INCREMENTAL LEARNING OF LATENT STRUCTURAL SVM FOR WEAKLY SUPERVISED IMAGE CLASSIFICATION

Thibaut Durand (1) *Nicolas Thome* (1) *Matthieu Cord* (1) *David Picard* (2)

(1) Sorbonne Universités, UPMC Univ Paris 06, UMR 7606, LIP6, F-75005, Paris, France

(2) ETIS/ENSEA, University of Cergy-Pontoise, CNRS, UMR 8051, France

ABSTRACT

Visual learning with weak supervision is a promising research area, since it offers the possibility to build large image datasets at reasonable cost. In this paper, we address the problem of weakly supervised object detection, where the goal is to predict the label of the image using object position as latent variable. We propose a new method that builds upon the Latent Structural SVM (LSSVM) formalism. Specifically, we introduce an original coarse-to-fine approach that limits the evolution of the latent parameter subspace. This incremental strategy drives the learning towards better solutions, providing a model with increased predictive accuracy. In addition, this leads to a significant speed up during learning and inference compared to standard sliding window methods. Experiments carried out on Mammal dataset validate the good performances and fast training of the method compared to state-of-the-art works.

Index Terms— Image Categorization, Weak Supervision, Object/Region Detectors, Latent SVM

1. CONTEXT

In image classification, the goal is to predict the semantic concept of an image according to its visual content. A major challenge is to fill the gap between low-level image descriptors and their semantic interpretation. One of the most successful image representation approaches is the Bag-of-Word (BoW) model [1], using SIFT features, and its extensions [2, 3, 4, 5, 6]. Another promising strategy is deep learning: recently, deep (convolutional) network [7], and biologically inspired model extensions [8, 9], show their ability to learn useful image representations for the classification task. Rather than using local descriptors, other methods use trained object and region detectors to represent the visual content of each image [10, 11], leading to compact and semantic signatures.

In this paper, we address the problem of visual learning with weak supervision. In this context, training data only provide image-level annotation (presence/absence of each category), and we model the (unknown) object location using latent variables. Learning weakly supervised object detectors is a very promising research area: if several millions of image-level annotated images are nowadays available, only thousands of accurate bounding box annotations exist [12].

In our context, handling weak supervision consists in learning a joint model for both localization and classification. A widely-used approach is the Latent SVM (LSVM) [13] and is extension to structured output: Latent Structural SVM LSSVM [14]. Despite the excellent performances for detection tasks, LSSVM performances for categorization [15, 16, 17, 18, 19, 20] are less impressive. We can point out two LSSVM limitations: the computation is very demanding and the optimization problem is hard (non-convex).

For modeling object positions, methods generally use a sliding window scheme. Regarding complexity, the evaluation of the classification function for each latent value (*e.g.,* object position) is the bottleneck for both learning and testing. A general algorithmic option to speed up training consists in using improved learning formulations, *e.g.,* cutting plane [21]. Another option is to limit the latent space exploration. In [18, 19], an incremental approach is proposed to gradually incorporate smaller and smaller regions during learning advance.

The second issue is the quality of the learned model caused by the non-convex optimization problem, potentially leading to (bad) local minima. To alleviate this problem, better learning algorithms have been proposed. In [15, 18, 19], different iterative methods based on curriculum learning [22] are introduced to find a better optimum. The basic idea is to start with easy samples, gradually adding more complex ones. The definition of easy *vs* hard examples is crucial: in [15], easy samples are defined as the ones that can be correctly predicted. In [18, 19] the size of the latent parameter space is used as an indication of the difficulty of the learning problem.

In this paper, we propose a novel method for weakly supervised image classification, where the evolution of the latent parameter space is done in a coarse-to-fine manner. The resulting algorithm, called Incremental LSSVM (ILSSVM), has two advantages compared to sliding window approaches: faster training and better predictive accuracy. Although our method shares some similarities with [18, 19], our incremental update is different, leading to further improvements. Our experiments validate the capacity of our method to outperform state-of-the-art works with a lower computation time.

