Exploiting Negative Evidence for Deep Latent Structured Models

Thibaut Durand, Nicolas Thome, Matthieu Cord

Exploiting Negative Evidence for Deep Latent Structured Models

Thibaut Durand, Nicolas Thome and Matthieu Cord

Abstract—The abundance of image-level labels and the lack of large scale detailed annotations (e.g. bounding boxes, segmentation masks) promotes the development of weakly supervised learning (WSL) models. In this work, we propose a novel framework for WSL of deep convolutional neural networks dedicated to learn localized features from global image-level annotations. The core of the approach is a new latent structured output model equipped with a pooling function which explicitly models negative evidence, e.g. a cow detector should strongly penalize the prediction of the bedroom class. We show that our model can be trained end-to-end for different visual recognition tasks: multi-class and multi-label classification, and also structured average precision (AP) ranking. Extensive experiments highlight the relevance of the proposed method: our model outperforms state-of-the art results on six datasets. We also show that our framework can be used to improve the performance of state-of-the-art deep models for large scale image classification on ImageNet. Finally, we evaluate our model for weakly supervised tasks: in particular, a direct adaptation for weakly supervised segmentation provides a very competitive model.

Index Terms—Weakly Supervised Learning, Convolutional Neural Networks, Structured Outputs, Image Classification, Ranking, Semantic Segmentation.

1 Introduction

Over the last few years, deep learning and Convolutional Neural Networks (ConvNets) [1] have become a key ingredient of visual recognition systems. They have been successfully applied to various visual recognition tasks, e.g. image classification [1], [2], [3], object detection [4], [5], [6], semantic segmentation [7], [8], [9]. Learning standard architectures (AlexNet [1], VGG16 [2], ResNet [3]) requires a huge number of training examples, which limits the number of potential training datasets. But it has been shown that these networks can be efficiently transfered on small datasets [10], [11]: the common practice is to use models pre-trained on large scale datasets, e.g. ImageNet [12], and to fine-tune them on the target domain.

As objects could be small and appear at different locations in the image, several frameworks [13], [14] rely on bounding boxes to train object-centric classifiers, and apply the classifiers by searching over different locations within the images. However, these rich annotations rapidly become costly to obtain and difficult to scale up [15], [16]. Another approach is to use Weakly Supervised Learning (WSL) models. The idea is to simultaneously learned a model to classify and localize objects, with only image-level labels. This is a challenging task, because the only available information for training is the presence / absence of the objects in the image [17], [18]. There is no information about the location or the size of the objects in the image. The aim of this paper is to learn localized representations for image classification with only image-level labels during training, indicating the presence or absence of a category. We call this problem weakly supervised because our model first localizes the discriminative regions, and then uses the predicted regions to perform image classification. We show that these localized representations can be used for object localization and semantic segmentation. We use the term WSL whatever the evaluated prediction.

Recently, several methods have been proposed for WSL of deep ConvNets with image-level labels [17], [18], [19], [20], [21], [22]. The key issue is to determine how to pool the regions to have a score per class. The output of the ConvNet is a detection map for each category, so to train it with standard classification loss, it is necessary to aggregate the maps into a global prediction for each class. This pooling issue is also present in WSL structured models [23], [24], [25], [26]. The most popular pooling is the `max` pooling [17], [19], [23], which selects the best region to perform prediction. In the case of binary classification, this pooling is an instantiation of the Multiple Instance Learning (MIL) paradigm [27]. In another way, some pooling strategies propose to use all regions to perform prediction, by marginalizing over the regions [18], [24], [25].

In this work, we propose a new approach to automatically learn localized features, with a pooling strategy explicitly encoding negative evidence. Our pooling function use both maximum and minimum scoring regions. The maximum regions seek discriminative regions for the class whereas the minimum regions seek regions indicating the absence of the class i.e. that provide a counter-evidence for the presence of a class. For instance, a cow detector should strongly penalize the prediction of the bedroom class. For a multi-label classification task, our model learn positive and negative correlation between object classes.

We propose several instantiations of our model for classification and structured Average Precision (AP) ranking. In particular, we show that our pooling function allows to solve exactly and efficiently both inference and loss-augmented inference problems in the AP ranking case, which is not the case with max pooling [28]. We apply our

- *E-mail: {thibaut.durand, nicolas.thome, matthieu.cord}@lip6.fr*

weakly supervised structured model to end-to-end training of deep ConvNets with weak supervision. We design a fully convolutional network architecture which enables fast region feature computation by convolutional sharing. Finally, we evaluate our model on different applications: classification, ranking, pointwise localization and segmentation. Our model outperforms previous state-of-the-art results on six classification datasets. We also perform an analysis of our model, including an ablation study.

Contributions. This paper extends two conference papers [29] and [30]. Our contributions are threefold. First, we introduce a unified framework for pooling, which generalizes standard WSL models as special cases, including our model [29] (Section 3.5). This study enables a better understanding of the difference between structured WSL models and their respective pooling functions, and is supplemented by a detailed experimental comparison in Section 6.3.4. Secondly, we propose a fully convolutional network based on ResNet-101 (Section 4.1) instead of VGG16, to learn feature maps with more accurate resolution than in [30], which is crucial for localization. Finally, we extend the experimental validation of [29], [30] at several levels. We perform experiments on the large scale ILSVRC 2012 classification dataset [12], showing a large improvement with respect to similar networks with the same number of parameters (Section 6.2). We also show that our model is able to learn localized features and can be successfully applied to weakly supervised object localization (Section 6.4) and semantic segmentation (Section 6.5).

The paper is organized as follows. In Section 2, we give an overview of the most relevant related work. Section 3 describes our weakly supervised structured model, which is based on negative evidence. We also provide a comparison between our model and existing weakly supervised structured models. In Section 4, we propose a fully convolutional network architecture, with a new prediction layer based on the weakly supervised structured model of Section 3. We detail the learning and instantiations for classification and ranking in Section 5. Section 6 presents the experimental studies.

2 State-of-the-art

Deep ConvNets. The computer vision community is currently witnessing a revolutionary change, essentially caused by ConvNets and deep learning. Beyond the outstanding success reached in the context of large scale classification (ImageNet [12] or Places [31]), deep features also prove to be very effecticient for transfer learning: state-of-the-art results on standard benchmarks are nowadays obtained with deep features [10]. Several studies reveal that performances can further be improved by collecting large datasets that are semantically closer to the target domain [31], or by fine-tuning the network [11].

Despite their excellent performances, current ConvNet architectures only carry limited invariance properties. [32] has shown that, although a small amount of shift invariance is built into the models through subsampling (pooling) layers, strong invariance is generally not dealt with. Recently, attempts have been made to overcome this limitation. Some methods revisit the BoW model with deep features as local region activations [33], [34] or design BoW layers [35]. The drawback of these models is that background regions are encoded into the final representation, decreasing its discriminative power. Another option to gain strong invariance is to explicitly align image regions, e.g. by using Weakly Supervised Learning (WSL) models.

2.1 Pooling scheme for WSL

Learning object detectors with image-level annotations is a common WSL problem. Most WSL methods are based on the Multiple Instance Learning (MIL) [27] paradigm: an image is regarded as a bag of instances (regions), and there is an asymmetric relationship between the bag and instance labels. A bag is positive if it contains at least one positive instance, and negative if all its instances are negative. MIL models thus perform image prediction through its max scoring region. The Latent SVM (LSVM) [36] is the most popular instantiation of MIL for computer vision, and its use in Deformable Part Model (DPM) [36] showed excellent performances for object detection. [17], [19], [21] use a spatial max pooling for WSL of deep ConvNets. Regarding non-convex optimization issues, a multi-fold MIL procedure is introduced in [37].

A limitation of the max pooling is related to its sensitivity to noise in the region scores, because it only uses the most discriminative region [20]. To increase robustness, several approaches propose to use several regions. The authors of [18] use the *global average pooling* (GAP), and show that this pooling can find all the discriminative regions of a category. [20], [22], [38] observe that this pooling have problems to identifying the extent of the object: the models trained with max pooling tend to underestimate object sizes, while the models trained with GAP overestimate them. [22] proposes a trade-off between max and average pooling by using a *log-sum-exp* pooling (LSE). Similarly, [38] introduces the *global weighted rank-pooling* (GWRP), where max pooling and GAP are special cases.

Recently, interesting MIL extensions have been introduced in [39], [40], [41], [42]. All these methods use a bag prediction strategy which departs from the standard max scoring function in MIL, especially due to the relaxation of the common Negative instances in Negative bags (NiN) MIL assumption. In the Learning with Label Proportion (LLP) framework [39], only label ratios between $\oplus/\ominus$ instances in the bags are provided during training. In [40], the LLP method of [39] is explicitly applied to MIL problems, in the context of video event detection. LLP outperforms baseline methods (mi/MI-SVM [43]), especially by its capacity to relax the NiN assumption. In [41], the authors question the NiN assumption by claiming that it is often violated in practice during image annotation: human rather label images based on their dominant concept than on the actual presence of the concept in each sub-region. To support the dominant concept annotation, the authors in [41] introduce a prediction function selecting the top scoring instances in each bag.

Other approaches depart from the NiN assumption by tracking the negative evidence of a class with regions [42], [44]. The main idea is to learn mutual exclusion constraints, model scene subcategories where the positive object class is unlikely to be found, or to capture specific parts which potentially indicate the presence of an object of a similar but distinct class. [44] proposes a generalization of LSVM by including negative latent variables. In [42], the authors introduce a WSL formulation specific to multi-class classification, where negative evidence is explicitly encoded by augmenting the model parameters to represent the positive/negative contribution of a part to a class.

In this paper, we incorporate the idea of negative evidence in the weakly supervised structured model. Our prediction function uses `max+min` region scores. The `min` scoring region accounts for the concept of negative evidence, and is capitalized on for learning a more robust model. In the experiments, we report that our region selection strategy outperforms approaches pooling over all regions.

2.2 Deep structured models & WSL

Deep structured models. Many problems in real-world applications (e.g. semantic segmentation, ranking) involve predicting a collection of random variables that are related to each other. Several approaches propose to combine Markov random fields (MRFs) with deep learning algorithms to estimate complex representations while taking into account the dependencies between the output random variables. In [45], the authors use a two step approach to combine ConvNets and fully-connected CRFs [46]. To jointly learn the ConvNet and CRF parameters, [47] reformulates the mean-field approximate inference for the fully-connected CRF as a Recurrent Neural Networks (RNN), and introduces CRF-RNN, a network that can be trained end-to-end. The authors show that joint learning of the ConvNet and the CRF parameters results in significant performance gains. At the same time, [48] proposes a general training algorithm to learn structured models, where MRF potentials are deep networks. [49] shows that for some kind of energy functions, proximal methods can be efficiently solved as a RNN feedfoward pass. Recently, [50] introduces the Structured Prediction Energy Networks (SPENs), where a deep architecture is used to define an energy function of candidate labels. [51] proposes direct loss minimization approach, and show its effectiveness for AP ranking problems.

