MANTRA: Minimum Maximum Latent Structural SVM for Image Classification and Ranking

Thibaut Durand, Nicolas Thome, Matthieu Cord
Sorbonne Universités, UPMC Univ Paris 06, CNRS, LIP6 UMR 7606, 4 place Jussieu, 75005 Paris
{thibaut.durand, nicolas.thome, matthieu.cord}@lip6.fr

Abstract

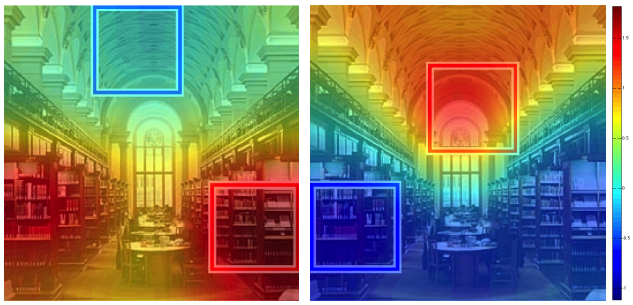
In this work, we propose a novel Weakly Supervised Learning (WSL) framework dedicated to learn discriminative part detectors from images annotated with a global label. Our WSL method encompasses three main contributions. Firstly, we introduce a new structured output latent variable model, Minimum mAximum lateNt sTRucturAl SVM (MANTRA), which prediction relies on a pair of latent variables: h^+ (resp. h^-) provides positive (resp. negative) evidence for a given output y. Secondly, we instantiate MANTRA for two different visual recognition tasks: multi-class classification and ranking. For ranking, we propose efficient solutions to exactly solve the inference and the loss-augmented problems. Finally, extensive experiments highlight the relevance of the proposed method: MANTRA outperforms state-of-the art results on five different datasets.

1. Introduction

Deep learning with Convolutional Neural Networks (CNN) [17] are becoming a key ingredient of visual recognition systems. Internal CNN representations trained on large scale datasets [17] currently provide state-of-the-art features for various tasks, *e.g.*, image classification or object detection [24, 10]. To overcome the limited invariance capacity of CNN, bounding box annotations are often used.

However, collecting full annotations for all images in a large dataset is an expensive task: whereas several millions of images annotated with a global label are nowadays available, while only thousands of accurate bounding box annotations exist [4]. This observation makes the development of Weakly Supervised Learning (WSL) models appealing.

Regarding WSL models, one of the most famous approach is the Deformable Part Model (DPM) [9], or its extension to structured output prediction, Latent Structural SVM (LSSVM) [38]. Recently, several attempts have been devoted to applying the LSSVM framework for object or scene recognition problems [18, 25, 29, 26, 3, 33, 32, 8].



a) $s_l(\mathbf{h}^+) = 1.8$; $s_l(\mathbf{h}^-) = 0.1$ b) $s_c(\mathbf{h}^+) = 1.5$; $s_c(\mathbf{h}^-) = -0.8$

Figure 1. MANTRA prediction maps for *library* classifier s_l a) and *cloister* classifier s_c b), for an image of class *library*. For each class, MANTRA score is $s(\mathbf{h}^+)+s(\mathbf{h}^-)$: $\mathbf{h}^+$ (red) provides localized evidence for the class, whereas $\mathbf{h}^-$ (blue) reveals its absence.

In this paper, we tackle the WSL problem of learning part detectors from images annotated with a global label. To this end, we introduce a novel structured output latent variable framework, MANTRA (Minimum mAximum lateNt sTRucturAl SVM), which incorporates a pair of latent variables $(\mathbf{h}^+, \mathbf{h}^-)$, and sum their prediction scores.

To illustrate the rationale of the approach, let us consider a multi-class classification instantiation of MANTRA, where latent variables $\mathbf{h}$ correspond to part localizations. $\mathbf{h}^+$ is the max scoring latent value for each class $\mathbf{y}$, *i.e.* the region which best represents class $\mathbf{y}$. $\mathbf{h}^-$ is the min scoring latent value, and can thus be regarded as an indicator of the absence of class $\mathbf{y}$ in the image.

To highlight the importance of the pair $(\mathbf{h}^+, \mathbf{h}^-)$, we show in Figure 1, for an image of the class *library*, classification scores for each latent location using (on the left) the *library* classifier s_l (the correct one) and (on the right) the *cloister* classifier s_c (a wrong one). $\mathbf{h}^+$ (resp. $\mathbf{h}^-$) regions are boxed in red (resp. blue). As we can see, the prediction score $s_l(\mathbf{h}^+) = 1.8$ for the correct *library* classifier is large, since the model finds strong local evidence $\mathbf{h}^+$ of its presence (bookcase), and no clear evidence of its absence (medium score $s_l(\mathbf{h}^-) = 0.1$). Contrarily, the prediction score for the *cloister* classifier s_c is substan-

tially smaller: although the model heavily fires on the vault ($s_c(\mathbf{h}^+) = 1.5$), it also finds clear evidence of the absence of *cloister*, here books ($s_c(\mathbf{h}^-) = -0.8$). As a consequence, MANTRA correctly predicts the class *library*[1].

From the intuition given in Figure 1, we provide in Section 3 a formal definition of MANTRA, and propose a generic and efficient optimization scheme to train it. In addition, we propose two instantiations of our model: multi-class classification (Section 4.1) and ranking, for which specific solutions must be designed to handle the large output space (Section 4.2). In the experiments (Section 5), we show that MANTRA trained upon deep features outperforms state-of-the-art performances on five different datasets. We now detail state-of-the-art methods which are the most connected to ours.

2. Related Works & Contributions

The computer vision community is currently witnessing a revolutionary change, essentially caused by Convolutional Neural Networks (CNN) and deep learning. CNN reached an outstanding success in the context of large scale image classification (ImageNet) [17], significantly outperforming handcrafted features based on Bag of Words (BoW) models [20, 27, 1, 11], or biologically-inspired networks [31, 35, 34]. Deep features also prove their efficiency for transfer learning: state-of-the-art performances on standard benchmarks (PASCAL VOC, 15-Scenes, MIT67, *etc*) are nowadays obtained with deep features as input. Recent studies reveal that performances can further be improved by collecting large datasets that are semantically closer to the target domain [42], or by fine-tuning the network with data augmentation [5].

Despite their excellent performances, current CNN architectures only carry limited invariance properties. Recently, attempts have been made to overcome this limitation. Some methods revisit the BoW model with deep features as local region activations [13, 12]. The drawback is that background regions are encoded into the final representation, decreasing its discriminative power. In [41], it is shown that aligning parts with poselet detectors makes human attribute recognition with deep features much more efficient. Although we share with [41] the motivation for part alignment, [41] focuses on specific pre-trained poselet detectors, which do not generalize to other tasks. In this paper, we tackle the problem of learning part detectors that are optimal for the task, in a weakly supervised manner.

