulfill the first requirement.

II. Related Work

As mentioned above our work is based on [1] which is also used as the baseline in our evaluation section. As local features we use HOG (Histogram of Oriented Gradients) features [2]. We train our latent structured SVM by an instance of the CCCP (Convex-concave Procedure) algorithm [3] which was suggested for latent structured problems in [4]. Our implementation of the CCCP algorithm is inspired by Zhu et al. [20]. Testing each possible rectangle within our images is sped up by computing the dot product of the linear SVM with integral images analogous to [5].

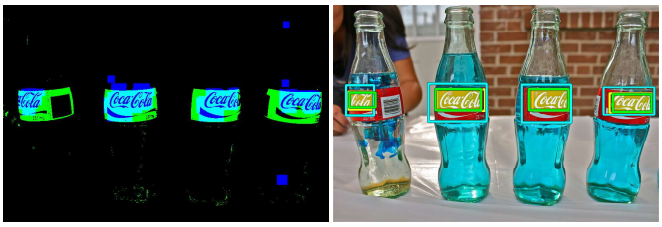


Fig. 1: Left: result of applying a color model (green pixels), a HOG model (blue), and their combination (cyan) to an image of class "Coca Cola brand logo". Right: initial (cyan) and final (green) result of our annotation algorithm. Yellow boxes (mostly covered) are ground truth.

In recent research, the idea of learning from weakly labeled data has attracted much interest. For instance, in [6] discriminative segment annotations from weakly labeled video are determined. One of the scenarios is called transductive segment annotation which denotes finding a common object within weakly labeled video frames. In [7] the task of annotating weakly labeled data exploiting a large amount of negative data is discussed. Also, the concept of Multiple Instance Learning (for instance used in [8], [9]) is related to our problem. In another recent work [10], synthetic training examples in weakly labeled videos are searched using only a few manually labeled training examples. Another interesting approach [11] has a similar goal as automatic annotation: by defining an "objectness" measure, image regions are evaluated for their probability of showing an interesting object.

III. Initial Bounding Box Estimates

Let F be the discrete and finite domain of feature values of a certain type. We aim to determine a subset $F_{pos} \subset F$ which consists of positive features, i.e. features indicative for the wanted object class. An image property which is independent of the unknown sizes and positions of the object instances is the binary property $f(I) \in \{0, 1\}$ which indicates whether image I contains feature $f \in F$ at least once. We can now compute the relative number $P_P(f(I))$ of images in positive image set P which produce feature f. Analogously, we determine the probability $P_N(f(I))$ for negative image set N. Our task is now to determine based on $P_P(f(I))$ and $P_N(f(I))$ whether feature f is indicative for the wanted object. Since P is significantly smaller than N, we can model our

expectation for the number of images in P with $f(I) = 1$ for a background feature by a normal distribution (which is an approximated binomial distribution for $n > 30$) with $\sigma_f^2 = \mu_f \cdot (1 - \mu_f)n^{-1}$ where $n = |P|$ is the sample size, $\mu_f = P_N(f(I))$ the expected relative occurrence frequency of $f(I)$ in the background, and $nx_f = nP_P(f(I))$ the number of images in P with $f(I) = 1$. This allows us to determine positive features by a decision function $c(f)$ based on a one-sided confidence interval on x_f:

$$c(f) = \delta(x_f \notin [0, \mu_f + \theta_f]) \tag{1}$$

where $\delta(A)$ is 1 if A is true and 0 otherwise. Note that we remove those features which are relatively less frequently present in positive images than in negative images (i.e. $x_f < \mu_f$). All features for which $c(f) = 1$ form the set of positive features F_{pos}. In order to merge multiple features, we simply intersect the sets of positive pixels by a logical AND. The constant z_F determines which (i.e. how many) features are considered positive. We choose z_F (and thus θ_f) dynamically for each feature type, in contrast to the baseline approach [1]. If a feature type is not discriminative with respect to an object class, we want a relatively large set F_{pos}. The reason is that we intersect sets of positive pixels from multiple feature types. False positive features are therefore less harmful to our model than false negatives. We consider the "best" feature f^* we observe for feature type F a good indicator for a feature's distinctiveness. The best ("most positive") feature f^* is the one feature $f \in F$ which has the smallest value $N(x_f; \mu_f, \sigma_f^2)$. Now let z_F^* be the value of z_F for which f^* is the only feature included in the interval of $c(f)$. The value of our threshold then depends on αz_F^*, i.e. $\theta_f = \alpha z_F^* \sigma_f$, where α is an empirical constant, for which we experimentally determine $\alpha = 0.4$. Note that α is an intuitive factor which is independent of the feature type as opposed to z_F. Fig. 1 illustrates the results of applying the threshold by showing positive pixels for two feature types.