WSL of deep structured models. The main idea of WSL is to model the missing information with latent/hidden variables. The most popular approach is the Latent Structural SVM (LSSVM) [23], which extends the Latent SVM [36]. The LSSVM model performs prediction by maximizing the joint posterior probability over the output and latent variables. The parameters of the model are learned by minimizing an upper bound on the empirical risk. One drawback of the LSSVM prediction is that the maximization over the latent variables is not robust to the inherent uncertainty of the latent variables. To address this issue, the Hidden Conditional Random Fields (HCRFs) [24] and the Marginal Structured SVM (MSSVM) [25] marginalize the latent variables to estimate the probability of the output variables. The ϵ-framework introduced in [26] proposes a trade-off between maximization and marginalization by using a *log-sum-exp* pooling. Other works explicitly model the uncertainty over the latent space [52], [53] and propose to predict the output variables by minimizing an entropy-based uncertainty measure. To put into perspective the differences between models and pooling functions, we introduce a general framework which includes LSSVM, HCRF, MSSVM, ϵ-framework and our model as special cases.

WSL ranking. 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. [54] shows that the learning objective should be tailored to the evaluation loss in order to obtain the best performance with respect to this loss. Optimizing ranking models with AP is challenging because the AP loss is non-differentiable and non-decomposable (it cannot be expressed as simple sums over the example labels). In the fully supervised case, an elegant instantiation of structural SVM is introduced in [55], making it possible to optimize a convex upper bound over the AP. On the contrary, few works tackle the problem of weakly supervised ranking from the latent structured output perspective, with the exception of [28]. In [28], the authors introduce LAPSVM, and point out that directly using LSSVM [23] 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. We show that our WSL model offers the ability to solve the loss-augmented inference with an elegant symmetrization due to the `max+min` prediction function.

WSL segmentation. We focus on segmentation models learned with only labels indicating the presence or absence of a class in the image. Many methods are based on the MIL framework: MIL-FCN [21] extends MIL to multi-class segmentation, while MIL-Base [20] generalizes MIL-FCN with a LSE pooling to speed up the convergence. EM-Adapt [56] includes an adaptive bias into the MIL framework, that boosts classes known to be present and suppresses all the others. Constrained CNN (CCNN) [57] uses a loss function optimized for any set of linear constraints on the output space of a ConvNet, to control the proportion of the labels of each class. Recently, [38] introduces a more complex network architecture, that is optimized with a combination of three losses.

3 Negative Evidence Model

We present here our latent structured model based on negative evidence. We begin by introducing the notations, then our prediction function, the learning formulation and the intuitions. Finally, we compare our model with exiting latent models.

3.1 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 define a scoring function $F_{\mathbf{w}}(\mathbf{x}, \mathbf{y}, \mathbf{h})$, with depends on the input data $\mathbf{x} \in \mathcal{X}$, the output $\mathbf{y} \in \mathcal{Y}$, the latent variable $\mathbf{h} \in \mathcal{H}$ and some parameters $\mathbf{w} \in \mathbb{R}^d$. Our goal is to learn a prediction function $f_{\mathbf{w}}$, parametrized by $\mathbf{w}$, so that the predicted output $\hat{\mathbf{y}}$ depends on $F_{\mathbf{w}}(\mathbf{x}, \mathbf{y}, \mathbf{h}) \in \mathbb{R}$. During training, we assume that we are given a set of N training pairs $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i^\star) \in \mathcal{X} \times \mathcal{Y} : i \in \{1, \ldots, N\}\}$, where $\mathbf{y}_i^\star$ is the ground-truth label of example i. Our goal is to optimize $\mathbf{w}$ in order to minimize a user-supplied loss function $\Delta(\mathbf{y}_i^\star, \mathbf{y})$ over the training set.

3.2 Negative Evidence Model

As mentioned in the introduction, the main intuition of our negative evidence 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. $\mathbf{h}_{i,\mathbf{y}}^-$) corresponding to the maximum (resp. minimum) scoring latent value, for input $\mathbf{x}_i$ and output $\mathbf{y}$:

$$\mathbf{h}_{i,\mathbf{y}}^+ = \arg\max_{\mathbf{h} \in \mathcal{H}} F_{\mathbf{w}}(\mathbf{x}_i, \mathbf{y}, \mathbf{h}) \tag{1}$$

$$\mathbf{h}_{i,\mathbf{y}}^- = \arg\min_{\mathbf{h} \in \mathcal{H}} F_{\mathbf{w}}(\mathbf{x}_i, \mathbf{y}, \mathbf{h}) \tag{2}$$

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

$$s_{\mathbf{w}}(\mathbf{x}_i, \mathbf{y}) = \frac{1}{2}\left(F_{\mathbf{w}}(\mathbf{x}_i, \mathbf{y}, \mathbf{h}_{i,\mathbf{y}}^+) + F_{\mathbf{w}}(\mathbf{x}_i, \mathbf{y}, \mathbf{h}_{i,\mathbf{y}}^-)\right) \tag{3}$$

Finally, our prediction is:

$$\hat{\mathbf{y}} = f_{\mathbf{w}}(\mathbf{x}_i) = \arg\max_{\mathbf{y} \in \mathcal{Y}} s_{\mathbf{w}}(\mathbf{x}_i, \mathbf{y}) \tag{4}$$

This maximization in Eq. (4) is known as the inference problem. Regarding the soring function in Eq. (3), we are here considering deep ConvNets models for $F_{\mathbf{w}}$. This generalizes the MANTRA model in [29], using a log-linear scoring function: $F_{\mathbf{w}}(\mathbf{x}, \mathbf{y}, \mathbf{h}) = \langle \mathbf{w}, \Psi(\mathbf{x}, \mathbf{y}, \mathbf{h}) \rangle$ where $\Psi(\mathbf{x}, \mathbf{y}, \mathbf{h})$ is a joint feature map that describes the relation between input $\mathbf{x}$, output $\mathbf{y}$, and latent variable $\mathbf{h}$.

3.3 Learning Formulation

During training, we enforce the following constraints:

$$\forall \mathbf{y} \neq \mathbf{y}_i^\star, \quad s_{\mathbf{w}}(\mathbf{x}_i, \mathbf{y}_i^\star) \geq \Delta(\mathbf{y}_i^\star, \mathbf{y}) + s_{\mathbf{w}}(\mathbf{x}_i, \mathbf{y}) \tag{5}$$

Each constraint in Eq. (5) requires the scoring value $s_{\mathbf{w}}(\mathbf{x}_i, \mathbf{y}_i^\star)$ for the correct output $\mathbf{y}_i^\star$ to be larger than the scoring value $s_{\mathbf{w}}(\mathbf{x}_i, \mathbf{y})$ for each incorrect output $\mathbf{y} \neq \mathbf{y}_i^\star$, plus a margin of $\Delta(\mathbf{y}_i^\star, \mathbf{y})$. $\Delta(\mathbf{y}_i^\star, \mathbf{y})$, 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. (5), let us notice that:

- $s_{\mathbf{w}}(\mathbf{x}_i, \mathbf{y}_i^\star)$, i.e. the score for the correct output $\mathbf{y}_i^\star$, can be increased if we find statistically high scoring variables $\mathbf{h}_{i,\mathbf{y}_i^\star}^+$, which represent strong evidence for the presence of $\mathbf{y}_i^\star$, while enforcing $\mathbf{h}_{i,\mathbf{y}_i^\star}^-$ variables not having large negative scores.
- $s_{\mathbf{w}}(\mathbf{x}_i, \mathbf{y})$, 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. (5) to be violated, we introduce the following loss function:

$$\ell_{\mathbf{w}}(\mathbf{x}_i, \mathbf{y}_i) = \max_{\mathbf{y} \in \mathcal{Y}} [\Delta(\mathbf{y}_i^\star, \mathbf{y}) + s_{\mathbf{w}}(\mathbf{x}_i, \mathbf{y}) - s_{\mathbf{w}}(\mathbf{x}_i, \mathbf{y}_i^\star)] \tag{6}$$

We show that $\ell_{\mathbf{w}}(\mathbf{x}_i, \mathbf{y}_i^\star)$ in Eq. (6) is an upper bound of $\Delta(\hat{\mathbf{y}}, \mathbf{y}_i^\star)$ in Appendix A.

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

$$\mathcal{P}(\mathbf{w}) = \frac{\lambda}{2}\|\mathbf{w}\|^2 + \frac{1}{N}\sum_{i=1}^{N} \ell_{\mathbf{w}}(\mathbf{x}_i, \mathbf{y}_i^\star) \tag{7}$$

where λ is the regularization parameter.

3.4 Negative Evidence Intuition

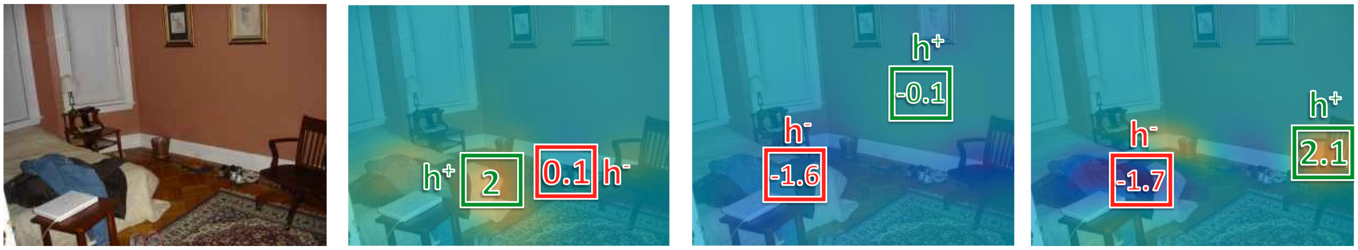
To illustrate the rationale of the approach, let us consider a multi-class classification instantiation of our negative evidence model, where $\mathbf{x}$ is the image, $\mathbf{y}$ is the label and the latent variables $\mathbf{h}$ correspond to region locations[1]. $\mathbf{h}^+$ is the max scoring 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 *bedroom* of MIT67 dataset [58], the heatmap representing the classification scores for each latent location using the *bedroom* classifier, the *airport inside* classifier and the *dining room* classifier. The $\mathbf{h}^+$ (resp. $\mathbf{h}^-$) regions are boxed in green (resp. red). As we can see, the prediction score for the correct class classifier (*bedroom*) is large, since the model finds strong local evidence $\mathbf{h}^+$ of it presence, and no clear evidence of its absence (medium score $F_{\mathbf{w}}(\mathbf{x}, \mathbf{y}{=}\textit{bedroom}, \mathbf{h}_{\mathbf{y}}^-) = 0.1$). For a wrong class very different like *airport inside*, the prediction score is very low, because there is not region similar to an airport. For a wrong class with similar objects like *dining room*, the maximum score for the *dining room* classifier is comparable with *bedroom* classifier: the model heavily fires on discriminative objects (bed for *bedroom* and chair for *dining room*). The prediction score $s_{\mathbf{w}}$ for the *dining room* classifier is significantly lower than for the *bedroom* classifier, because it also finds clear evidence of the absence of *dining room*, here bed ($F_{\mathbf{w}}(\mathbf{x}, \mathbf{y} = \textit{dining room}, \mathbf{h}_{\mathbf{y}}^-) = -1.7$). As a consequence, our negative evidence model correctly predicts the class *bedroom*. Another example on the fine-grained bird classification task is shown in Figure 3. In the Section 6.3.2, we analyze experimentally the selected regions for multi-label classification task and we show that our model can learn the positive and negative correlations between the classes.