The DPM model [9] is extremely popular for WSL, due to its excellent performances for weakly supervised object detection. Several attempts have been devoted to using DPM and its generalization to structured output prediction, LSSVM [38], for weakly supervised scene recognition and object localization. Some approaches learn single part detectors [18, 25, 29, 3], whereas other methods enrich the model with multiple parts [30, 26], optionally incorporating priors, *e.g.* sparsity or diversity, in order to learn sensible models [16, 33]. Due to the non-convexity of LSSVM objective, other approaches attempt to improve LSSVM training. In [18, 29, 3], different solutions are explored to apply the curriculum learning idea, *i.e.* how to find easy samples to incrementally solve the non-convex optimization problem and reach better local optima. However, it should be noted that all these methods still use the LSSVM prediction rule. We follow a different route with MANTRA, by proposing a new prediction function which combines a max and min scoring.

In this paper, we also tackle the important problem of learning to rank, since many computer vision tasks are evaluated with ranking metrics, *e.g.* Average Precision (AP) in PASCAL VOC. Optimizing ranking models with AP is challenging. In the fully supervised case, an elegant instantiation of structural SVM is introduced in [39], making it possible to optimize a convex upper bound over AP. On the contrary, few works tackle the problem of weakly supervised ranking from the latent structured output perspective, with the exception of [2]. In [2], the authors introduce LAPSVM, and point out that directly using LSSVM [38] for this purpose is not practical, mainly because no algorithm for solving the loss-augmented inference problem exists. LAPSVM introduces a tractable optimization by defining an ad-hoc prediction rule dedicated to ranking: first the latent variables are fixed, and then an optimal ranking with fixed latent variables is found. Our WSL method is applicable to any structured output space, and we show its relevance for weakly supervised AP ranking.

This paper presents a weakly supervised learning scheme, which encompasses the following contributions:

- We introduce a new latent structured output learning framework, MANTRA, which prediction function is based on a pair of latent variables $(\mathbf{h}^+, \mathbf{h}^-)$. In addition, we propose a direct cutting plane optimization procedure for training the model, which is efficient.
- We propose two instantiations of MANTRA: multi-class classification and ranking. We show that both inference and loss-augmented inference problems can be solved exactly and efficiently in the ranking case.
- We report excellent results in different visual recognition tasks: MANTRA outperforms state-of-the-art performances on five challenging visual datasets. In particular, we highlight that the model is able to detect parts witnessing the absence of a class, and show that the proposed ranking instantiation is able to further improve performances by a large margin.

[1] Many additional visualizations highlighting the relevance of training MANTRA with $(\mathbf{h}^+, \mathbf{h}^-)$ are shown in Supplementary, Figures 2,3,4.

3. Proposed Weakly Supervised Model

We present here the proposed WSL model: Minimum mAximum lateNt sTRucturAl SVM (MANTRA).

Notations We first give some basic notations used in the (latent) structured output learning framework. We consider an input space $\mathcal{X}$, that can be arbitrary, and a structured output space $\mathcal{Y}$. For $(\mathbf{x}, \mathbf{y}) \in (\mathcal{X} \times \mathcal{Y})$, we are interested in the problem of learning a discriminant function of the form: $f : \mathcal{X} \to \mathcal{Y}$. In order to incorporate hidden parameters that are not available at training time, we augment the description between an input/output pair with a latent variable $\mathbf{h} \in \mathcal{H}$. We assume that a joint feature map $\Psi(\mathbf{x}, \mathbf{y}, \mathbf{h}) \in \mathbb{R}^d$, describing the relation between input $\mathbf{x}$, output $\mathbf{y}$, and latent variable $\mathbf{h}$, is designed. Our goal is to learn a prediction function $f_{\mathbf{w}}$, parametrized by $\mathbf{w} \in \mathbb{R}^d$, so that the predicted output $\hat{\mathbf{y}}$ depends on $\langle \mathbf{w}, \Psi(\mathbf{x}, \mathbf{y}, \mathbf{h}) \rangle \in \mathbb{R}$. During training, we assume that we are given a set of N training pairs $(\mathbf{x}_i, \mathbf{y}_i) \in (\mathcal{X} \times \mathcal{Y})$, $i \in \{1; N\}$. Our goal is to optimize $\mathbf{w}$ in order to minimize a user-supplied loss function $\Delta(\mathbf{y}_i, \hat{\mathbf{y}})$ over the training set.

3.1. MANTRA Model

As mentioned in the introduction, the main intuition of the proposed MANTRA model is to equip each possible output $\mathbf{y} \in \mathcal{Y}$ with a pair of latent variables $(\mathbf{h}^+_{i,\mathbf{y}}, \mathbf{h}^-_{i,\mathbf{y}})$. $\mathbf{h}^+_{i,\mathbf{y}}$ (resp. $\mathbf{h}^-_{i,\mathbf{y}}$) corresponds to the max (resp. min) scoring latent value, for input $\mathbf{x}_i$ and output $\mathbf{y}$:

$$\mathbf{h}^+_{i,\mathbf{y}} = \arg\max_{\mathbf{h} \in \mathcal{H}} \langle \mathbf{w}, \Psi(\mathbf{x}_i, \mathbf{y}, \mathbf{h}) \rangle$$
$$\mathbf{h}^-_{i,\mathbf{y}} = \arg\min_{\mathbf{h} \in \mathcal{H}} \langle \mathbf{w}, \Psi(\mathbf{x}_i, \mathbf{y}, \mathbf{h}) \rangle$$

For an input/output pair $(\mathbf{x}_i, \mathbf{y})$, the scoring of the model, $D_{\mathbf{w}}(\mathbf{x}_i, \mathbf{y})$, sums $\mathbf{h}^+_{i,\mathbf{y}}$ and $\mathbf{h}^-_{i,\mathbf{y}}$ scores, as follows:

$$D_{\mathbf{w}}(\mathbf{x}_i, \mathbf{y}) = \langle \mathbf{w}, \Psi(\mathbf{x}_i, \mathbf{y}, \mathbf{h}^+_{i,\mathbf{y}}) \rangle + \langle \mathbf{w}, \Psi(\mathbf{x}_i, \mathbf{y}, \mathbf{h}^-_{i,\mathbf{y}}) \rangle \quad (1)$$

Finally, MANTRA prediction outputs $\hat{\mathbf{y}} = f_{\mathbf{w}}(\mathbf{x}_i)$ which maximizes $D_{\mathbf{w}}(\mathbf{x}_i, \mathbf{y})$ with respect to $\mathbf{y}$:

$$\hat{\mathbf{y}} = f_{\mathbf{w}}(\mathbf{x}_i) = \arg\max_{\mathbf{y} \in \mathcal{Y}} D_{\mathbf{w}}(\mathbf{x}_i, \mathbf{y}) \quad (2)$$