Since we do not know the number of instances in a positive image, we iteratively fit Gaussian Mixture models with $k \in \{1, ..., 5\}$ components into the set of positive pixels. For each k, we determine the likelihood of mixture parameters and find the value of k for which the largest increase in likelihood is observed (as long as the increase is larger than an empirically chosen threshold). We then estimate the i-th bounding box as centered around $(\mu_{i,x}, \mu_{i,y})$ with width and height of $3.2\sigma_{i,x}$ and $3.2\sigma_{i,y}$, where $\mu_{i,x}$, $\mu_{i,y}$, $\sigma_{i,x}$, and $\sigma_{i,y}$ are the mean values and standard deviations of the i-th Gaussian mixture component in x- and y-direction, respectively. The enlargement constant 3.2 proved to be a good choice based on experiments. We remove bounding boxes with extreme aspect ratios and merge overlapping boxes into a single larger box.

IV. Latent Structured Learning for Bounding Box Improvement

In this section we explain how we improve our box estimations by latent structured training. In our case, input instances are images, and the structured output label space Y is the space of possible rectangles. Since our initial labels will in many cases be incorrect, we do not know if a label describes the actual object position. Thus, we treat the true object position as an unobservable, i.e. latent, property h_j for each instance j, initialized by the respective bounding box from our initial model. Our feature representation $\Psi(x, y_j, h_j)$ depends on the latent variable h_j which is the current estimate for the object position. We now train a latent structured SVM with linear kernel and model vector $\mathbf{w}$ in order to find optimal values for our latent variables with decision function

$$f_{\mathbf{w}}(x) = \underset{(y,h) \in Y \times H}{argmax} \ \langle \mathbf{w}, \Psi(x, y, h) \rangle \tag{2}$$

To solve this task, we minimize the empirical risk on the training set, which is the average loss $\Delta(y_j, \hat{y}, \hat{h})$ on the training data. Like [20], we use the standard binary loss function. Following [12] and [4], the optimization problem can then be written as minimizing the difference of two convex functions, so we can solve it using an instance of the CCCP algorithm [3] as suggested for solving latent structured problems by Yu and Joachims [4]. The CCCP algorithm iteratively solves the optimization problem and updates the latent variables. Upon termination, the latent variables are then returned by our algorithm as our final bounding box estimations.

V. Implementation

For our initial model we use two features: pixel colors and Histograms of Oriented Gradients (HOG) [2]. Following the method of [1] based on [13], we create a histogram of positive colors. The correlation between pixels and features is trivial. The HOG features [2] are extracted on a dense 8×8 grid over multiple scales. We concatenate 2×2 HOG cells for each grid point in order to obtain a more meaningful description, and cluster the features into $10,000$ visual words. Positive pixels are pixels under patches belonging to positive HOG features.

For the feature representation $\Psi(x_j, y_j, h_j)$ of our training instances for the CCCP algorithm, we again use the HOG-based visual words from before as bag-of-words (BOW) histogram extracted from rectangle h_j. Since HOG features live on a regular grid, we define a number of reasonable aspect ratios on up to 17 different image scales. For solving the optimization problem (finding model vector $\mathbf{w}$) within the CCCP algorithm , we use the cutting plane algorithm proposed by Joachims et al. [14] and the solver of SVMLight [15]. Note that the algorithm requires finding the one rectangle with the maximum score for the current given model vector $\mathbf{w}$ for each negative training example which we implement efficiently based on the technique of efficient sub window search proposed by Lampert et al. [5].