3.5 Discussion

To put into perspective connections between negative evidence and existing latent structured models, we introduce

[1] This analysis can be extended to any problems where $\mathbf{h}$ is a part of $\mathbf{x}$, e.g. $\mathbf{h}$ is a paragraph of a text $\mathbf{x}$.



$\mathbf{y}^\star = bedroom$
original image

$\mathbf{y} = bedroom$
$s_{\mathbf{w}}(\mathbf{x},\mathbf{y}) = 2.1$
$F_{\mathbf{w}}(\mathbf{x},\mathbf{y},\mathbf{h}_{\mathbf{y}}^+) = 2$: high
$F_{\mathbf{w}}(\mathbf{x},\mathbf{y},\mathbf{h}_{\mathbf{y}}^-) = 0.1$: medium

$\mathbf{y} = airport\ inside$
$s_{\mathbf{w}}(\mathbf{x},\mathbf{y}) = -1.7$
$F_{\mathbf{w}}(\mathbf{x},\mathbf{y},\mathbf{h}_{\mathbf{y}}^+) = -0.1$: low
$F_{\mathbf{w}}(\mathbf{x},\mathbf{y},\mathbf{h}_{\mathbf{y}}^-) = -1.6$: low

$\mathbf{y} = dining\ room$
$s_{\mathbf{w}}(\mathbf{x},\mathbf{y}) = 0.4$
$F_{\mathbf{w}}(\mathbf{x},\mathbf{y},\mathbf{h}_{\mathbf{y}}^+) = 2.1$: high
$F_{\mathbf{w}}(\mathbf{x},\mathbf{y},\mathbf{h}_{\mathbf{y}}^-) = -1.7$: low

Fig. 1: Negative evidence intuition. The heatmaps and the predicted regions ($\mathbf{h}_{\mathbf{y}}^+$ in green, $\mathbf{h}_{\mathbf{y}}^-$ in red) for different learned class models (*bedroom*, *airport inside* and *dining room*) are shown on a *bedroom* image $\mathbf{x}$. $F_{\mathbf{w}}(\mathbf{x},\mathbf{y},\mathbf{h}_{\mathbf{y}}^+)$ (resp. $F_{\mathbf{w}}(\mathbf{x},\mathbf{y},\mathbf{h}_{\mathbf{y}}^-)$) is the score of the $\mathbf{h}_{\mathbf{y}}^+$ (resp. $\mathbf{h}_{\mathbf{y}}^-$) region for class $\mathbf{y}$, and $s_{\mathbf{w}}(\mathbf{x},\mathbf{y})$ is the predicted score for class $\mathbf{y}$. The *bedroom* and *dining room* models have high score for max regions because each model focus on objects discriminative for the class (bed for *bedroom* and chair for *dining room*). The min region brings complementary information to max region: the min regions score of *dining room* have a low score because the *dining room* model has found a negative evidence (bed) for the absence of *dining room* class.

the following generalized scoring function, with ”inverse temperature” β_h^+ and β_h^- parameters smoothing between max, softmax and average:

$$s_{\mathbf{w}}^{(\beta_h^+,\beta_h^-)}(\mathbf{x},\mathbf{y}) = \frac{1}{2\beta_h^+}\log\frac{1}{|\mathcal{H}|}\sum_{\mathbf{h}\in\mathcal{H}}\exp[\beta_h^+ F_{\mathbf{w}}(\mathbf{x},\mathbf{y},\mathbf{h})] + \frac{1}{2\beta_h^-}\log\frac{1}{|\mathcal{H}|}\sum_{\mathbf{h}\in\mathcal{H}}\exp[\beta_h^- F_{\mathbf{w}}(\mathbf{x},\mathbf{y},\mathbf{h})] \quad (8)$$

As shown in Table 1, the scoring function in Eq. (8) includes several existing models as special cases. When $\beta_h^+ = \beta_h^- \to +\infty$, it maximizes over latent variables and is equivalent to LSSVM [23] or max pooling for deep ConvNets [17]. When $\beta_h^+ = \beta_h^- = 1$, is is equivalent HCRF [24] or MSSVM [25], which marginalize over latent variables. GAP [18] ($\beta_h^+ = \beta_h^- \to 0$) also sum over latent variables, but unlike HCRF or MSSVM, all the latent variables have the same importance. The ϵ-framework [26] proposes a trade-off between max and average. This pooling strategy is also used to learn ConvNets [20], [22]. The prediction function is equivalent to our model when $\beta_h^+ \to +\infty$ and $\beta_h^- \to -\infty$, and pools over both maximum and minimum scores.

From the prediction function in Eq. (8), the conditional probability of output $\mathbf{y}$ given an input $\mathbf{x}$ can be defined as follows: $P(\mathbf{y}|\mathbf{x}) \propto \exp\left[\beta_{\mathbf{y}} \cdot s_{\mathbf{w}}^{(\beta_h^+,\beta_h^-)}(\mathbf{x},\mathbf{y})\right]$.

Model	β_h^+	β_h^-
HCRF [24] / MSSVM [25]	1	1
GAP [18]	$\to 0$	$\to 0$
LSSVM [23] / max [17]	$+\infty$	$+\infty$
Our model	$+\infty$	$-\infty$
ϵ-framework [26] / LSE [20], [22]	$\beta_h^+ = \beta_h^- \in (1,+\infty[$	

TABLE 1: Model comparison with corresponding parameters.

In Section 6.3.4, we provide a systematic comparison of the different pooling functions given in Table 1 to highlight their strengths and weaknesses in different contexts.

4 ResNet-WELDON Network Architecture

Based on the model presented in previous section, we propose ResNet-WELDON, a new weakly supervised learning dedicated to learn localized visual features by using only image-level labels during training. The proposed network architecture is decomposed into two sub-networks: a deep feature extraction network based on Fully Convolutional Network (FCN) and a prediction network, as illustrated in Figure 2. The feature extraction net purpose is to extract a fixed-size deep descriptor for each region in the image, while the prediction net outputs a structured output.

Notation. We note $F_{\mathbf{w}}^l(\mathbf{x},\mathbf{y},\mathbf{h})$ the output of the layer l at the location $\mathbf{h}$ of the feature map (or category) $\mathbf{y}$ for the input image $\mathbf{x}$. $\mathbf{w}$ are the parameters of the ConvNet.

4.1 Feature extraction network

The feature extraction network is dedicated to compute a fixed-size representation for any region of the input image. When using ConvNets as feature extractors, the most naive option is to process input regions independently, i.e. to resize each region to match the size of a full image for ConvNet architectures trained on large scale databases such as ImageNet (e.g. 224×224 for ResNet-101 [3]). This is the approach followed in R-CNN [4], or in MANTRA [29]. This is, however, highly inefficient since feature computation in (close) neighbor regions is not shared. Recent improvements in SPP nets [33], fully convolutional network (FCN) [7] or fast R-CNN [5] process images of any size by using only convolutional/pooling layers of ConvNets trained on ImageNet, subsequently applying max pooling to map each region into a fixed-size vector. They convolutionalize standard classification networks (AlexNet, VGG16) by replacing fully connected by convolution layers.

To have higher spatial resolution on the top of the network, we use the recently introduced the ResNet-101 [3], which is by design fully convolutional. ResNet-101 has 100 convolutional layers followed by global average pooling and fully-connected layer. To have feature map with spatial information, we remove the fully-connected layer (as usually done in the litterature), and the global average pooling,

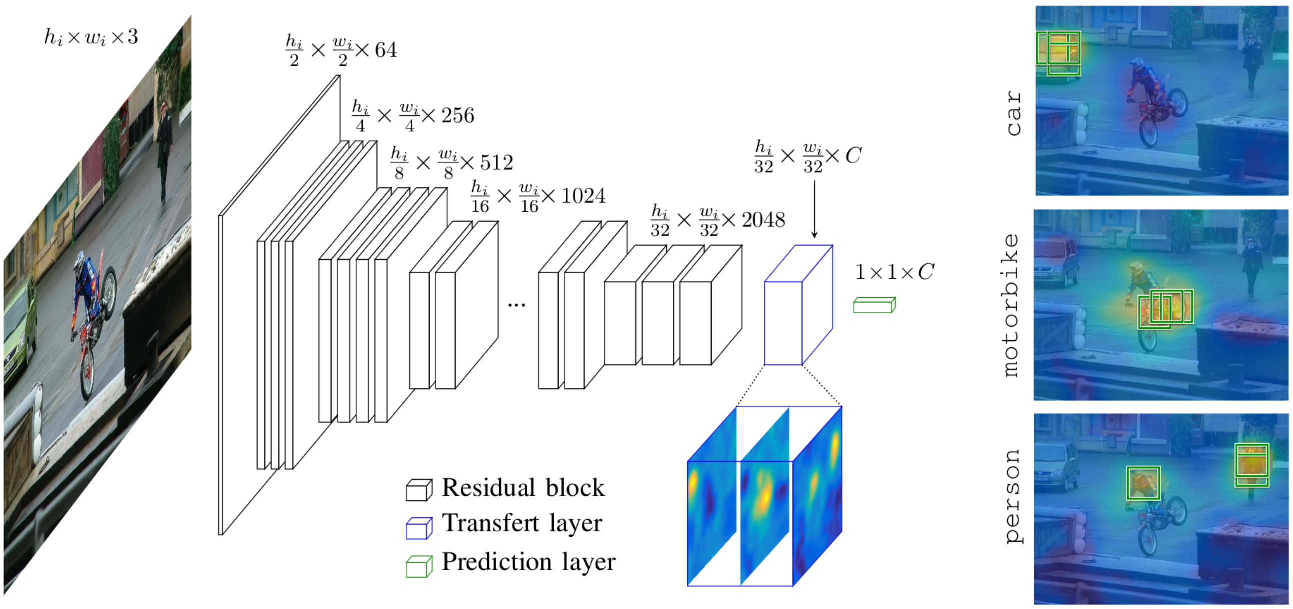

$h_i \times w_i \times 3$
$\frac{h_i}{2} \times \frac{w_i}{2} \times 64$
$\frac{h_i}{4} \times \frac{w_i}{4} \times 256$
$\frac{h_i}{8} \times \frac{w_i}{8} \times 512$
$\frac{h_i}{16} \times \frac{w_i}{16} \times 1024$
...
$\frac{h_i}{32} \times \frac{w_i}{32} \times 2048$
$\frac{h_i}{32} \times \frac{w_i}{32} \times C$
$1 \times 1 \times C$
Residual block
Transfert layer
Prediction layer
car
motorbike
person