3.2. Learning Formulation

During training, we enforce the following constraints:

$$\forall \mathbf{y} \neq \mathbf{y}_i, \quad D_{\mathbf{w}}(\mathbf{x}_i, \mathbf{y}_i) \geq \Delta(\mathbf{y}_i, \mathbf{y}) + D_{\mathbf{w}}(\mathbf{x}_i, \mathbf{y}) \quad (3)$$

Each constraint in Eq. (3) requires the scoring value $D_{\mathbf{w}}(\mathbf{x}_i, \mathbf{y}_i)$ for the correct output $\mathbf{y}_i$ to be larger than the scoring value $D_{\mathbf{w}}(\mathbf{x}_i, \mathbf{y})$ for each incorrect output $\mathbf{y} \neq \mathbf{y}_i$, plus a margin of $\Delta(\mathbf{y}_i, \mathbf{y})$. $\Delta(\mathbf{y}_i, \mathbf{y})$, a user-specified loss, makes it possible to incorporate domain knowledge into the penalization. To give some insights of how the model parameters can be adjusted to fulfill constraints in Eq. (3), let us notice that:

- $D_{\mathbf{w}}(\mathbf{x}_i, \mathbf{y}_i)$, *i.e.* the score for the correct output $\mathbf{y}_i$, can be increased if we find statistically high scoring variables $\mathbf{h}^+_{i,\mathbf{y}_i}$, which represent strong evidence for the presence of $\mathbf{y}_i$, while enforcing $\mathbf{h}^-_{i,\mathbf{y}_i}$ variables not having large negative scores.

- $D_{\mathbf{w}}(\mathbf{x}_i, \mathbf{y})$, *i.e.* the score for an incorrect output $\mathbf{y}$, can be decreased if we find low scoring variables $\mathbf{h}^+_{i,\mathbf{y}}$, limiting evidence of the presence of $\mathbf{y}$, while seeking $\mathbf{h}^-_{i,\mathbf{y}}$ variables with large negatives scores, supporting the absence of output $\mathbf{y}$.

To allow some constraints in Eq. (3) to be violated, we introduce the following loss function:

$$\ell_{\mathbf{w}}(\mathbf{x}_i, \mathbf{y}_i) = \max_{\mathbf{y} \in \mathcal{Y}} \left[\Delta(\mathbf{y}_i, \mathbf{y}) + D_{\mathbf{w}}(\mathbf{x}_i, \mathbf{y}) - D_{\mathbf{w}}(\mathbf{x}_i, \mathbf{y}_i) \right] \quad (4)$$

We show in supplementary material A.1 that $\ell_{\mathbf{w}}(\mathbf{x}_i, \mathbf{y}_i)$ in Eq. (4) is an upper bound of $\Delta(\hat{\mathbf{y}}, \mathbf{y}_i)$.

Using the standard max margin regularization term $\|\mathbf{w}\|^2$, our primal objective function is defined as follows:

$$\mathcal{P}(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2 + \frac{C}{N} \sum_{i=1}^{N} \ell_{\mathbf{w}}(\mathbf{x}_i, \mathbf{y}_i) \quad (5)$$

3.3. Optimization

The problem in Eq. (5) is not convex with respect to $\mathbf{w}$. To solve it, we propose an efficient optimization scheme based on a cutting plane algorithm with the one-slack formulation [15]. Our objective function in Eq. (5) can thus be rewritten as follows:

$$\min_{\mathbf{w}, \xi} \frac{1}{2}\|\mathbf{w}\|^2 + C\xi \quad s.t. \quad \forall (\hat{\mathbf{y}}_1, \ldots, \hat{\mathbf{y}}_N) \in \mathcal{Y}^N \quad (6)$$
$$\frac{1}{N} \sum_{i=1}^{N} \Delta(\mathbf{y}_i, \hat{\mathbf{y}}_i) + D_{\mathbf{w}}(\mathbf{x}_i, \hat{\mathbf{y}}_i) - D_{\mathbf{w}}(\mathbf{x}_i, \mathbf{y}_i) \leq \xi$$

The key idea of the 1-slack formulation in Eq. (6) is to replace the N slack variables (weak constraints) by a single shared slack variable ξ (strong constraint). It is shown in [15] that this formulation helps speeding up the training of structural SVMs, reducing the complexity from being super-linear to linear in the number of training examples.

Cutting Plane Algorithm Based on the 1-slack formulation, we use a cutting plane strategy to optimize Eq. (6). Compared to sub-gradient methods, the cutting plane approach takes an optimal step in the current cutting plane model, leading to faster convergence [36].

For convex optimization problems, the idea of the cutting plane method is to build an accurate approximation, underestimating the objective function. However, it cannot be

directly applied for solving non-convex optimization problems, because the cutting plane approximation might not be underestimating the objective at all points, with the risk of missing good local minima [6]. Based on [6], we derive a non-convex cutting plane algorithm to solve Eq. (6). In particular, we use a method to detect and solve conflicts when adding a new cutting plane, as in [6], in order to avoid overestimating the objective function. It is important to stress that the proposed approach consists in a direct optimization, contrarily to iterative methods, which usually solve a set of approximate problems, *e.g.* CCCP [40].

The overall training scheme of MANTRA is shown in Algorithm 1. Starting from an initial cutting plane (Line 1), each cutting plane iteration consists in solving the resulting Quadratic Problem (QP) problem with the working set of cutting planes H (Line 5). As in [15], we solve the QP in the dual, because $|H|$ is generally much smaller than the input dimension. The dual formulation of Eq. (6) (Line 5) is derived in supplementary material A.2.1. Then, the current $\mathbf{w}$ solution (Line 7) is used to find the most violated constraint $\hat{\mathbf{y}}$ for each example (Line 10). The $\hat{\mathbf{y}}$'s are used to compute $g^{(t)}$ from the subgradient $\nabla_{\mathbf{w}}\ell_{\mathbf{w}}$ (Line 13), which computation is given in supplementary material A.2.2. $g^{(t)}$ serves to update the working set H at the next iteration. Finally, when adding a new cutting plane, we detect and solve conflicts (Line 15) using the method detailed in [6]. The algorithm stops as soon as no constraint can be found that is violated by more than the desired precision ε (Line 16).

Algorithm 1 Cutting Plane Algorithm for training MANTRA

Input: Training set $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1,\dots,N}$, precision ε, C.