For determining new values for the latent variables, we search for the highest scoring rectangles within all positive images. We ensure that the updated bounding box still has some small minimum overlap (0.05) with the bounding box from the initial model preventing predictions of multiple instances from "moving" towards a single object instance. For each iteration, we determine the one aspect ratio in terms of HOG grid cells which describes the majority of instances and then allow only 1 cell deviation from this ratio. Finally, we perform three post-processing steps: (1) We search in double-resolution images for very small instances. (2) On the smallest scale, we snap detections to the initial positive pixels. (3) We perform non-maximum suppression.

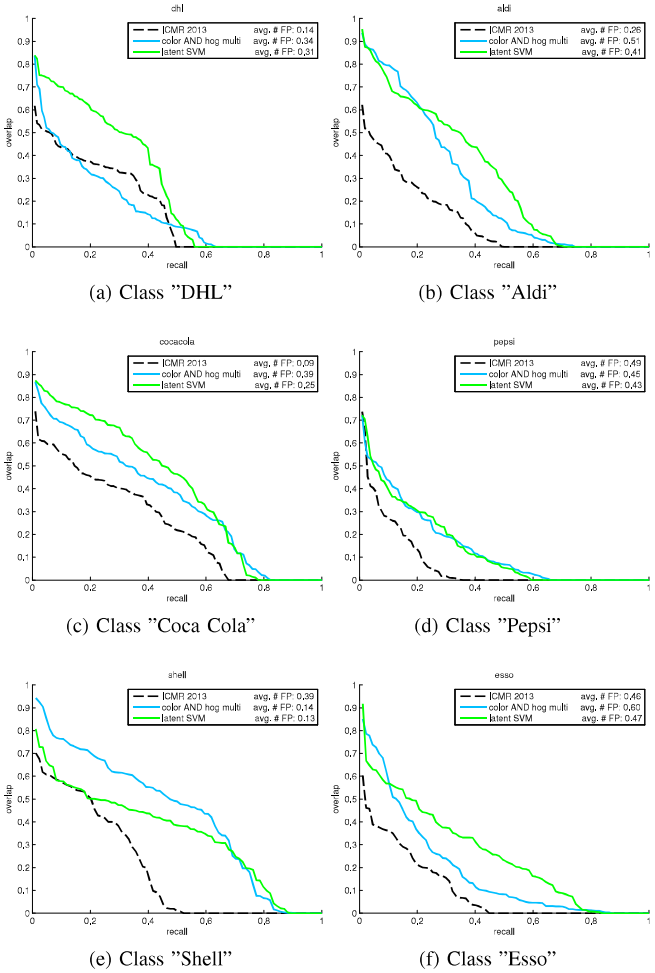
(a) (b) (c) (d) (e)

Fig. 2: Results from FlickrLogos-32 (a-c), Oxford Flowers (d), and 3D Objects (e) for the initial model (cyan) and the CCCP algorithm (green). False positives have respective darker colors in (b). Yellow marks the ground truth annotations.

VI. Evaluation

We evaluate our method on 3 publicly available datasets: FlickrLogos-32 [16], Oxford 17 Flowers [17], and 3D Object Categories [18]. Oxford 17 Flowers and a subset of FlickrLogos-32 were already used in the baseline work of [1]. We assess performance by overlap-recall (OR) plots which show which ratio of object instances (i.e. recall or ratio of true positive detections) are detected at a certain overlap with ground truth rectangles. The CCCP algorithm always used 1000 negative images from the negative class of the FlickrLogos-32 dataset. Since we detect multiple objects per image, we also state the average *absolute* number of false positive rectangles per image (not to be confused with a *relative* false positive rate) as "avg. # FP" in the legend of each plot. A few example results are given in Fig. 1 and Fig. 2. Fig. 2a, 2b, and 2e show examples where the CCCP algorithm finds a better bounding box than the initial model. Also, a false positive detection is shown in Fig. 2b (dark cyan for initial model, dark green for CCCP). Figures 2c and 2d show examples where the CCCP algorithm fails to improve the detections. In figure 2c, the CCCP algorithm returns a partial detection, while in 2d the bounding box is too large.