Fig. 2: ResNet-WELDON deep architecture is decomposed into two sub-networks: a feature extraction network (left) and a prediction network (right) The feature extraction network is based on ResNet-101 to extract local features from whole images with good spatial resolution. Then a transfert layer is used to learn class-specific heatmaps (*car*, *motorbike* and *person*), and finally a prediction layer aggregates the heatmaps to produce a single score for each class. Finally, we show for each class the 3 regions with the highest score on the right.

which has not learnable parameter. The architecture with only the convolutional layers (and spatial pooling) allows to process images of arbitrary sizes, and the sharing of intermediate features over overlapping image regions. With this architecture, the spatial information is naturally preserved throughout the network: for an input image size of 224×224, the output size is 7×7. Spatial resolution impacts the localization and discriminability of the learned representations. We thus expect the resolution of the feature maps to be a key component for our model: finer maps keep more spatial resolution and lead to more specific regions. Moreover, ResNet is more effective at image classification while being parameter- and time-efficient than VGG16.

The input of the feature extraction network is an RGB image $h_i \times w_i$, where h_i (resp. w_i) is the height (resp. width). The output is a $h \times w \times 2048$ feature map, where $h = \frac{h_i}{32}$ and $w = \frac{w_i}{32}$ are number of sliding window positions in the horizontal and vertical direction in the image, respectively (see Figure 2). The weights of the feature extraction network are initialized on ImageNet.

4.2 Prediction network design

This part aims at selecting relevant regions to properly predict the global (structured) image label.

4.2.1 Transfer layer

The first layer of the prediction network is a transfert layer. Its goal is to transfer weights of the feature extraction network from large scale datasets to new target datasets. It transforms the output of the feature extraction network F^{fe} into a feature map F^t of size $h \times w \times C$, where C is the number of categories (see Figure 2). This layer is convolutional layer, composed of C filters, each of size $1 \times 1 \times 2048$. Due to the kernel size of the convolution, this layer preserves the spatial resolution of the feature maps. The output of this layer can be seen as localization heatmaps. In Figure 2, we show the heatmaps for differentes categories: *car*, *motorbike* and *person*.

4.2.2 Weakly-Supervised Prediction (WSP) layer

The second layer is a spatial pooling layer s aggregates the score maps into classification scores: for each output $\mathbf{y} \in \{1, \ldots, C\}$, the score over the $h \times w$ regions are aggregated into a single scalar value. We note $F^t_{\mathbf{w}}(\mathbf{x}_i, \mathbf{y}, \mathbf{h})$ is the score of region $\mathbf{h}$ from image $\mathbf{x}_i$ for category $\mathbf{y}$, and $\mathcal{H} = \{1, \ldots, r_i\}$ the region index set, and r_i is the number of regions for image $\mathbf{x}_i$. The output s of the prediction layer is a vector $1 \times 1 \times C$. As mentioned in Section 2, the standard approach for WSL inherited from MIL is to select the $\max$ scoring region. The output score is the score of the region with the maximum score. We propose to improve this strategy in two complementary directions: use negative evidence and several instances.

WELDON Pooling. This pooling improves max pooling by incorporate negative evidence. The prediction consists in summing the $\max$ and $\min$ scoring regions. Based on recent MIL insights on learning with top instances [41], we also propose to extend the selection of a single region to multiple regions. Formally, let $h_z \in \{0,1\}$ be the binary variable denoting the selection of the z^{th} region from layer F^t. We propose the scoring function s^{top}, which selects the k^+ highest scoring regions as follows:

$$s^{top}_{\mathbf{w},k^+}(F^t(\mathbf{x}_i, \mathbf{y})) = \frac{1}{k^+} \sum_{z=1}^{r_i} h^+_z F^t_{\mathbf{w}}(\mathbf{x}_i, \mathbf{y}, z) \tag{9}$$

$$\text{where } \mathbf{h}^+ = \underset{\mathbf{h} \in \{0,1\}^{r_i}}{\arg\max} \sum_{z=1}^{r_i} h_z F^t_{\mathbf{w}}(\mathbf{x}_i, \mathbf{y}, z) \;\; \text{s.t.} \; \sum_{z=1}^{r_i} h_z = k^+$$

where $F^t_{\mathbf{w}}(\mathbf{x}_i, \mathbf{y}, z)$ is the value of the z^{th} region score for class y. Beyond the relaxation of the NiN assumption, which is sometimes inappropriate (see Section 2), the intuition behind F^{top} is to provide a more robust region selection strategy. Indeed, using a single area for training the model necessarily increases the risk of selecting outliers.

To incorporate negative evidence in our prediction function, we propose the scoring function s^{low}, which selects the k^- lowest scoring regions as follows:

$$s^{low}_{\mathbf{w},k^-}(F^t(\mathbf{x}_i, \mathbf{y})) = \frac{1}{k^-}\sum_{z=1}^{r_i} h^-_z F^t_{\mathbf{w}}(\mathbf{x}_i, \mathbf{y}, z) \tag{10}$$

$$\text{where } \mathbf{h}^- = \underset{\mathbf{h}\in\{0,1\}^{r_i}}{\arg\min} \sum_{z=1}^{r_i} h_z F^t_{\mathbf{w}}(\mathbf{x}_i, \mathbf{y}, z) \;\text{ s.t. } \sum_{z=1}^{r_i} h_z = k^-$$

The final prediction simply consists in summing F^{top} and F^{low}:

$$s_{\mathbf{w}}(\mathbf{x}_i, \mathbf{y}) = \frac{1}{2}\left(s^{top}_{\mathbf{w},k^+}(F^t(\mathbf{x}_i, \mathbf{y})) + s^{low}_{\mathbf{w},k^-}(F^t(\mathbf{x}_i, \mathbf{y}))\right) \tag{11}$$

This prediction function is equivalent to MANTRA prediction function whenever $k^+ = k^- = 1$.

5 Learning & Instantiations

As shown in Figure 2, the WELDON model outputs $s \in \mathbb{R}^C$. This vector represents a structured output, which can be used in a multi-class or multi-label classification framework, but also in a ranking problem formulation.

5.1 Training formulation

In this paper, we consider three different structured prediction for WELDON, and their associated loss functions during training.

5.1.1 Classification

Multi-class classification. In this simple case, C is the number of classes and the output space is $\mathcal{Y} = \{1, \dots, C\}$. We use the usual softmax activation function on top of the spatial aggregation s. The probability of class $\mathbf{y}$ for image $\mathbf{x}$ is: $P(\mathbf{y}|\mathbf{x}) = \exp(s_{\mathbf{w}}(\mathbf{x}, \mathbf{y}))/\sum_{\mathbf{y}'\in\mathcal{Y}}\exp(s_{\mathbf{w}}(\mathbf{x}, \mathbf{y}'))$ with its corresponding log loss during training.

Multi-label classification. In the case of multiple labels, we use a one-against-all strategy, as [17]. For C different classes, we train the C binary classifiers jointly, using logistic regression for prediction $P(\mathbf{y}|\mathbf{x}) = (1+\exp(-s_{\mathbf{w}}(\mathbf{x}, \mathbf{y})))^{-1}$, with its associated log loss.

5.1.2 Ranking: Average Precision

We also tackle the problem of optimizing ranking metrics, and especially Average Precision (AP). We use a latent structured output ranking formulation, following [55]: our input is a set of N training images $\mathbf{x} = \{x_i : i \in 1, \dots, N\}$, with their binary labels y_i, and our goal is to predict a ranking matrix $\mathbf{y} \in \mathcal{Y}$ of size $N \times N$ providing an ordering of the training examples. Our ranking feature map for category c is expressed as:

$$F_{\mathbf{w}_c}(\mathbf{x}, \mathbf{y}, \mathbf{h}) = \sum_{p\in\mathcal{P}}\sum_{n\in\mathcal{N}} \mathbf{y}_{pn}(F^{fe}_{\mathbf{w}}(x_p, c, h^{pn}) - F^{fe}_{\mathbf{w}}(x_n, c, h^{np})) \tag{12}$$

where $\mathcal{P}$ (resp. $\mathcal{N}$) is the set of positive (resp. negative) examples. h^{pn} (resp. h^{np}) is a vector which represent the selected region for image x_p (resp x_n) when we consider the couple of image (p, n), and $\mathbf{h}$ is the set of selected regions for all pair of examples $(p, n) \in \mathcal{P} \times \mathcal{N}$

$$\mathbf{h} = \{(h^{pn}, h^{np}) \in \{0,1\}^{r_p} \times \{0,1\}^{r_n}, \tag{13}$$
$$\sum_{z=1}^{r_p} h^{pn}_z = k,\; \sum_{z=1}^{r_n} h^{np}_z = k,\; (p, n) \in \mathcal{P} \times \mathcal{N}\}$$

where r_p is the number of regions for image x_p (resp. x_n). $F^{fe}_{\mathbf{w}}(x_p, c, h^{pn})$ is the score for category c of region h^{pn} of image x_p

During training, we aim at minimizing the following loss: $\Delta_{ap}(\mathbf{y}^\star, \mathbf{y}) = 1 - AP(\mathbf{y}^\star, \mathbf{y})$, where $\mathbf{y}^\star$ is the ground-truth ranking. Since AP is non-smooth, we use the following surrogate (upper-bound) loss:

$$\ell_{\mathbf{w}}(\mathbf{x}, \mathbf{y}^\star) = \max_{\mathbf{y}\in\mathcal{Y}}\left[\Delta_{ap}(\mathbf{y}^\star, \mathbf{y}) + s_{\mathbf{w}}(\mathbf{x}, \mathbf{y})\right] - s_{\mathbf{w}}(\mathbf{x}, \mathbf{y}^\star) \tag{14}$$

The maximization in Eq (14) is generally referred to as Loss-Augmented Inference (LAI). Exhaustive maximization is intractable due to the huge size of the structured output space. The problem is even exacerbated in the WSL setting, see [28]. We exhibit here the following result for WELDON:

Proposition 1. *For each training example, let us denote $s(i) = s^{top}_{\mathbf{w},k}(F^t_{\mathbf{w}}(x_i, c)) + s^{low}_{\mathbf{w},k}(F^t_{\mathbf{w}}(x_i, c))$ in Eq (11). Inference and LAI for the WELDON ranking model can be solved exactly by sorting examples in descending order of score $s(i)$.*

The proof is given in Appendix B and comes from an elegant symmetrization of the problem due to the $\max + \min$ operation. Proposition 1 shows that the optimization over regions, i.e. score $s(i)$, decouples from the maximization over output variables $\mathbf{y}$. This reduces inference and LAI optimization to fully supervised problems. Inference solution directly corresponds to $s(i)$ sorting. It also allows to use our model 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 [55], which finds a globally optimal solution. Note that it is possible to use faster methods [59] to address large-scale problem if required.