1: Initialize $t \leftarrow 1$, $\{\hat{\mathbf{y}}_i, \mathbf{h}^+_{i,\hat{\mathbf{y}}_i}, \mathbf{h}^-_{i,\hat{\mathbf{y}}_i}\}_{i=1,\dots,N}$ and compute initial cutting plane $(g^{(1)}, c^{(1)})$
2: **repeat**
3: // Update working set and solve QP
4: $H \leftarrow (H_{ij})_{1\leq i,j\leq t}$ where $H_{ij} = \langle g^{(i)}, g^{(j)} \rangle$
5: $\alpha \leftarrow \arg\max_{\alpha \geq 0} \alpha^T c - \frac{1}{2}\alpha^T H \alpha \quad \text{s.t.} \quad \alpha^T 1 \leq C$
6: $\xi \leftarrow \frac{1}{C}(\alpha^T c - \alpha^T H \alpha)$
7: $\mathbf{w} \leftarrow \sum_{i=1}^{t} \alpha_i g^{(i)}$
8: $t \leftarrow t + 1$
9: **for** i=1 **to** N **do**
10: $\hat{\mathbf{y}}_i = \arg\max_{\mathbf{y}\in\mathcal{Y}} \ell_{\mathbf{w}}(\mathbf{x}_i, \mathbf{y}_i)$ // Loss-augmented inference
11: **end for**
12: // Compute new cutting plane and solve conflict
13: $g^{(t)} \leftarrow \frac{1}{N}\sum_{i=1}^{N} -\nabla_{\mathbf{w}}\ell_{\mathbf{w}}(\mathbf{x}_i, \mathbf{y}_i)$
14: $c^{(t)} \leftarrow \frac{1}{N}\sum_{i=1}^{N} \Delta(\mathbf{y}_i, \hat{\mathbf{y}}_i)$
15: $(g^{(t)}, c^{(t)}) \leftarrow SolveConflict(\mathbf{w}, g^{(t)}, c^{(t)})$
16: **until** $\langle \mathbf{w}, g^{(t)} \rangle \geq c^{(t)} - \xi - \varepsilon$

Output: $\mathbf{w}$

4. MANTRA Instantiation

MANTRA instantiation consists in specifying a particular joint feature Ψ and loss function Δ. For each instantiation, training the model requires solving two problems: inference (Eq. (2)), and loss-augmented inference (Eq. (7)):

$$\hat{\mathbf{y}} = \arg\max_{\mathbf{y}\in\mathcal{Y}} \Delta(\mathbf{y}_i, \mathbf{y}) + D_{\mathbf{w}}(\mathbf{x}_i, \mathbf{y}) \tag{7}$$

In this section, we instantiate MANTRA for two WSL detection tasks: multi-class classification and ranking.

4.1. Multi-class Instantiation

For multi-class classification, the input $\mathbf{x}$ is an image, and the latent variable $\mathbf{h}$ is the location of a region (bounding box) in the image. The output space is the set of classes $\mathcal{Y} = \{1, \dots, K\}$, where K is the number of classes. We use the standard joint feature map $\Psi(\mathbf{x}, \mathbf{y}, \mathbf{h}) = \{I(\mathbf{y} = 1)\Phi(\mathbf{x}, \mathbf{h}), \dots, I(\mathbf{y} = K)\Phi(\mathbf{x}, \mathbf{h})\}$, where $\Phi(\mathbf{x}, \mathbf{h}) \in \mathbb{R}^d$ is a vectorial representation of image $\mathbf{x}$ at location $\mathbf{h}$, and $I(\mathbf{y} = k) = 1$ if $\mathbf{y} = k$ and $I(\mathbf{y} = k) = 0$ if $\mathbf{y} \neq k$. $\Psi(\mathbf{x}, \mathbf{y}, \mathbf{h})$ is then a $(K \times d)$-dimensional vector. The loss function Δ is the $0/1$ loss. The inference and the loss-augmented inference are exhaustively solved.

4.2. Ranking Instantiation

Notations Following [39], our input for ranking is a set of N images x_i: $\mathbf{x} = \{x_i, i = 1, \dots, N\}$. During training, each image is given its class information: $x_i \in \mathcal{P}$ if it is labeled as positive, $x_i \in \mathcal{N}$ otherwise. The structured output is a ranking matrix $\mathbf{y}$ of size $N \times N$ providing an ordering of the training examples, such that (a) $y_{ij} = 1$ if $x_i \prec_{\mathbf{y}} x_j$[2]; (b) $y_{ij} = -1$ if $x_j \prec_{\mathbf{y}} x_i$; (c) $y_{ij} = 0$ if x_i and x_j are assigned the same rank. $\mathbf{y}^*$ is the ground-truth ranking matrix, *i.e.* $y^*_{ij} = 1$ for $(x_i, x_j) \in \mathcal{P} \times \mathcal{N}$, $y^*_{ii'} = 0$ and $y^*_{jj'} = 0$ for $(x_i, x_{i'}) \in \mathcal{P} \times \mathcal{P}$ and $(x_j, x_{j'}) \in \mathcal{N} \times \mathcal{N}$.

Joint Feature Map $\Psi(\mathbf{x}, \mathbf{y}, \mathbf{h})$ is defined as follows:

$$\Psi(\mathbf{x},\mathbf{y},\mathbf{h}) = \frac{1}{|\mathcal{P}||\mathcal{N}|} \sum_{x_i\in\mathcal{P}} \sum_{x_j\in\mathcal{N}} y_{ij}[\Phi(x_i, h_{i,j}) - \Phi(x_j, h_{j,i})] \tag{8}$$

The latent space $\mathcal{H}$ corresponds to the set of latent variables for each pair of positive-negative examples: $\mathbf{h} = \{(h_{i,j}, h_{j,i}) \in \mathcal{H}_i \times \mathcal{H}_j, (x_i, x_j) \in \mathcal{P} \times \mathcal{N}\}$, where $\mathcal{H}_i$ (resp. $\mathcal{H}_j$) is the set of locations in image x_i (resp. x_j). $\Phi(x_i, h_{i,j}) \in \mathbb{R}^d$ is thus a vectorial representation of image x_i at location $h_{i,j}$. Note that $\Psi(\mathbf{x}, \mathbf{y}, \mathbf{h})$ in Eq. (8) is a generalization of the feature map used in [2], where the selection of bounding boxes is specific to each image pair.

[2] *i.e.* x_i is ranked ahead of x_j.

Loss Function During training, the goal is to minimize a given ranking loss function. In this paper, we especially focus on AP, with $\Delta_{ap}(\mathbf{y}^*, \mathbf{y}) = 1 - AP(\mathbf{y}^*, \mathbf{y})$. As mentioned in Section 2, optimizing over Δ_{ap} is difficult, because Δ_{ap} does not decompose linearly in the examples [39]. In the WSL setting, the problem is exacerbated: for example, no efficient algorithm currently exists for solving the loss-augmented inference problem in the LSSVM case [38], as pointed out in [2].