For our first experiment we select the same six logo classes as the baseline approach [1]. Our results are shown in Fig. 4 where the baseline is depicted by a dashed black curve. For most classes the baseline is outperformed significantly by the initial model (cyan line), except for "DHL" where it is roughly on par with regards to area under the curve. For all classes except "Pepsi" and "Shell" the results after using the CCCP algorithm (green line) are better than the initial model. The average number of false positive rectangles of 0.33 over all six classes is slightly higher than the baseline with 0.305, which estimates only one rectangle per image. We also test our approach on the remaining 26 classes of FlickrLogos-32 and combined the results into one curve in Fig. 4a. The CCCP algorithm only slightly improves the results compared to the initial model. However, only for 7 classes, the initial model yields better results due to the tendency of CCCP to converge towards partial, but more distinctive objects. One example for this issue is shown in Fig. 2c. Note that the CCCP algorithm still finds a distinct object which all positive images have in common. In another experiment we evaluate the usefulness of automatic object annotation for a logo recognition system based on k-nn classification of the top-k retrieval results [19]. Each query image is classified by performing a majority voting among the top k retrieved images. We create the reference database either by extracting RootSIFT features (1) from full images, (2) from the regions determined by our method, or (3) from the regions obtained by manual labeling.Table I states the average recall of the k-nn classifier over all logo classes. Note that the database also has non-logo images which may also affect the recall of a k-nn classifier. Our bounding boxes improve the performance compared to retrieval on full images and is slightly inferior to manual annotations.



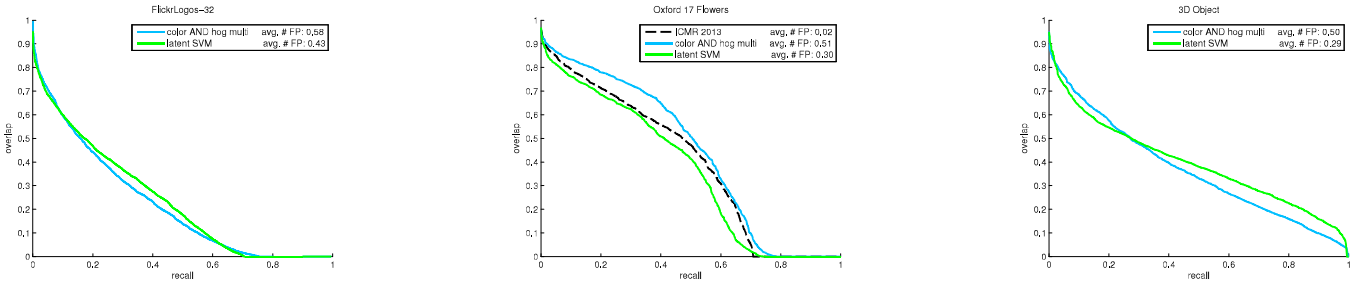
(a) Class "DHL" (b) Class "Aldi"

(c) Class "Coca Cola" (d) Class "Pepsi"

(e) Class "Shell" (f) Class "Esso"

Fig. 3: OR curves for six logo classes used in [1].

Alike the baseline approach we also use the Oxford 17 Flowers dataset [17]. For our initial color model we use the images of all other flower classes as negative set. For



(a) FlickrLogos-32 (b) Oxford 17 Flowers (c) 3D Object Categories

Fig. 4: OR curves for three datasets.

k	1	3	5	7	9
full images	0.82	0.81	0.79	0.78	0.77
our boxes	0.86	0.84	0.83	0.82	0.81
human annotations	0.87	0.85	0.85	0.84	0.84

TABLE I: Average recall of k-nn classification on retrieval results for FlickrLogos-32.

the HOG features and CCCP algorithm we use the negative set of the FlickrLogos-32 dataset. Our results are shown in figure 4b. Interestingly, the CCCP algorithm does not improve the results beyond the initial model and is even slightly below the baseline. Note, however, that the baseline uses a manually chosen aspect ratio. Also, HOG is not a suitable feature for flowers (let alone as BOW) and the requirement that the background is more diverse than the object is not often met.

For our final experiment we use the 3D Object Categories [18] dataset. Each class consists of ten different objects in front of different backgrounds. We pick a subset of 150 images with roughly similar viewpoints for each of the ten object classes. Our OR curve over all instances from all ten classes is shown in Fig. 4c. The CCCP algorithm improves the overlap with the ground truth, since unlike the Oxford Flowers dataset, the objects have more distinct gradients. An example result for the class "stapler" is given in Fig. 2e.