2. INCREMENTAL LSSVM

We consider a multi-class problem, where input training data are composed of n labeled images $\{(x_1, y_1), \ldots, (x_n, y_n)\}$, where x_i is an image, and $y_i \in \mathcal{Y} = \{1, 2, \ldots, K\}$ its label. The latent variable $h_i = (h_{i1}, h_{i2}, h_{i3}, h_{i4})$ represents the bounding box of the predictive object location (see Fig.1).

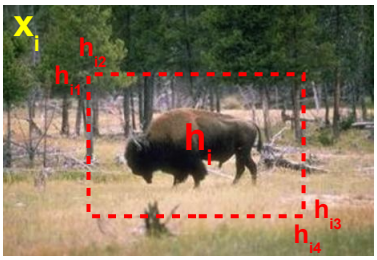


Fig. 1: Illustrative figure for latent variable model. x_i represents the i-th image with its bounding box h_i.

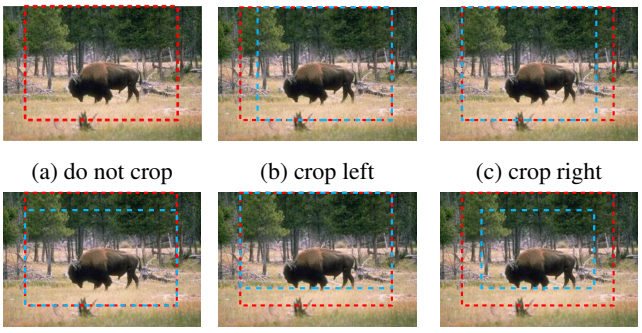
2.1. Incremental latent variable model

We propose an original evolution of the latent parameter subspace based on cropping, to explore only few boxes at each iteration. The idea is to start with the whole the image as bounding box, because we are sure that the object is inside, and gradually cropping it. The goal is to gradually center the box on the object. Hence, the latent optimization starts from a coarse model of the object and refines it at each iteration. At each iteration, the latent variable can take 6 values, which corresponds to : do not crop, crop to the left, right, top, bottom and 4 sides (illustration Fig. 2). The latent parameter subspace depends on the previous latent value, so each image can have a different subspace. For iteration t, the latent parameter subspace $\mathcal{H}_i$ for the image i is:

$$\begin{aligned}\mathcal{H}_i^t = \{&(h_{i1}^{t-1}, h_{i2}^{t-1}, h_{i3}^{t-1}, h_{i4}^{t-1}), (h_{i1}^{t-1}+k, h_{i2}^{t-1}, h_{i3}^{t-1}, h_{i4}^{t-1}), \\ &(\hat{h}_{i1}^{t-1}, h_{i2}^{t-1}+k, h_{i3}^{t-1}, h_{i4}^{t-1}), (h_{i1}^{t-1}, h_{i2}^{t-1}, h_{i3}^{t-1}-k, h_{i4}^{t-1}), \\ &(h_{i1}^{t-1}, h_{i2}^{t-1}, h_{i3}^{t-1}, h_{i4}^{t-1}-k), (h_{i1}^{t-1}+k, h_{i2}^{t-1}+k, h_{i3}^{t-1}-k, h_{i4}^{t-1}-k)\}\end{aligned} \quad (1)$$

where k is the crop step, and $h_i^{t-1} = (h_{i1}^{t-1}, h_{i2}^{t-1}, h_{i3}^{t-1}, h_{i4}^{t-1})$ is the predicted latent value at iteration $t-1$.

This model is interesting for two reasons: it is faster and have a better generalization than sliding window approaches. Using a small subspace allows to be faster in inference, because for multi-class classification, the inference time is proportional to the number of class and the dimension of the latent parameter space. At each iteration, ILSSVM explores only 6 windows per image, whereas a sliding window approach explores more than ten thousands of windows per image. To limit the dimension of the latent parameter space, [18] used a generic object detector to generate about 1500 boxes per image. Note that this requires the use of an external knowledge, and the computation time remains important.