5.2 Optimization

Our model is based on the architecture ResNet-101 [3]. We initialize it from a model pre-trained on ImageNet [12] and train it with Stochastic Gradient Descent (SGD) with momentum with image-level labels only. All the layers of the network are fine tuned. For multi-class and multi-label predictions, error gradients are well-known. For the ranking instantiation, we detail the gradient:

$$\frac{\partial \ell}{\partial \mathbf{w}} = \frac{\partial s_{\mathbf{w}}(\mathbf{x}, \tilde{\mathbf{y}})}{\partial \mathbf{w}} - \frac{\partial s_{\mathbf{w}}(\mathbf{x}, \mathbf{y}^\star)}{\partial \mathbf{w}}$$

where $\tilde{\mathbf{y}}$ is the LAI solution. When learning ResNet-WELDON, the gradients are backpropagated through the spatial pooling layer only within the selected regions, all other gradients being discarded. The selection of relevant regions for backpropagation is a key to learn precisely localized features without any spatial supervision [22].

6 Experiments

Our deep ConvNet architecture is based on ResNet-101. We evaluate our ResNet-WELDON strategy on several Computer Vision benchmarks corresponding to various visual recognition tasks. Absolute comparison with state-of-the-art methods is provided in Section 6.1, while Section 6.2 reports results on the large scale dataset ILSVRC. Section 6.3 analyzes the impact of the different improvements for training deep WSL ConvNets. Finally, we evaluate our model on the challenging weakly supervised segmentation application. The code is publicly available at https://github.com/durandtibo/weldon.resnet.pytorch.

Experimental Setup. In order to get results in very different recognition contexts, several datasets are used: object recognition (PASCAL VOC 2007 [60], PASCAL VOC 2012 [61], MS COCO [62]), scene categorization (MIT67 [58]), action recognition (VOC 2012 Action [61]) and fine-grained recognition (CUB-200 [63]). For MIT67, CUB-200, VOC 2007 and 2012, the performances are evaluated following the standard protocol. On CUB-200, we follow the standard protocol without the bounding boxes and part annotations. On VOC 2012, we used VOC evaluation server to evaluate. On MS COCO, we follow the protocol in [17] to perform classification experiments. On VOC 2012 Action, we use the same weakly supervised protocol as in [29], with evaluation on the *val* set. We also evaluate our model on the the *val* set of ILSVRC [12]. Table 2 summarizes dataset information.

Dataset	Train	Test	Classes	Eval.
VOC07	5,011	4,952	20	mAP
VOC12	11,540	10,991	20	mAP
VOCAction	2,296	2,292	10	mAP
COCO	82,783	40,504	80	mAP
MIT67	5,360	1,340	67	accuracy
CUB-200	5,994	5,794	200	accuracy
ILSVRC 2012	1,281,167	50,000	1000	accuracy

TABLE 2: Dataset information: number of train and test images, number of classes and evaluation measures (mAP: mean Average Precision).

6.1 State-of-the-art comparison

Firstly, we compare the proposed ResNet-WELDON model to state-of-the-art methods. We use different image size as input of our model, and scale combination is performed using an Object-Bank [64] strategy –as done in [30]. The image size and the values of k^+/k^- are given in Table 3. An analysis of the number of selected regions is done in Section 6.3.4, showing further improvements by careful tuning. Results are gathered in Table 4 and Table 5. In this section, we note ResNet-max (resp. ResNet-GAP) the special case of ResNet-WELDON where the spatial pooling is equivalent to the max pooling (resp. GAP).

Object datasets. We report in Table 4 the performances for object datasets, and we can see that ResNet-WELDON outperforms all recent methods based on deep features by a large margin. More specifically, the improvement compared to deep features computed on the whole image ([2], [3], [11]) is significant: there is an improvement over the best method [3] of 5.2 pt (resp. 4.2 and 9.8) on VOC 2007 (resp. VOC 2012 and MS COCO). Note that since our model used a ResNet-101 architecture, the performance gain directly measures the relevance of using a WSL method, which selects localized evidence for performing prediction, rather than relying on the whole image information. Both ResNet-101 and ResNet-WELDON have the same number of parameters. Compared to SPP net [33], the improvement of 12 pt on VOC 2007 highlights the superiority of region selection based on supervised information, rather than using handcrafted aggregation with spatial-pooling BoW models.

Image size	Size before pooling	k^+, k^-
224×224	7×7	5
280×280	9×9	10
320×320	10×10	20
374×374	12×12	30
448×448	14×14	50
560×560	18×18	75
747×747	24×24	100

TABLE 3: Multi-scale setup. We detail the input image sizes, along with the sizes of the feature maps before spatial pooling and the parameter values used in the spatial pooling.

Method	VOC07	VOC12	MSCOCO
Return Devil [11]	82.4	-	-
VGG16 [2]	89.3	89.0	-
SPP net [33]	82.4	-	-
NUS-HCP [65]	85.2	84.2	-
Nonlinear Embeddings [66]	86.1	-	-
ResNet-101 [3] *	89.8	89.2	72.5
DeepMIL [17]	-	86.3	62.8
MANTRA [29]	85.8	-	-
WELDON [30]	90.2	-	68.8
ProNet [22]	-	89.3	70.9
RRSVM [67]	92.9	-	-
SPLeaP [68]	88.0	-	-
ResNet-max	92.0	90.9	78.9
ResNet-WELDON	**95.0**	**93.4**	**80.7**

TABLE 4: mAP results on object recognition datasets. ResNet-WELDON and state-of-the-art methods results are reported. Half at the top shows the performances using global image representation, whereas the half at the bottom shows performances for models based on regions selection. * means that the results are obtained by fine-tuning the network on the dataset with the online code https://github.com/facebook/fb.resnet.torch.

The most important comparison is the improvement over other recent WSL methods on deep features [17], [22], [29], [30], [67]. We outperform the deep WSL ConvNet in [17], the approach which is the most closely connected to ours, by 7.1 pt (resp. 17.9) on VOC 2012 (resp. MS COCO). This big improvement illustrates the positive impact of incorporating MIL relaxations for WSL training of deep ConvNets, *i.e.* negative evidence scoring and top-instance selection. We also note a significant gain of 4.1 pt (resp 9.8) on VOC 2012 (resp. MS COCO) with ProNet [22], that relaxes the max pooling with a *log-sum-exp* pooling. Unlike our model, these models use a VGG16, but a fair comparison of the pooling of these methods with the same feature extraction network is done in Section 6.3.4. We also report the results of ResNet-max, where the only difference with respect to ResNet-WELDON is that the spatial pooling is a max pooling. We note that ResNet-WELDON is 2 or 3 pt better than ResNet-max on the three datasets. ResNet-

WELDON also outperforms by 2.1 pt the RRSVM [67] on VOC 2007, which learn a constrained aggregation operator on all the regions. Compared to [30], the improvement of 4.8 pt (resp. 11.9) on VOC 2007 (resp. MS COCO) essentially shows the importance of the FCN ResNet-101 that preserves spatial information throughout the network, and allows finer maps to learn more specific regions.

Method	CUB-200	MIT67	VOCAct
CaffeNet Places [31]	-	68.2	-
MOP CNN [34]	-	68.9	-
Nonlinear Embeddings [66]	63.1	-	-
Compact Bilinear Pooling [69]	84.0	76.2	-
ResNet-101 [3] *	72.5	78.0	77.9
Two-level attention [70]	69.7	-	-
STN [71]	84.1	-	-
MANTRA [29]	-	76.6	-
Negative parts [42]	-	77.1	-
MetaObject-CNN [72]	-	78.9	-
NAC [73]	81.0	-	-
GAP GoogLeNet [18]	63.0	66.6	-
WELDON [30]	-	78.0	75.0
Part-Stacked CNN [74] †	76.6	-	-
SPLeaP [68]	-	73.5	-
ResNet-max	76.1	68.3	79.9
ResNet-GAP	82.7	**85.3**	85.5
ResNet-WELDON	**85.6**	84.0	**86.4**

TABLE 5: Results on scene, action and fine-grained datasets. The performances on MIT67 and CUB-200 (resp. VOC 2012 Action) are evaluated with multi-class accuracy (resp. mAP). ResNet-WELDON and state-of-the-art methods results are reported. Half at the top shows the performances using global image representation, whereas the half at the bottom shows performances for models based on regions selection. † uses part-annotations during training.

Scene, action and fine-grained datasets. We also validate our model on scene, action and fine-grained classification. The results are reported in Table 5 and illustrate the big improvement of ResNet-WELDON compared to deep features computed on the whole image [2], [3], [31] and global image representation with deep features computed on image regions: MOP CNN [34] and Compact Bilinear Pooling [69] – these models use a VGG16. It is worth noticing that ResNet-WELDON also outperforms recent part-based methods [68], [72] including negative evidence during training [29], [42], but most of these models use a VGG16 architecture. This validates that our region selection approach is better than using all regions. ResNet-WELDON also significantly outperforms, on CUB-200 and MIT67, the recent GAP GoogLeNet [18], which used a global average pooling. On CUB-200, we can also note that our model is 9 pt better than Part-Stacked CNN [74], which uses bounding boxes and part annotations during training. This validates that our model can automatically find discriminative regions, even in the case of fine-grained classification. Finally, we report the results of ResNet-max and ResNet-GAP, which have the same architeture as ResNet-WELDON, except the spatial pooling. ResNet-WELDON and ResNet-GAP significantly outperform ResNet-max on the three datasets. We can note that the ResNet-GAP is better than ResNet-WELDON on MIT67. In Section 6.3.4, we show that using a lot of regions is important on MIT67 dataset.