We show here that inference and loss-augmented inference can be solved exactly and efficiently with MANTRA. Firstly, we show (Lemma 1) that in our ranking instantiation, $D_{\mathbf{w}}$ in Eq. (1) can be computed a standard fully supervised feature map. This result has major consequences, which enables to decouple the optimization over $\mathbf{y}$ and $\mathbf{h}$.

Lemma 1. *$\forall(\mathbf{x}, \mathbf{y})$, $D_{\mathbf{w}}(\mathbf{x}, \mathbf{y})$ in Eq. (1), for the ranking instantiation of Ψ given in Eq. (8), rewrites as $A(\mathbf{x}, \mathbf{y})$:*

$$A(\mathbf{x}, \mathbf{y}) = \frac{1}{|\mathcal{P}||\mathcal{N}|} \sum_{x_i \in \mathcal{P}} \sum_{x_j \in \mathcal{N}} y_{ij} (\langle \mathbf{w}, \Phi^+_-(x_i) \rangle - \langle \mathbf{w}, \Phi^+_-(x_j) \rangle$$

$$\langle \mathbf{w}, \Phi^+_-(x_i) \rangle = \max_{h \in \mathcal{H}_i} \langle \mathbf{w}, \Phi(x_i, h) \rangle + \min_{h \in \mathcal{H}_i} \langle \mathbf{w}, \Phi(x_i, h) \rangle$$

The proof of Lemma 1 is given in supplementary B.1, and comes from an elegant symmetrization of the problem due to the $\max + \min$ operation. The supervised feature map $\Phi^+_-(x_i)$ is the solution of the optimization over $\mathbf{h}$, whatever $\mathbf{y}$ value.

We now explain how inference and loss-augmented inference can be efficiently solved with MANTRA.

Proposition 1. *Inference for the MANTRA ranking instantiation is solved exactly by sorting the examples in descending order of score $s(i) = \langle \mathbf{w}, \Phi^+_-(x_i) \rangle$*

Proof. Since the inference consists in solving $\max_{\mathbf{y}} A(\mathbf{x}, \mathbf{y})$, this is a direct consequence of Lemma 1: the problem reduces to solving a fully supervised ranking inference problem, where each example x_i is represented by $\Phi^+_-(x_i)$. This is solved by sorting the example in descending order of score $s(i) = \langle \mathbf{w}, \Phi^+_-(x_i) \rangle$ [39]. □

Proposition 2. *Loss-augmented inference for MANTRA (Eq. (7)), with the instantiation of Eq. (8), is equivalent to:*

$$\hat{\mathbf{y}} = \arg\max_{\mathbf{y} \in \mathcal{Y}} \; [\Delta(\mathbf{y}^*, \mathbf{y}) + A(\mathbf{x}, \mathbf{y})] \tag{9}$$

Proposition 2 directly follows from Lemma 1. This is a key result, since it allows to use MANTRA with different loss functions, as soon as there is an algorithm to solve the loss-augmented inference in the fully supervised setting. To solve it with Δ_{ap}, we use the greedy algorithm proposed by [39], which finds a globally optimal solution (see complexity analysis in supplementary B.2). Note that it is possible to use faster methods [23] to address large-scale problem if required.

5. Experiments

In this section, we present an evaluation and analysis of MANTRA for multi-class classification and ranking tasks. In our implementation, we use MOSEK[3] to solve the Quadratic Problem (QP) at each cutting plane iteration (Line 5 of Algorithm 1 for MANTRA). The regularization parameter C is fixed to a large value, *e.g.* 10^5[4].

5.1. Multi-class Classification

In this section, we analyze our multi-class model (section 4.1) for different bounding box scales (from 30% to 90% of the image size, with a step of 10%).

Datasets We evaluate our multi-class model for 4 different visual recognition tasks: scene categorization [20] (15-Scene dataset), cluttered indoor scenes [28] (MIT 67 Indoor Scenes), fine-grained recognition [37] (People Playing Musical Instrument, PPMI) and complex event and activity images [21] (UIUC-Sports dataset). Performances are evaluated with multi-class accuracy and follow the standard protocol for all databases (more details in supplementary C.1).

Features Each image region is described using deep features computed with Caffe CNN library [14]. We use the output of the sixth layer (after the rectified linear unit transformation (ReLU)), so that each region is represented by a 4096-dimensional vector. For UIUC-Sports and PPMI (resp. 15 Scene and MIT67), we use deep features based on a model pre-trained on ImageNet (resp. Places [42]).

5.1.1 MANTRA Results

Firstly, we report MANTRA results with respect to the scale in Figure 2. It is worth pointing out that parts learned with a single region by MANTRA are able to improve performances over deep features computed on the whole image ($s = 100\%$), *e.g.* 5 pt for PPMI. It confirms that using regions allows to find more discriminant representations. We observe that the performances on small scales remain very good: for example, results for scale $s = 40\%$ are as good as for $s = 100\%$ in PPMI and UIUC; although performances slightly decrease for 15-Scene and MIT67, they remain very competitive (see Table 1).

The previous results suggest the idea of combining several scales, which are expected to convey complementary informations. To perform scale combination, we use an Object-Bank (OB) [22] strategy, which is often used in WSL works [30, 16, 33]. Our OB is simple, using max-pooling over P parts models and K classes, without SPM. Our final representation is thus compact ($P \times K$). Ultimately, we use a linear SVM classifier for classification.

[3] www.mosek.com

[4] MANTRA performances remain steady once C is sufficiently large.

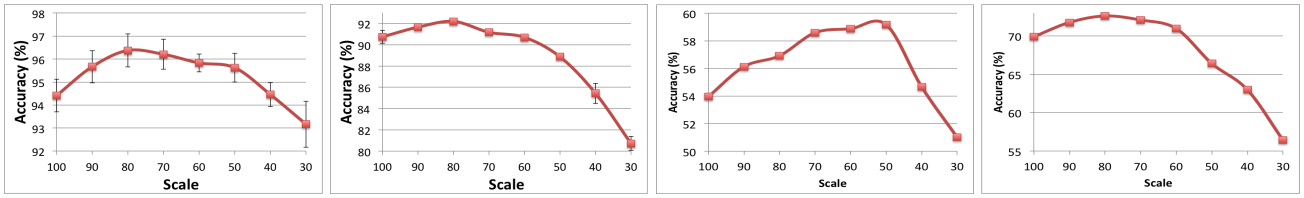


Figure 2. Predictive (multi-class) accuracy (%) (values are reported in Table 1 of supplementary material C.1)

Results for our multi-scale method are shown in Table 1: we can notice that performances improve compared to the best mono-scale results (4 pt for MIT67, 7 pt for PPMI), validating the fact that taking into account different scales enable catching complementary and discriminative localized information.