VII. Conclusion and Future Work

We have presented an approach to automatic object annotation from weak image labels. We initially estimate sets of positive features of two different types and estimate multiple bounding boxes per positive image. We then use a latent SVM learning algorithm (namely the CCCP algorithm) to refine the bounding box estimations based on BoW histograms. In our experiments we show that our initial model outperforms previous work and in many cases the CCCP algorithm further improves results.. Even though there is still room for improvement, our results are promising given the limited amount information we exploit. In future work we could consider additional features for our learning algorithm in order to exploit information complementary to color and HOG. Also, our approach still has a few empirical static constants which should be selected adaptively. Experiments show that we would obtain better results for some classes by using class-optimized parameters.

References

[1] C. X. Ries, F. Richter, and R. Lienhart, "Towards automatic object annotations from global image labels," in *ICMR 2013*, 2013, pp. 207–214.

[2] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *CVPR 2005*, 2005, pp. 886–893.

[3] A. L. Yuille and Anand Rangarajan, "The concave-convex procedure," *Neural Computation*, vol. 15, no. 4, pp. 915–936, 2003.

[4] C.-N. John Yu and T. Joachims, "Learning structural svms with latent variables," in *ICML 2009*, 2009, pp. 1169–1176.

[5] C.H. Lampert, M.B. Blaschko, and T. Hofmann, "Efficient subwindow search: A branch and bound framework for object localization," *IEEE PAMI*, vol. 31, no. 12, pp. 2129 –2142, Dec. 2009.

[6] K. Tang, S. Rahul, Y. Jay, and L. Fei-Fei, "Discriminative segment annotation in weakly labeled video," in *CVPR 2013*, 2013.

[7] P. Siva, C. Russell, and T. Xiang, "In defence of negative mining for annotating weakly labelled data," in *ECCV 2012*, 2012, pp. 594–608.

[8] P. Viola, J. C. Platt, and C. Zhang, "Multiple instance boosting for object detection," in *NIPS 18*. 2006, NIPS 18, pp. 1419–1426, MIT Press.

[9] O. Maron and A. L. Ratan, "Multiple-instance learning for natural scene classification," in *ICML 1998*, 1998, pp. 341–349.

[10] C.-Y. Chen and K. Grauman, "Watching unlabeled video helps learn new human actions from very few labeled snapshots," in *CVPR 2013*, 2013.

[11] B. Alexe, T. Deselaers, and V. Ferrari, "Measuring the objectness of image windows," *IEEE PAMI*, vol. 34, no. 11, pp. 2189–2202, 2012.

[12] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun, "Support vector machine learning for interdependent and structured output spaces," in *ICML 2004*, 2004, pp. 104–.

[13] M. J. Jones and J. M. Rehg, "Statistical color models with application to skin detection," *IJCV*, vol. 46, no. 1, pp. 81–96, Jan. 2002.

[14] T. Joachims, T. Finley, and C.-N. J. Yu, "Cutting-plane training of structural svms," *Machine Learning*, vol. 77, no. 1, pp. 27–59, Oct. 2009.

[15] T. Joachims, "Making large-scale svm learning practical. advances in kernel methods - support vector learning," pp. 169–184. MIT Press, 1999.

[16] S. Romberg, L. G. Pueyo, R. Lienhart, and R. van Zwol, "Scalable logo recognition in real-world images," in *ICMR 2011*, 2011, pp. 25:1–25:8.

[17] M-E. Nilsback and A. Zisserman, "A visual vocabulary for flower classification," in *CVPR 2011*, 2011, vol. 2, pp. 1447–1454.

[18] S. Savarese and L. Fei-Fei, "Generic object categorization, localization and pose estimation," in *ICCV 2007*, 2007.

[19] S. Romberg and R. Lienhart, "Bundle min-hashing," *IJMIR*, vol. 2, no. 4, pp. 243–259, 2013.

[20] L. Zhu, Y. Chen, A. L. Yuille, and W. T. Freeman. "Latent hierarchical structural learning for object detection," in *CVPR 2010.*, 2011, pp. 1062–1069.