(a) do not crop (b) crop left (c) crop right

(d) crop top (e) crop bottom (f) crop 4 sides

Fig. 2: Example of possible cropping (blue boxes) for a current bounding box (red)

Although an incremental learning is performed in [18, 19], a sliding window scheme is still used. As learning pursue, the latent variable space increases in [18, 19], whereas it remains constant with our approach. In addition, contrarily to all sliding window strategies in [15, 18, 19], ILSSVM does not require knowledge on the size and the ratio of objects. The boxes of each image can have a different size and a different ratio, and consequently adapt themselves to objects.

2.2. ILSSVM learning scheme formulation

We want to learn a LSSVM discriminant function of the form:

$$(y, h) = \underset{y \in \mathcal{Y}, h \in \mathcal{H}}{\arg\max} \langle w, \Psi(x_i, y, h) \rangle \quad (2)$$

where $\Psi(x, y, h)$ is the joint feature. $\Psi(x, y, h)$ is the image representation of the box defined by h for the image x and for the class y. In multi-class classification, the joint feature $\Psi(x, y, h) \in \mathbb{R}^{K \times d}$ is:

$$\Psi(x, y, h) = ([y = 1]\Phi(x, h), \ldots, [y = K]\Phi(x, h))$$

with $\Phi(x, h) \in \mathbb{R}^d$ is the image representation of the box h for the image x and $[y = \bar{y}] = \begin{cases} 1 & \text{if } y = \bar{y} \\ 0 & \text{otherwise} \end{cases}$

For training the discriminant function, we use a Latent Structural SVM formulation [14]. The objective function at the iteration t is:

$$\begin{aligned}\mathcal{P}_t(w) = &\underbrace{\frac{1}{2}\|w\|^2 + \frac{C}{n}\sum_{i=1}^{n}\left(\max_{(y,h) \in \mathcal{Y} \times \mathcal{H}_i^t} [\Delta(y_i, y) + \langle w, \Psi(x_i, y, h) \rangle]\right)}_{p(w)} \\ &\underbrace{- \frac{C}{n}\sum_{i=1}^{n} \max_{h \in \mathcal{H}_i^t} \langle w, \Psi(x_i, y_i, h) \rangle}_{q(w)}\end{aligned} \quad (3)$$

where C is the penalty parameter and $\Delta(y, \bar{y}) = [y \neq \bar{y}]$ is the loss function that penalizes misclassification. This objective function enforces the score of the ground truth class for

each image to be above the highest score of an incorrect score plus one. We do not require any ground truth localization information in this optimization.

Image representation. For image representation, we use the foreground-background feature representation introduced in [18]. We pool low-level features separately in the foreground and background to form the image-level representation. As reported in [18], it provides better classification performances than using foreground only, because background provides strong context for classification. To capture the spatial structure of the object, we use a spatial pyramid $1\times1, 3\times3$ for the foreground region. Each region is represented with a BoW models [1] using SIFT descriptors.

2.3. Optimization and classification

To solve the global optimization problem, we propose an iterative algorithm (Algorithm 1) that alternates between solving an LSSVM optimization problem with the current latent parameter subspace (line 4-8) and updating the latent variable subspace ($\mathcal{H}_i^t$) of each example (line 11).

Algorithm 1 ILSSVM Learning

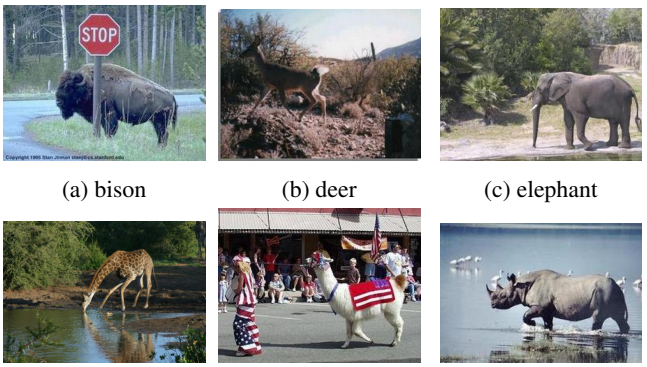
Require: training set $\{(x_i, y_i)\}_{i=1}^n$, $\{h_i^0\}_{i=1}^n$