6.2 Large-scale Image Classification

We also evaluate ResNet-WELDON on ILSVRC classification challenge [12] to show the scalability and the efficiency of our model for large-scale image classification. Table 6 summarizes the classification performances of ResNet-WELDON and existing models. To have a fair comparison between models, we only report results for single model. For our model, we use a mono-scale model with an input image size 448×448, and $k^+ = k^- = 50$.

We can see that ResNet-WELDON outperforms most of existing models trained using whole image (VGG16 [2], GoogleNet [75], ResNet-152 [3]) and regions (RRSVM [67], GoogleNet-GAP [18]). Our model have similar performances that ResNeXt-101 [76], which proposes a new residual block with a multi-branch architecture. ResNet-WELDON is slightly worse than Inception-ResNet-v2 (12 crops) [77] that combines both ResNet and Inception architectures. Better results can be obtained by learning ensemble of models.

We also reported results for different ResNets. The ResNet-101 is directly comparable to our model, because it corresponds to our initial model. It is important to note that with the same number of parameters and a very similar architecture, our model have a significant performance gain with respect to ResNet-101 (1 crop): +3.2 pt (resp. +2.0) on top-1 (resp. top-5) error. We also report the results with multi-crops post-processing, which a widely used post-processing to boost performances. Compared to our approach, multi-crops strategy extracts regions with a fixed grid, whereas our model automatically selects relevant regions. The important gain validates the relevance of our region selection approach. We also compare our model to deeper ResNet models: ResNet-WELDON is +0.9 pt (resp. +0.7) better than the deeper model ResNet-200, which have about the double of parameters. We can also note that ResNet-WELDON prediction is simple because it needs only 1 forward on the image to predict image label, whereas the multi-crops prediction needs several forwards on different image regions.

Model	Top-1 error	Top-5 error
VGG16 (144 crops) [2]	24.4	7.2
GoogleNet (144 crops) [75]	-	7.89
ResNet-152 (10 crops) [3]	21.43	5.71
RRSVM [67]	22.9	6.7
GoogleNet-GAP [18]	35.0	13.2
Inception-ResNet-v2 (12 crops) [77]	**18.7**	**4.1**
ResNeXt-101 (1 crop) [76]	19.1	4.4
ResNet-101 † (1 crop)	22.44	6.21
ResNet-101 † (10 crops)	21.08	5.35
ResNet-152 † (10 crops)	20.69	5.21
ResNet-200 † (10 crops)	20.15	4.93
ResNet-WELDON	19.21	4.23

TABLE 6: Classification error on the ILSVRC validation set with single model. ResNet-101 is our initial model. † is the results of pretrained model given at https://github.com/facebook/fb.resnet.torch.

6.3 ResNet-WELDON Analysis

In this section, we analyze our model. In Section 6.3.1, we analyze the impact of the different contributions with an ablation study. Then, we show some visual results in Section 6.3.2. In Section 6.3.3, we compare our ranking model with existing model. Finally, we analyze our pooling function and we compare it with standard pooling functions (Section 6.3.4).

6.3.1 Ablation study

We analyze the impact on prediction performances of the different contributions of ResNet-WELDON. Our baseline model a) is the WSL ConvNet model using an aggregation function s=`max` at the prediction layer (Figure 2). We present results for an input image of size 448×448, but similar behaviors are observed for other scales. It gives a network similar to [17], trained at a single scale. To measure the importance of the difference between ResNet-WELDON and a), we perform a systematic evaluation on the performance when the following variations are incorporated:

b) Fine-tuning (FT) the network on the target dataset.
c) Use of k top instances instead of the `max`. We use $k = 30$.
d) Incorporation of negative evidence through `max+min` aggregation function. When b)+c) are combined, we use k lowest-instances instead of the `min`, with $k = 30$.

a) max	b) FT	c) k=30	d) min	VOC07	VOCAct	MIT67	CUB-200
✓				86.8	71.8	62.4	66.0
✓	✓			91.3	77.9	65.4	72.8
✓	✓	✓		91.7	82.1	72,2	78.9
✓	✓		✓	92.2	82.4	69.6	78.2
✓	✓	✓	✓	93.7	85.4	77.2	82.4

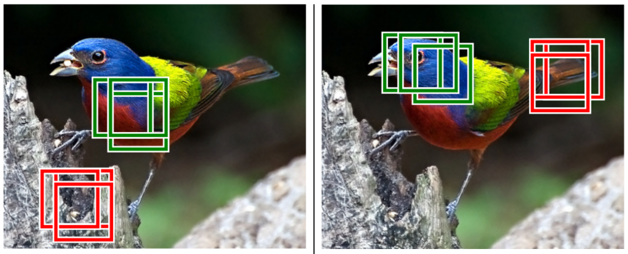
TABLE 7: Ablation study of our WSL deep ConvNet contributions on object (VOC 2007), action (VOC 2012 Action), scene (MIT67) and fine-grained (CUB-200) datasets.

The results are reported in Table 7 for object (VOC 2007), action (VOC 2012 action), scene (MIT67) and fine-grained (CUB-200) datasets. From this systematic evaluation, we can draw the following conclusions:

1) The fine-tuning (b)) significantly impacts performances, with +4.5 pt (resp. +6.1, +3.0 and +6.8) gain on VOC 2007 (resp. VOC 2012 Action, MIT67 and CUB-200). It validates that jointly updating all network parameters is crucial, in particular for fine-grained datasets.
2) Both c) and d) improvements result in a very large performance gain on all datasets w.r.t the baseline `max` pooling with fine-tuning. But the trends change according to the datasets. d) leads to a better improvement than c) on VOC 2007 and VOC 2012 Action datasets: d) has a gain of +0.9 pt (resp. +4.5) whereas c) has a gain of +0.3 pt (resp. +4.2) on VOC 2007 (resp. VOC 2012 Action). On the contrary, on MIT67 and CUB-200, c) leads to a better improvement than d): c) has a gain of +6.8 pt (resp. +6.1) whereas c) has a gain of +4.2 pt (resp. +5.4) on MIT67 (resp. CUB-200).
3) Combining c) and d) improvements further boost performances with respect to the best of c) or d): +5 pt on MIT67, +3.5 pt on CUB-200, +1.5 pt on VOC 2007 and 3 pt on VOC 2012 Action. This shows the complementarity of these two extensions at the aggregation level. This confirms that our pooling with several instances and negative evidence is relevant.

6.3.2 Visual results

To illustrate the region selection policy performed by ResNet-WELDON, we show in Figure 3 the top 3 positive (resp. top 3 negative) regions selected by the model in green (resp. red), on the CUB-200 dataset. We show the results for two similar bird species, where the main difference is that the *indigo bunting* is completely blue, whereas the *painted bunting* is multicolor. The *painted bunting* model has high scores for all regions, because all regions are correlated with the class. On the contrary, the *indigo bunting* model has high score on the head – because the head is blue – but very low score on the tail because it is not blue. The red-black tail is a clear evidence of the absence of *indigo bunting* class.



painted bunting (1.9)
top-3=1.8 low-3=0.1

indigo bunting (0.2)
top-3=1.9 low-3=-1.7

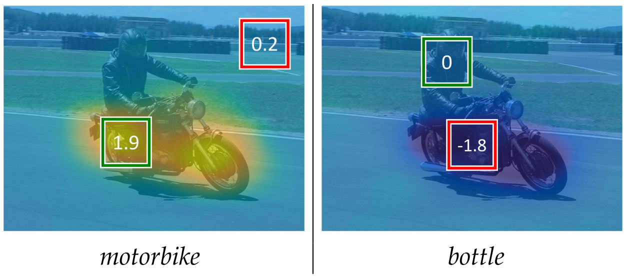
Fig. 3: Visual results of ResNet-WELDON with the final prediction score. We visualize the predicted regions for the ground truth model (left column) and a model of a similar incorrect class (right column). The top 3 positive (resp. top 3 negative) regions selected by the model are in green (resp. red).

In Figure 4, we show the heatmaps of 2 classes: *motorbike* which is present in the image and *bottle* which is absent. The *motorbike* model outputs high scores for the motorbike regions and medium scores for the other regions including the person regions. The *motorbike* model learns that the *person* class is positively correlated with the *motorbike* class. This positive correlation is confirmed by the co-occurrence matrix on VOC 2007 shown in Figure 5: 70% of the *motorbike* images also contain a *person*. On the contrary, we observe that *motorbike* regions have very low scores for *bottle* model and act as negative evidence for the *bottle* class, because the model learns that *motorbike* never occurs with a *bottle*. This negative correlation is confirmed by the co-occurrence matrix. An interesting behavior of the negative evidence model is that it can model positive and negative correlation between object classes.

6.3.3 Ranking analysis

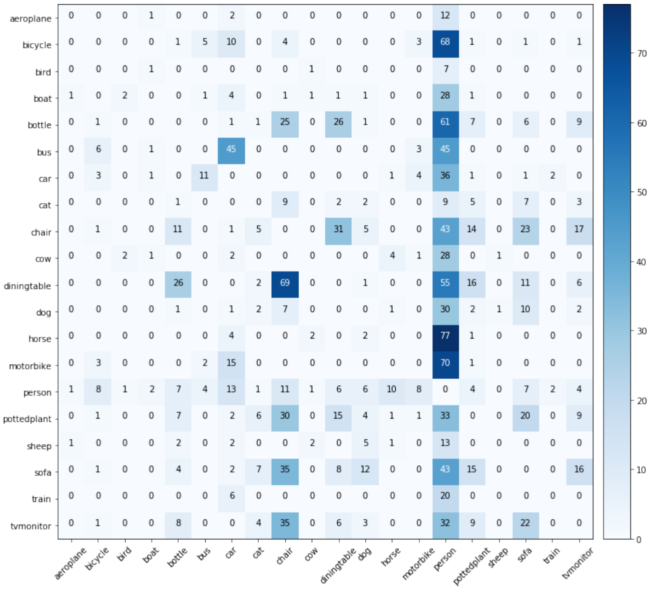
In this section, we compare models optimized with classification loss and ranking AP loss. We report results for model using `max` pooling and `max+min` pooling ($k = 1$). The results are shown in Table 8 and are obtained with an input image size 448×448. To have a fair comparison, we use the same network architecture for all experiments.