	PPMI	UIUC	15Sc	MIT67
Deep features				
ImageNet [14]	54.5*	94	88	58.5
Places [42]	38.6*	94.1	90.2	68.2
MOP-CNN [12]	-	-	-	68.9
Part-based				
SPM [20]	39.1	71.6	81.4	34.4
Object Bank [22]	-	77.9	80.9	37.6
RBoW [26]	-	-	78.6	37.9
DSS [32]	49.4	-	85.5	-
LPR [30]	-	86.3	85.8	44.8
IFV + BoP [16]	-	-	-	63.1
MLrep+IFV [7]	-	-	-	66.9
[33]	-	86.4	86.0	51.4
MANTRA	**66.2**	**97.3**	**93.4**	**76.6**

Table 1. Performances of MANTRA and comparison to state-of-the art works (* is our re-implementation).

We also compare MANTRA to state-of-the-art works in Table 1. We can notice that the improvement over part-based models, which use weaker features and essentially based on LSSVM [38], *e.g.* HoG, is huge. We also provide comparisons to recent methods based on deep features: we report performances with models pre-trained on ImageNet, but also using Places, a large-scale scene dataset recently introduced in [42]. As we can verify, Places is better-suited for scene recognition (performance boost in 15-Scene and MIT67), whereas ImagetNet has an edge over Places for object classification (PPMI). For UIUC, both models present similar performances. In Table 1, we can see that MANTRA can further improve performances over the best deep features (ImageNet or Places) by a large margin on the 4 databases, *e.g.* 8.5 pt on the challenging MIT67 dataset, or 11 pt on PPMI. As mentionned in Section 2, internal representations learned by ConvNets present limited invariance power: learning strong invariance is therefore challenging [41]. We show here that the proposed WSL scheme is able to efficiently learn strong invariance by aligning image regions, increasing performances when built upon strong deep features. MANTRA also significantly outperforms MOP-CNN [12] which uses VLAD pooling with deep features extracted at different scales. This shows the capacity of our model to seek discriminative part regions, whereas background and non-informative parts are incorporated into image representation in [12].

5.1.2 MANTRA Analysis

In this section, we provide an analysis of our method. We study the training time with respect to the number of regions, we show visual results, and compare MANTRA to LSSVM [38].

Time analysis The Figure 3 shows the training time required on 1 CPU (2.7 Ghz, 32 Go RAM) to train models on UIUC-Sports and MIT67. Results for 15 Scene and PPMI are reported in supplementary material C.1. Training MANTRA is fast: for example, it takes 1 minute at scale 30% of UIUC, where the training set is composed of 540 images and $\sim$ 36 000 regions. The training time increases linearly with respect to the number of regions per image. It is the expected behavior, because the most time consuming step of Algorithm 1 is the loss-augmented inference, which is proportional to the size of the latent space when it is solved exhaustively. This confirms that the proposed 1-slack cutting plane strategy to solve the optimization problem (section 3.3) is efficient.

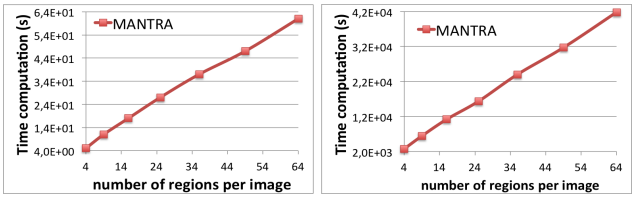


Figure 3. MANTRA training time (seconds) *vs* number of regions per image (values are reported in Tab 2 of supp. mat. C.1)

Comparison to LSSVM As previously mentioned, most of state-of-the-art WSL works are based on DPM [9] or LSSVM [38]. To highlight model differences between MANTRA and LSSVM, we carry out experiments with the same (deep) features, and evaluate performances on the same splits. For small scales, the choice of a proper region for classification is crucial. In Table 2, we report classification performances for both methods at scale 30%. Results clearly show the superiority of our model: MANTRA outperforms LSSVM by a very large margin, *e.g.* $\sim$ 30 pt increase on PPMI and MIT67. In Table 2, we also report the training time. MANTRA training is much faster than LSSVM's: for example, MANTRA is 30 times faster for UIUC Sports. The significant speedup for training MANTRA can be explained by the fact that LSSVM uses CCCP [40] to solve the non-convex optimization problem.

To further analyze the performance gain of MANTRA *vs* LSSVM, we isolate in Table 2 the impact of the new prediction function (section 3.1) by training MANTRA with CCCP (MANTRA-C), and the non-convex cutting-plane (NCCP) optimization (section 3.3), by training LSSVM with NCCP (LSSVM-N). CCCP leads to slightly better results for LSSVM, because the decomposition proposed by [38] exploits the structure of the optimization problem. In contrast, MANTRA objective (Eq. (5)) does not directly rewrites as a difference of convex (DC) functions. We can still use the generic DC decomposition of an arbitrary function f[5] (Theorem 1 of [40]) to use CCCP for MANTRA: results in Table 2 show that both optimizations give similar performances, because the decomposition is not driven by the structure of the problem, while MANTRA CCCP being significantly slower. The conclusion of this study is that the superiority of MANTRA *vs* LSSVM is due to the new prediction function.

	UIUC	15-Scene	PPMI	MIT67
Multi-class accuracy (%)				
LSSVM	73.3±0.3	65 ± 1.5	13.3	26.6
MANTRA	**93.2 ± 1**	**80.7±0.7**	**51.0**	**56.4**
LSSVM-N	71.6 ± 1.3	64.3±0.9	13.6	25.2
MANTRA-C	**93.2 ± 0.9**	80.4±0.6	50.9	**56.5**
Average training time (seconds)				
LSSVM	1863	14179	21327	156360
MANTRA	**61**	**843**	**2593**	**41805**

Table 2. Performances comparison and training time between MANTRA and LSSVM for scale 30%. MANTRA-C is MANTRA with CCCP, and LSSVM-N is LSSVM with NCCP.

Visual analysis We illustrate in Figure 4 the form of the regions corresponding to $\mathbf{h}_b^+$ and $\mathbf{h}_b^-$ templates for the *badminton* classifier in the UIUC-Sports dataset. On top row, we show regions for images of the *badminton* class: we can notice that $\mathbf{h}_b^+$ regions are semantically correlated to the badminton class, most often showing a person playing the game. It should be noted that $\mathbf{h}_b^-$ scores are positive, meaning that the model does not find strong evidence for the absence of the class. On bottom row, we show the parts $\mathbf{h}_b^-$ for *non-badminton* images. These regions focus on representative elements of outdoor scenes, whereas badminton is an indoor sport. In addition, $\mathbf{h}_b^-$ scores are negative and often below -0.5, clearly indicating the absence of the *badminton* class. Other visual results are shown in Supplementary C.1.