1: Set $t = 1$ and initialize $\{\mathcal{H}_i^t\}_{i=1}^n$ with Eq. 1
2: **repeat**
3: Set $j = 0$ and $v_j = \frac{C}{n}\sum_{i=1}^n \Psi(x_i, y_i, h_i^t)$
4: **repeat**
5: Solve $w_j = \text{argmin}_w\,[p(w) - \langle w, v_j\rangle]$
6: $j \leftarrow j + 1$
7: Compute $v_j = \nabla_w q(w_j) = \frac{C}{n}\sum_{i=1}^n \Psi(x_i, y_i, h_i^j)$ where $h_i^j = \text{argmax}_{h\in\mathcal{H}_i^t}\langle w_j, \Psi(x_i, y_i, h)\rangle$
8: **until** $[p(w_j) - q(w_j)] - [p(w_{j-1}) - q(w_{j-1})] < \varepsilon$
9: $w_{t+1} = w_{j-1}$ and $\{h_i^{t+1}\}_{i=1}^n = \{h_i^{j-1}\}_{i=1}^n$
10: $t \leftarrow t + 1$
11: update $\{\mathcal{H}_i^t\}_{i=1}^n$ with Eq. 1
12: **until** objective function $\mathcal{P}_t(w)$ do not decrease

Ensure: w_t and $\{h_i^t\}_{i=1,\ldots,n}$

The main difficulty of this algorithm is to solve the LSSVM optimization problem. Unlike [18], we introduce a well formulated optimization based on concave-convex procedure (CCCP [23]) to solve our optimization problem (Eq. 3). At iteration t, the objective function $\mathcal{P}_t(w)$ is non-convex but can be written as the difference of convex functions: $p(w) - q(w)$. The CCCP algorithm is guaranteed to decrease the objective function at every iteration and to converge to a local minimum or saddle point. We first initialize latent variable and compute initial hyperplane v_0 (line 3). Then we alternate between solving the resulting convex problem (line 5) and linearizing the concave part ($-q$) at the current solution w_j (line 7).

To solve the optimization problem (line 5), we use the cutting-plane algorithm with "1-slack" LSSVM formulation [21]. Note that this much faster optimization scheme (time complexity linear in the number of training examples, and linear in the desired precision) is not used in [18, 19].



(a) bison (b) deer (c) elephant

(d) giraffe (e) llama (f) rhino

Fig. 3: Images of the different categories of Mammal dataset

Image classification. For classification, we use the same coarse-to-fine approach. We start with a box initialized on the whole image, and gradually crop it until convergence.

3. EXPERIMENTS

In this section, we describe our experimental setup and we show our results on the Mammal dataset [24].

3.1. Dataset and setup

The Mammal dataset [24] consists of 6 mammal categories : bison, deer, elephant, giraffe, llama and rhino. This dataset is challenging since there are few images per class, and the image resolutions are diverse. We split the images of each category into approximately 90% for training and 10% for testing, and use ten different splits to compare our method with the sliding window. The parameters are set to $C = 10^3$ and $\varepsilon = 10^{-3}$. For each experiment, we report the mean and the standard deviation of the test lost for the 10 splits. We use the same splits for all experiments. The local descriptors are SIFT, extract with `vl_dsift` of VLFeat [25] (step 2, size 4 pixels). To avoid having empty boxes with our method, we imposed a minimal area of 2000 pixels.