We compare our model with the LAPSVM [28], which is, to the best of our knowledge, the only method that optimizes an AP-based loss function over weakly supervised data. We note that both methods optimizing AP ranking during training have better results than classification baseline on all datasets. For our model, we note a significant



motorbike *bottle*

Fig. 4: Visual results of ResNet-WELDON. We show the heatmaps of 2 categories: *motorbike* which is present in the image and *bottle* which is absent. For each class, we show the maximum (resp. minimum) region in green (resp. red) with its corresponding score. The model learns that the *person* class is positively correlated with *motorbike* class. On the contrary, the *bottle* model learns that the motorbike is a negative evidence of the class: the motorbike has a very low score, which shows the absence of *bottle* class.



	aeroplane	bicycle	bird	boat	bottle	bus	car	cat	chair	cow	diningtable	dog	horse	motorbike	person	pottedplant	sheep	sofa	train	tvmonitor
aeroplane	0	0	0	1	0	0	2	0	0	0	0	0	0	0	12	0	0	0	0	0
bicycle	0	0	0	0	1	5	10	0	4	0	0	0	0	3	68	1	0	1	0	1
bird	0	0	0	1	0	0	0	0	0	1	0	0	0	0	7	0	0	0	0	0
boat	1	0	2	0	0	1	4	0	1	1	1	1	0	0	28	1	0	0	0	0
bottle	0	1	0	0	0	0	1	1	25	0	26	1	0	0	61	7	0	6	0	9
bus	0	6	0	1	0	0	45	0	0	0	0	0	0	3	45	0	0	0	0	0
car	0	3	0	1	0	11	0	0	0	0	0	0	1	4	36	1	0	1	2	0
cat	0	0	0	0	1	0	0	0	9	0	2	2	0	0	9	5	0	7	0	3
chair	0	1	0	0	11	0	1	5	0	0	31	5	0	0	43	14	0	23	0	17
cow	0	0	2	1	0	0	2	0	0	0	0	0	4	1	28	0	1	0	0	0
diningtable	0	0	0	0	26	0	0	2	69	0	0	1	0	0	55	16	0	11	0	6
dog	0	0	0	0	1	0	1	2	7	0	0	0	1	0	30	2	1	10	0	2
horse	0	0	0	0	0	0	4	0	0	2	0	2	0	0	77	1	0	0	0	0
motorbike	0	3	0	0	0	2	15	0	0	0	0	0	0	0	70	1	0	0	0	0
person	1	8	1	2	7	4	13	1	11	1	6	6	10	8	0	4	0	7	2	4
pottedplant	0	1	0	0	7	0	2	6	30	0	15	4	1	1	33	0	0	20	0	9
sheep	1	0	0	0	2	0	2	0	0	2	0	5	1	0	13	0	0	0	0	0
sofa	0	1	0	0	4	0	2	7	35	0	8	12	0	0	43	15	0	0	0	16
train	0	0	0	0	0	0	6	0	0	0	0	0	0	0	20	0	0	0	0	0
tvmonitor	0	1	0	0	8	0	0	4	35	0	6	3	0	0	32	9	0	22	0	0

Fig. 5: Normalized co-occurrence matrix between the classes of PASCAL VOC 2007 (trainval). For each category (row), we show the percentage of other objects (column) that appear in the same image. For instance, we observe that in *horse* images, there is usually a *person* (77%), but there is never an *aeroplane* (0%).

improvement of +2.2 pt on VOC 2007, +2.2 pt on VOC 2012 Action and +1 pt on MS COCO. This validates that optimizing AP ranking during training is better than optimizing classification when the performances are evaluated with AP. We can also see that our ranking model outperforms LAPSVM model: +3.3 pt on VOC 2007, +7.4 pt on VOC 2012 Action and +0.8 pt on MS COCO.

Dataset	VOC07	VOCAct	MS COCO
`max` + classif. loss	86.8	71.8	77.4
`max` + AP loss (LAPSVM [28])	87.9	73.3	77.9
`max+min` + classif. loss	89.9	78.5	77.7
`max+min` + AP loss	91.2	80.7	78.7

TABLE 8: Comparison of optimization with classification loss and ranking AP loss.

6.3.4 Pooling analysis

In this section, we analysis our pooling function, and we compare it with standard pooling functions presented in Section 3.5. We report results for an input image of size 448 × 448, but similar behaviors are observed for other scales. To have fair comparison, all the experiments uses the same network (ResNet-101). We analyse the impact of the number of selected instances. We show in Figure 6 the performance with respect to the proportion of selected regions. We can note that the Global Average Pooling (GAP) [18] (resp. MANTRA pooling) is a special case of WELDON pooling, when the proportion of selected regions is 1 (resp. $\sim$0).

Firstly, we can see that negative evidence is important, because on all dataset, MANTRA is better than `max` pooling. In particular, we observe a large improvement of +4.5 pt on VOC 2012 Action dataset, where the context plays an important role [78]. We can also see that region selection is important: on all dataset except MIT67, the WELDON pooling with a proportion of selected instances in [0.2, 0.8] is equal or better than GAP. The WELDON pooling has similar or better results than GAP by using only 20% of regions. Using more regions (50%) gives better results than GAP: +0.4 pt on VOC 2007, +0.3 pt on VOC 2012, +0.6 pt on VOC 2012 Action, +1.9 pt and +2.1 pt on CUB-200. On MIT67, we can see that using a large number of regions is better ($\geq$ 80%). This shows that a large number of regions are discriminants.

We also compare our pooling function to `max` and LSE pooling functions. The LSE pooling is a soft extension of `max` pooling. On all datasets, LSE is better than `max` pooling: +6 pt on CUB-200 and +5.5 on MIT67. This shows that using several regions is more robust than using only the best region. We also note that LSE pooling performances are closed to MANTRA pooling performances. MANTRA pooling, which used only 2 regions, is as efficient as the LSE pooling which used all regions. Except on MS COCO, we observe that the GAP is better than LSE which is better than `max`: using more regions increase the robustness. On MS COCO, we see that the region selection is important because a lot of objects have small sizes: the gain is +3 pt between WELDON with 20% of regions and GAP.

6.3.5 ResNet-WELDON vs. WELDON

We show the improvements of ResNet-WELDON over WELDON [30]. In Tables 4 and 5, we observe that ResNet-WELDON based on ResNet-101 delivers better classification results than WELDON based on VGG16. We also compare the training time (forward+backward) of both architectures for different image sizes. The results in Table 9 are evaluated for one epoch on VOC 2007. We see that the ResNet-WELDON is faster than WELDON for all image sizes. We note that the larger the image size, the greater the improvement. The limitation of the FCN VGG16 is because of the first fully-connected layer that has 4,096 filters of large 7 × 7 spatial size and becomes the computational bottleneck after converting the VGG16 to a FCN [79]. These results show that ResNet-WELDON is better and faster than WELDON.

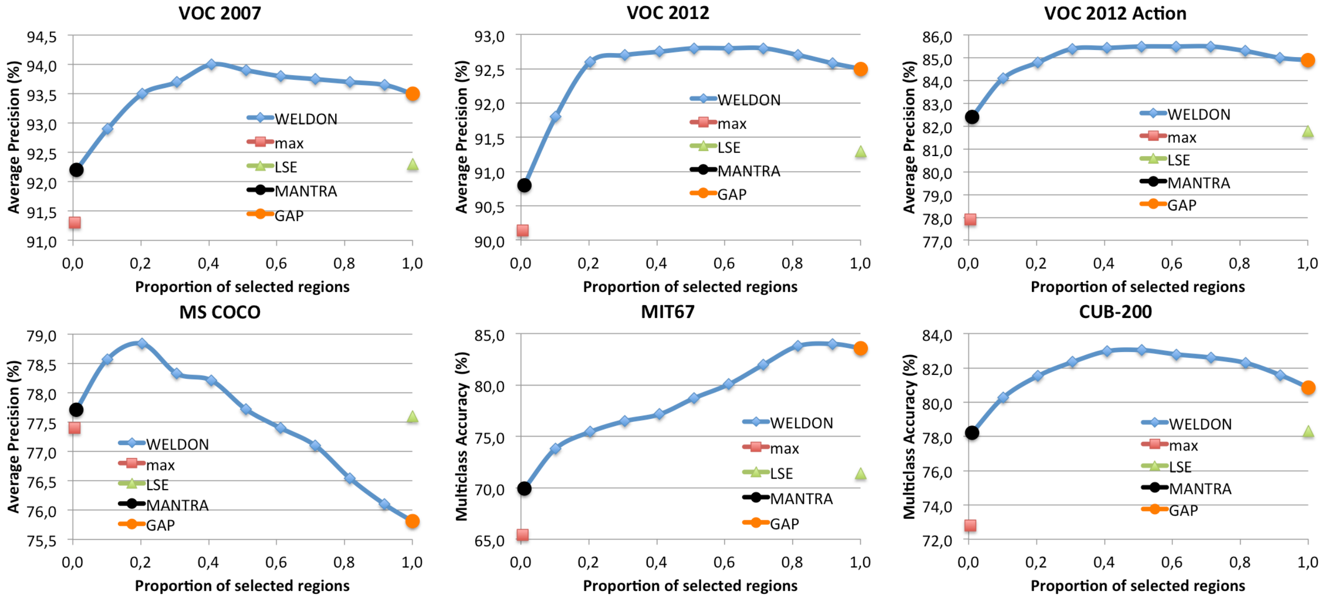


Fig. 6: Pooling analysis. We compare different spatial pooling strategies on 6 datasets for an input image of size 448×448. The x-axis shows the proportion of selected regions and the y-axis the performance. We can see that MANTRA always outperforms the max pooling, which validates the relevance of negative evidence. We can also see that our spatial performes equally or better than GAP with only 20% of regions.

Image size	224	320	448	560	747
WELDON [30]	91	193	391	682	1302
ResNet-WELDON	75	150	286	480	866
Speed-up	+21%	+29%	+36%	+42%	+50%

TABLE 9: Training time (s) for one epoch with different image sizes on VOC 2007.

6.4 Pointwise object localization

In this section, we evaluate the localization performances of our model on VOC 2012 validation set [61] and MS COCO validation set [62]. We use the pointwise localization metric, which is a standard evaluation metric for weakly supervised models introduced by [17]. We report the results of our model and three other methods in Table 10. Our model is trained with an input image 448×448 and $k^+ = k^- = 30$. The DeepMIL [17] is a MIL-based architecture – max pooling. In spite of its simple architecture, our model outperforms the complex cascaded architecture of ProNet [22] and WSLocalization [80], which used a complex strategy based on search- trees to predict locations. Our model outperforms these models on the two datasets, but these models use older architecture. To have a fair comparison, we also report the result of our model for max pooling, and call it ResNet-max. This model is equivalent to DeepMIL and has lower performances than ResNet-WELDON on the two datasets.

Method	VOC 2012	MS COCO
DeepMIL [17]	74.5	41.2
ProNet [22]	77.7	46.4
WSLocalization [80]	79.7	49.2
ResNet-max	80.2	50.6
ResNet-WELDON	**81.5**	**51.7**

TABLE 10: Pointwise object localization performances (MAP) on VOC 2012 and MS COCO.