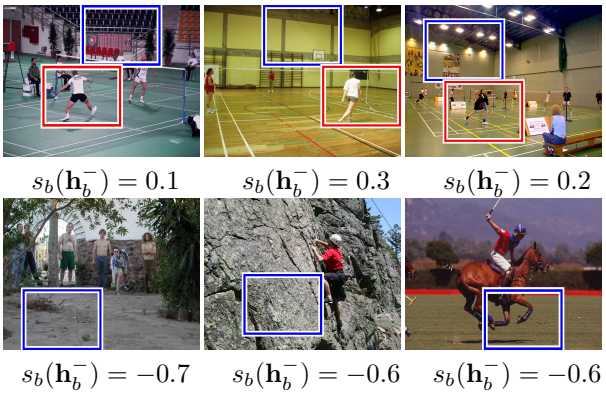


Figure 4. Examples of predicted regions for badminton classifier (s_b) on images from *badminton* class (top) and other classes (bottom). We also report the corresponding scores of each region $\mathbf{h}_b^-$.

5.2. Ranking

We evaluate our ranking model (section 4.2) for 2 different applications: action classification (VOC 2011), and object recognition (VOC 2007). The performances on the 2 datasets are evaluated with a ranking measure (MAP).

5.2.1 Action Classification

Setup The VOC 2011 Action Classification dataset includes 10 different action classes. We use standard ($\sim$2400-dim) poselets as region features [2]. We compare WSL models optimizing accuracy, *i.e.* LSSVM-Acc and MANTRA-Acc, and models explicitly optimizing AP, *i.e.* MANTRA-AP and LAPSVM [2]. Since the dataset contains Bounding Box (BB) annotations, we evaluate both ranking (MAP) and detection (average overlap between predicted and ground truth BB) performances. Experiments are carried out on the *trainval* set in a weakly supervised setup, *i.e.* without bounding box for training and testing, for 5 random splits (with 80% for training, 20% for testing).

Results As shown in Table 3, MANTRA-Acc outperforms LSSVM-Acc by $\sim$ 6 pt, again validating the

[5] $f(\mathbf{w}) = f_{vex}(\mathbf{w}) + f_{cave}(\mathbf{w}) = f(\mathbf{w}) + \lambda g(\mathbf{w}) - \lambda g(\mathbf{w})$ where g is a convex function, and λ a positive constant.

relevance of the new model introduced in this paper. MANTRA-Acc also performs similarly to LAPSVM [2], which is, to the best of our knowledge, the only method that optimizes an AP-based loss function over weakly supervised data. MANTRA-AP can further improve performances over MANTRA-Acc by 7 pt, which confirms the relevance of optimizing AP during training. T-test shows that MANTRA-AP is significantly better than LAPSVM with a risk of 0.1% (see supplementary C.2).

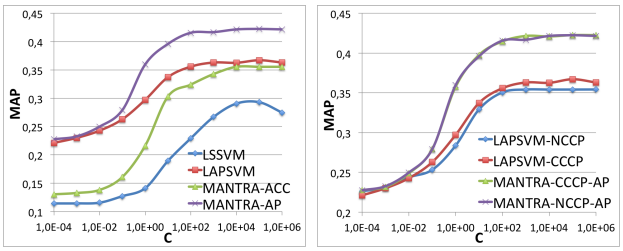
As we can see in Table 3, detection results are strongly correlated to ranking performances: MANTRA-AP also outperforms LAPSVM in terms of detection metric. T-test also reveals that the difference is significant with a risk of 0.1% (see supplementary C.2). Detection performances also give a quantitative validation that MANTRA is able to localize semantic parts, here people performing the action. We can interpret the use of the $\max + \min$ operation as a regularizer of the latent space, which exploits the capacity of $\mathbf{h}^-$ to witness the absence of a class to find more semantic part predictions $\mathbf{h}^+$.

Method	Ranking MAP (%)	Detect. ov. (%)
LSSVM-Acc	29.5 ± 1.3	12.7 ± 0.3
MANTRA-Acc	35.2 ± 1.2	18.9 ± 0.9
LAPSVM	36.7 ± 0.8	20.1 ± 0.7
MANTRA-AP	$\mathbf{42.2 \pm 1.3}$	$\mathbf{26.5 \pm 1.4}$

Table 3. Ranking and detection results on VOC 2011 Action. Performances per split are given in supplementary material C.2

Note that our protocol differs from [2], which evaluates on the test set and uses bounding box annotations. When using the same protocol as in [2], LAPSVM reaches 44.3% *vs* 47.1% for MANTRA-AP. Note that with this protocol, the prediction function used in test is the same for both models.

Impact of hyperparameter C: we show in Figure 5 performance variations *vs* C. We can observe that all methods reach optimal scores for large values: cross-validation on the train set always leads to $C = 10^4$ or 10^5 optimal values. We show in Figure 5b) the results of LAPSVM with NCCP and MANTRA-AP with CCCP. As in Table 2, the superiority of MANTRA is due to the prediction function.



a) impact of hyperparameter C b) optimization *vs* prediction function

Figure 5. Analysis of ranking performances

5.2.2 Object Recognition

Finally, we perform experiment on the VOC 2007 database, which is the most famous object recognition benchmark used in the last decade. We extract deep features pre-trained on ImageNet using MatConvNet library [5]. As in [5], we take the output of the seventh layer of `imagenet-vgg-m-2048`, after the ReLU. As done for the multi-class classification (section 5.1.1), we extract deep features at different scales, and combine them with ObjectBank (OB) [22] to have a multi-scale model. We compare our model instantiated for multi-class classification (MANTRA-Acc) and ranking (MANTRA-AP) to state-of-the-art results.

Results The performance obtained with deep features computed on the whole image is 77%, which is conform to what is reported in [5]. As shown in Table 4, MANTRA-Acc based on these features can improve performances by more than 5 pt, reaching 82.6%. MANTRA-AP further significantly improves performances by more than 3 pt, reaching 85.8%, again supporting the relevance of optimizing AP during training. Compared to recent methods, our model also outperforms [24] and SPP-net [13], which used a spatial pyramid pooling layer. To the best of our knowledge, the best published score on VOC 2007 is 82.4% [5], where fine-tuning with a ranking-based objective function is used. MANTRA-AP outperforms this method by more than 3 pt, without fine-tuning and data augmentation.