3.2. Results

To compare our method, we reimplemented an iterative multi-scale sliding window LSSVM learning scheme similar to [19]. This is a strong baseline since it is reported in [19] that this incremental learning favorably impacts both accuracy and computation time compared to standard sliding window LSSVM [15].

split	**SW** (6 scales)	**ILSSVM**
1	22,58	12,90
2	29,03	25,81
3	22,58	22,58
4	16,13	25,81
5	45,16	38,71
6	25,81	22,58
7	35,48	16,13
8	25,81	12,90
9	35,48	22,58
10	35,48	32,26
mean	$29,35 \pm 8,53$	$23,23 \pm 8,16$

Table 1: Classification error for the 10 splits for multi-scale sliding window (SW) and our method (ILSSVM)

Table 1 reports the results for ILSSVM and the multi-scale sliding window approach. ILSSVM brings a substantial gain of 6 pt. We performed a Student t-test and verify that this difference is statistically significant. This validates the fact that our incremental learning improves predictive accuracy. In addition, we want to stress that we also experiment a mono-scale sliding window scheme. We noticed extreme variations depending on the chosen window size: for "good" window (100×150 pixels, which is approximately the size of animals), classification error drops below 20%, whereas for "bad" windows (50×75 pixels), classification error can strongly increase ($\sim 45\%$). In [15], a mono-scale sliding window scheme is evaluated. However, the window size is setup to approximately match the object size. Therefore, the results reported in [15] are not comparable with ours because of the use of this strong prior knowledge. Also note that the results are slightly different from [15] because we do not use the same image representation.

Another important criterion of comparison is the computation time. We compare ILSSVM with sliding window approaches. Table 2 details the computation time for different methods. To learn the detectors for the 10 splits, 1 hour is needed for ILSSVM, whereas 30 hours are required for multi-scale sliding window with 6 scales and 1 ratio. It shows that our method is at least 30 time faster than multi-scale sliding window.

method	**time**
ILSSVM	1 h
one-scale sliding window	3 h
multi-scale sliding window (6 scales, 1 ratio)	30 h
multi-scale sliding window (6 scales, 6 ratios)	250 h

Table 2: Time Comparisons for the ten splits on 1 CPU

As a conclusion, ILLSVM performs better and faster than multi-scale sliding window methods.

3.3. Parameter of evolution of the latent variables

We study the influence of the crop step, which is the principal parameter of our method. To have a crop step adapted to every image, we use a step proportional to the maximum between width and height.

crop step (%)	13	17	20	25	30
error (%)	25,81	23,87	23,23	23,55	25,16

Table 3: Evolution of classification error with respect to the crop step

Table 3 details the results for different crop step values. The experiments show a small variation (less than 3%) of the classification error according to the crop step. Therefore, we can conclude that our method is robust to this parameter. In particular, it is less critical than the scale in sliding window methods. Using a small step is less efficient because the difference between boxes is not significant, and using a large step leads to model overfitting.

Fig. 4 show qualitative results of predicted boxes. The detection is coarse, but the boxes are centered on the objects. Moreover, it is worth mentioning that we use detection as intermediate step, while our ultimate goal is classification.

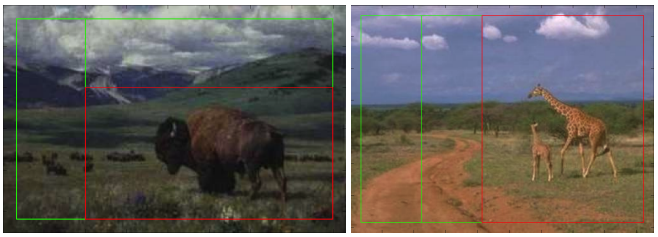


Fig. 4: Examples of predicted boxes for a step of 20% at different iterations. The green box is the initial box, and the red one, the final one.

4. CONCLUSION

We have presented an original coarse-to-fine approach for weakly supervised image classification based on Latent Structural SVM formulation. The key to our model is the small and incremental latent parameter space, which allows to find better optimum with small computation time. The evaluation on Mammal dataset shows that our method performs better and at least 30 time faster than multi-scale sliding window, and is robust to crop parameter. In the future, we plan to extend our approach for weakly supervised object detection, *i.e.* when the ultimate goal is an accurate object localization.

5. REFERENCES

[1] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," in *Proceedings of the International Conference on Computer Vision*, 2003.

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

[3] Florent Perronnin and Christopher R. Dance, "Fisher kernels on visual vocabularies for image categorization," in *CVPR*, 2007.

[4] Sandra Eliza Fontes de Avila, Nicolas Thome, Matthieu Cord, Eduardo Valle, and Arnaldo de Albuquerque Araújo, "Pooling in image representation: The visual codeword point of view," *Computer Vision and Image Understanding*, vol. 117, no. 5, pp. 453–465, 2013.

[5] D. Picard and P.H. Gosselin, "Efficient image signatures and similarities using tensor products of local descriptors," *CVIU*, vol. 117, pp. 680687, 2013.

[6] Hanlin Goh, Nicolas Thome, Matthieu Cord, and Joo-Hwee Lim, "Unsupervised and supervised visual codes with restricted boltzmann machines," in *Proceedings of the 12th European conference on Computer Vision - Volume Part V*, 2012, ECCV'12, pp. 298–311.

[7] Alex Krizhevsky, Ilya Sutskever, and Geoff Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*. 2012.

[8] Thomas Serre, Lior Wolf, Stanley Bileschi, Maximilian Riesenhuber, and Tomaso Poggio, "Robust object recognition with cortex-like mechanisms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 3, pp. 411–426, Mar. 2007.

[9] Christian Theriault, Nicolas Thome, and Matthieu Cord, "Hmax-s: Deep scale representation for biologically inspired image categorization," in *ICIP*, 2011.

[10] Eric P. Xing Li-Jia Li, Hao Su and Li Fei-Fei, "Object bank: A high-level image representation for scene classification & semantic feature sparsification," in *Neural Information Processing Systems (NIPS)*, Vancouver, Canada, December 2010.

[11] Thibaut Durand, Nicolas Thome, Matthieu Cord, and Sandra Eliza Fontes de Avila, "Image classification using object detectors," in *20th IEEE International Conference on Image Processing (ICIP)*, 2013.

[12] Matthew Blaschko, Pawan Kumar, and Ben Taskar, "Tutorial: Visual learning with weak supervision," http://www.centrale-ponts.fr/tutorials/cvpr2013/, CVPR 2013.

[13] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2010.

[14] Chun-Nam Yu and T. Joachims, "Learning structural svms with latent variables," in *International Conference on Machine Learning (ICML)*, 2009.

[15] P. Kumar, B. Packer, and D. Koller, "Self-paced learning for latent variable models," in *Advances in Neural Information Processing Systems (NIPS 2010)*, 2010.

[16] Megha Pandey and Svetlana Lazebnik, "Scene recognition and weakly supervised object localization with deformable part-based models," in *ICCV*, 2011.

[17] Sobhan Naderi Parizi, John G. Oberlin, and Pedro F. Felzenszwalb, "Reconfigurable models for scene recognition," in *CVPR*, 2012.

[18] Olga Russakovsky, Yuanqing Lin, Kai Yu, and Li Fei-Fei, "Object-centric spatial pooling for image classification," in *ECCV*, 2012.

[19] H. Bilen, V.P. Namboodiri, and L.J. Van Gool, "Object classification with latent window parameters," in *International Journal of Computer Vision*, 2013.

[20] Gaurav Sharma, Frederic Jurie, and Cordelia Schmid, "Expanded parts model for human attribute and action recognition in still images," *2013 IEEE Conference on Computer Vision and Pattern Recognition*, vol. 0, pp. 652–659, 2013.

[21] T. Joachims, T. Finley, and Chun-Nam Yu, "Cutting-plane training of structural svms," *Machine Learning*, vol. 77, no. 1, pp. 27–59, 2009.

[22] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston, "Curriculum learning," in *ICML*, 2009, p. 6.

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

[24] Geremy Heitz, Gal Elidan, Benjamin Packer, and Daphne Koller, "Shape-based object localization for descriptive classification," *Int. J. Comput. Vision*, vol. 84, no. 1, pp. 40–62, Aug. 2009.

[25] A. Vedaldi and B. Fulkerson, "VLFeat: An open and portable library of computer vision algorithms," http://www.vlfeat.org/, 2008.