	[24]	[13]	[5]	MANTRA-Acc	MANTRA-AP
MAP(%)	77.7	80.1	82.4	82.6	**85.8**

Table 4. Ranking performances on VOC 2007.

6. Conclusion

This paper introduces a new latent structured output model, MANTRA, which prediction function is based on two latent variables $(\mathbf{h}^+, \mathbf{h}^-)$. The intuition behind $\mathbf{h}^-$ is as follows: for an incorrect output, it seeks negative evidence against it. For a correct output, it prevents from having large negative values for any region, thus $\mathbf{h}^-$ acts as a latent space regularizer exploiting contextual information. Another important contribution is the MANTRA ranking instantiation, for which efficient solutions are introduced to solve the challenging (loss-augmented) inference problem.

The experiments show that MANTRA outperforms state-of-the art performances on five datasets. We can especially point out the large improvement of the AP ranking optimization. Future works include adapting MANTRA for other structured visual applications, *e.g.* semantic segmentation or metric learning [19].

Acknowledgments This research was supported by a DGA-MRIS scholarship.

References

[1] S. Avila, N. Thome, M. Cord, E. Valle, and A. Araujo. Pooling in image representation: the visual codeword point of view. *Computer Vision and Image Understanding*, 2012. 2

[2] A. Behl, C. V. Jawahar, and M. P. Kumar. Optimizing average precision using weakly supervised data. In *CVPR*, 2014. 2, 4, 5, 7, 8

[3] H. Bilen, V. Namboodiri, and L. Van Gool. Object classification with latent window parameters. In *IJCV*, 2013. 1, 2

[4] M. Blaschko, P. Kumar, and B. Taskar. Tutorial: Visual learning with weak supervision, CVPR 2013. 1

[5] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *BMVC*, 2014. 2, 8

[6] T.-M.-T. Do and T. Artières. Regularized bundle methods for convex and non-convex risks. *JMLR*, 2012. 4

[7] C. Doersch, A. Gupta, and A. A. Efros. Mid-level visual element discovery as discriminative mode seeking. In *NIPS*, 2013. 6

[8] T. Durand, N. Thome, M. Cord, and D. Picard. Incremental learning of latent structural svm for weakly supervised image classification. In *ICIP*, 2014. 1

[9] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *PAMI*, 2010. 1, 2, 7

[10] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014. 1

[11] H. Goh, N. Thome, M. Cord, and J.-H. Lim. Learning Deep Hierarchical Visual Feature Coding. *IEEE Transactions on Neural Networks and Learning Systems*, 2014. 2

[12] Y. Gong, L. Wang, R. Guo, and S. Lazebnik. Multi-scale orderless pooling of deep convolutional activation features. In *ECCV*, 2014. 2, 6

[13] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *ECCV*, 2014. 2, 8

[14] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *ACM International Conference on Multimedia*, 2014. 5, 6

[15] T. Joachims, T. Finley, and C.-N. Yu. Cutting-plane training of structural svms. *Machine Learning*, 2009. 3, 4

[16] M. Juneja, A. Vedaldi, C. V. Jawahar, and A. Zisserman. Blocks that shout: Distinctive parts for scene classification. In *CVPR*, 2013. 2, 5, 6

[17] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*. 2012. 1, 2

[18] P. Kumar, B. Packer, and D. Koller. Self-paced learning for latent variable models. In *NIPS*, 2010. 1, 2

[19] M. T. Law, N. Thome, and M. Cord. Fantope regularization in metric learning. In *CVPR*, 2014. 8

[20] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006. 2, 5, 6

[21] L.-J. Li and F.-F. Li. What, where and who? classifying events by scene and object recognition. In *ICCV*, 2007. 5

[22] E. P. X. Li-Jia Li, Hao Su and L. Fei-Fei. Object bank: A high-level image representation for scene classification & semantic feature sparsification. In *NIPS*, 2010. 5, 6, 8

[23] P. Mohapatra, C. Jawahar, and M. P. Kumar. Efficient optimization for average precision svm. In *NIPS*. 2014. 5

[24] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *CVPR*, 2014. 1, 8

[25] M. Pandey and S. Lazebnik. Scene recognition and weakly supervised object localization with deformable part-based models. In *ICCV*, 2011. 1, 2

[26] S. N. Parizi, J. G. Oberlin, and P. F. Felzenszwalb. Reconfigurable models for scene recognition. In *CVPR*, 2012. 1, 2, 6

[27] F. Perronnin and C. R. Dance. Fisher kernels on visual vocabularies for image categorization. In *CVPR*, 2007. 2

[28] A. Quattoni and A. Torralba. Recognizing indoor scenes. In *CVPR*, 2009. 5

[29] O. Russakovsky, Y. Lin, K. Yu, and L. Fei-Fei. Object-centric spatial pooling for image classification. In *ECCV*, 2012. 1, 2

[30] F. Sadeghi and M. F. Tappen. Latent pyramidal regions for recognizing scenes. In *ECCV*, 2012. 2, 5, 6

[31] T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, and T. Poggio. Robust object recognition with cortex-like mechanisms. *PAMI*, 2007. 2

[32] G. Sharma, F. Jurie, and C. Schmid. Discriminative spatial saliency for image classification. In *CVPR*, 2012. 1, 6

[33] J. Sun and J. Ponce. Learning discriminative part detectors for image classification and cosegmentation. In *ICCV*, 2013. 1, 2, 5, 6

[34] C. Thériault, N. Thome, and M. Cord. Dynamic scene classification: Learning motion descriptors with slow features analysis. In *CVPR*, 2013. 2

[35] C. Thériault, N. Thome, and M. Cord. Extended Coding and Pooling in the HMAX Model. *IEEE Transactions on Image Processing (TIP)*, 2013. 2

[36] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *JMLR*, 2005. 3

[37] B. Yao and L. Fei-Fei. Grouplet: A structured image representation for recognizing human and object interactions. In *CVPR*, 2010. 5

[38] C.-N. Yu and T. Joachims. Learning structural svms with latent variables. In *ICML*, 2009. 1, 2, 5, 6, 7

[39] Y. Yue, T. Finley, F. Radlinski, and T. Joachims. A support vector method for optimizing average precision. In *SIGIR*, 2007. 2, 4, 5

[40] A. L. Yuille and A. Rangarajan. The concave-convex procedure. *Neural Computation*, 2003. 4, 7

[41] N. Zhang, M. Paluri, M. Ranzato, T. Darrell, and L. Bourdev. PANDA: Pose Aligned Networks for Deep Attribute Modeling. In *ECCV*, 2014. 2, 6

[42] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning Deep Features for Scene Recognition using Places Database. *NIPS*, 2014. 2, 5, 6