6.5 Weakly supervised image segmentation

In this section, we show that our model can be applied to weakly supervised image segmentation, while being trained from global image labels only. We evaluate our model on the VOC 2012 image segmentation dataset [61], consisting of 20 foreground object classes and one background class. We train our model with the *train* set (1,464 images) and the extra annotations provided by [81] (resulting in an augmented set of 10,582 images), and test it on the *validation* set (1,449 images). The network is trained with an input image 448×448 and $k^+ = k^- = 50$. The performance is measured in terms of pixel Intersection-over-Union (IoU) averaged across the 21 categories. As in existing methods, we use a fully connected CRF (FC-CRF) [46] to post-process the final segmentation mask. We use the class-specific heatmaps generated by the network as unary potentials of the FC-CRF.

Method	Mean IoU
MIL-FCN [21]	24.9
MIL-Base+ILP+SP-sppxl [20]	36.6
EM-Adapt +FC-CRF [56]	33.8
CCNN + FC-CRF [57]	35.3
SEC [38]	**50.7**
ResNet-WELDON + FC-CRF	42.1

TABLE 11: Comparison of weakly supervised semantic segmentation methods on VOC 2012.

Results. The result of our method is presented in Table 11, and some predictions are shown in Figure 7. We compare it to weakly supervised methods that use only image labels during training. Firstly, we note a large improvement with respect to MIL models based on max pooling [21] (+17.2 pt) or its soft extension [20] (+5.5 pt). This validates the efficient of our negative evidence pooling for segmentation. ResNet-WELDON also significantly outperforms the recent CCNN [57] that uses a loss function to optimize for any set

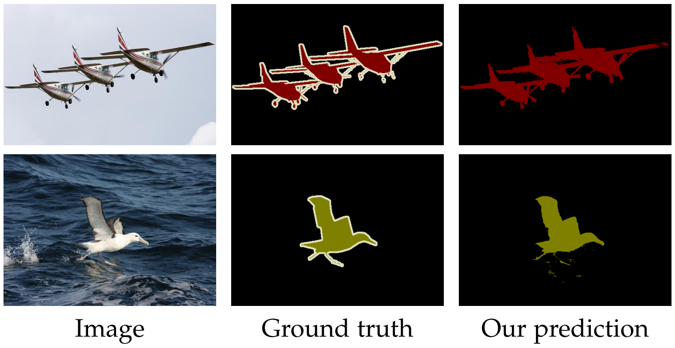


Fig. 7: Segmentation examples on VOC 2012. We show the prediction of our model after FC-CRF and the ground truth.

of linear constraints on the output space of the ConvNet. [38] achieves the best results by using a quite more complex strategy. The training scheme incorporates different terms, which are specifically tailored to segmentation: one enforces the segmentation mask to match low-level image boundaries, another one incorporates prior knowledge to support predicted classes to occupy a certain image proportion. In contrast, our model is generic, and is trained in the same manner for the classification and segmentation. It would be possible to use the specific segmentation priors of [38] in our model to boost performances.

7 Conclusion

We propose a new structured output latent variable model, based on negative evidence. Based on this model, we propose ResNet-WELDON, a new weakly supervised learning dedicated to learn discriminative localized visual features by using only image-level labels during training. ResNet-WELDON model uses a fully convolutional architecture, where the spatial information is naturally preserved throughout the network. Extensive experiments have shown the effectiveness of our model for image classification. We show the scalability and the efficiency of our model on large scale ILSVRC dataset. We also present a detailed experimental comparison of different pooling functions on several datasets. Finally, we show that ResNet-WELDON can be successfully apply for weakly supervised segmentation.

References

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

[2] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in *ICLR*, 2015.

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.

[4] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Region-based convolutional networks for accurate object detection and segmentation," *TPAMI*, 2016.

[5] R. Girshick, "Fast R-CNN," in *ICCV*, 2015.

[6] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object detection via region-based fully convolutional networks," in *NIPS*, 2016.

[7] J. Long, E. Shelhamer, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," in *CVPR*, 2015.

[8] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs," in *ICLR*, 2015.

[9] J. Dai, K. He, Y. Li, S. Ren, and J. Sun, "Instance-sensitive fully convolutional networks," in *ECCV*, 2016.

[10] H. Azizpour, A. S. Razavian, J. Sullivan, A. Maki, and S. Carlsson, "Factors of transferability for a generic convnet representation," *TPAMI*, 2016.

[11] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the Devil in the Details: Delving Deep into Convolutional Nets," in *BMVC*, 2014.

[12] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *International Journal of Computer Vision (IJCV)*, 2015.

[13] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and Transferring Mid-Level Image Representations using Convolutional Neural Networks," in *CVPR*, 2014.

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

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

[16] A. Bearman, O. Russakovsky, V. Ferrari, and L. Fei-Fei, "What's the Point: Semantic Segmentation with Point Supervision," in *ECCV*, 2016.

[17] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Is object localization for free? Weakly-supervised learning with convolutional neural networks," in *CVPR*, 2015.

[18] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning Deep Features for Discriminative Localization," in *CVPR*, 2016.

[19] G. Papandreou, I. Kokkinos, and P.-A. Savalle, "Modeling Local and Global Deformations in Deep Learning: Epitomic Convolution, Multiple Instance Learning, and Sliding Window Detection," in *CVPR*, 2015.

[20] P. O. Pinheiro and R. Collobert, "From image-level to pixel-level labeling with convolutional networks," in *CVPR*, 2015.

[21] D. Pathak, E. Shelhamer, J. Long, and T. Darrell, "Fully Convolutional Multi-Class Multiple Instance Learning," in *ICLR (Workshop)*, 2015.

[22] C. Sun, M. Paluri, R. Collobert, R. Nevatia, and L. Bourdev, "ProNet: Learning to Propose Object-Specific Boxes for Cascaded Neural Networks," in *CVPR*, 2016.

[23] C.-N. Yu and T. Joachims, "Learning structural svms with latent variables," in *ICML*, 2009.

[24] A. Quattoni, S. Wang, L.-P. Morency, M. Collins, and T. Darrell, "Hidden conditional random fields," *TPAMI*, 2007.

[25] W. Ping, Q. Liu, and A. Ihler, "Marginal structured svm with hidden variables," in *ICML*, 2014.

[26] A. G. Schwing, T. Hazan, M. Pollefeys, and R. Urtasun, "Efficient Structured Prediction with Latent Variables for General Graphical Models," in *ICML*, 2012.

[27] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez, "Solving the multiple instance problem with axis-parallel rectangles," *Artif. Intell.*, 1997.

[28] A. Behl, P. Mohapatra, C. V. Jawahar, and M. P. Kumar, "Optimizing average precision using weakly supervised data," *TPAMI*, 2015.

[29] T. Durand, N. Thome, and M. Cord, "MANTRA: Minimum Maximum Latent Structural SVM for Image Classification and Ranking," in *ICCV*, 2015.

[30] ——, "WELDON: Weakly Supervised Learning of Deep Convolutional Neural Networks," in *CVPR*, 2016.

[31] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, "Learning Deep Features for Scene Recognition using Places Database." in *NIPS*, 2014.

[32] N. Zhang, M. Paluri, M. Ranzato, T. Darrell, and L. Bourdev, "PANDA: Pose Aligned Networks for Deep Attribute Modeling," in *ECCV*, 2014.

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

[34] Y. Gong, L. Wang, R. Guo, and S. Lazebnik, "Multi-scale orderless pooling of deep convolutional activation features," in *ECCV*, 2014.

[35] R. Arandjelović, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "NetVLAD: CNN architecture for weakly supervised place recognition," in *CVPR*, 2016.

[36] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object Detection with Discriminatively Trained Part Based Models," in *TPAMI*, 2010.

[37] R. G. Cinbis, J. Verbeek, and C. Schmid, "Weakly Supervised Object Localization with Multi-fold Multiple Instance Learning," in *TPAMI*, 2016.

[38] A. Kolesnikov and C. H. Lampert, "Seed, expand and constrain: Three principles for weakly-supervised image segmentation," in *ECCV*, 2016.

[39] F. X. Yu, D. Liu, S. Kumar, T. Jebara, and S.-F. Chang, "∝svm for learning with label proportions." in *ICML*, 2013.

[40] K.-T. Lai, F. X. Yu, M.-S. Chen, and S.-F. Chang, "Video event detection by inferring temporal instance labels," in *CVPR*, 2014.

[41] W. Li and N. Vasconcelos, "Multiple Instance Learning for Soft Bags via Top Instances," in *CVPR*, 2015.

[42] S. N. Parizi, A. Vedaldi, A. Zisserman, and P. F. Felzenszwalb, "Automatic discovery and optimization of parts for image classification," in *ICLR*, 2015.

[43] S. Andrews, I. Tsochantaridis, and T. Hofmann, "Support vector machines for multiple-instance learning," in *NIPS*, 2003.

[44] H. Azizpour, M. Arefiyan, S. N. Parizi, and S. Carlsson, "Spotlight the negatives: A generalized discriminative latent model," in *BMVC*, 2015.

[45] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs," in *ICLR*, 2015.

[46] P. Krähenbühl and V. Koltun, "Efficient inference in fully connected crfs with gaussian edge potentials," in *NIPS*, 2011.

[47] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. Torr, "Conditional random fields as recurrent neural networks," in *ICCV*, 2015.

[48] L. Chen, A. G. Schwing, A. L. Yuille, and R. Urtasun, "Learning deep structured models," in *ICML*, 2015.

[49] S. Wang, S. Fidler, and R. Urtasun, "Proximal deep structured models," in *NIPS*, 2016.

[50] D. Belanger and A. McCallum, "Structured prediction energy networks," in *ICML*, 2016.

[51] Y. Song, A. G. Schwing, R. S. Zemel, and R. Urtasun, "Training deep neural networks via direct loss minimization," in *ICML*, 2016.

[52] K. Miller, M. P. Kumar, B. Packer, D. Goodman, and D. Koller, "Max-margin min-entropy models," in *AISTATS*, 2012.

[53] D. Bouchacourt, S. Nowozin, and M. Pawan Kumar, "Entropy-based latent structured output prediction," in *ICCV*, 2015.

[54] D. Bouchacourt, P. K. Mudigonda, and S. Nowozin, "Disco nets : Dissimilarity coefficients networks," in *NIPS*, 2016.

[55] Y. Yue, T. Finley, F. Radlinski, and T. Joachims, "A support vector method for optimizing average precision," in *SIGIR*, 2007.

[56] G. Papandreou, L.-C. Chen, K. Murphy, and A. L. Yuille, "Weakly- and semi-supervised learning of a DCNN for semantic image segmentation," in *ICCV*, 2015.

[57] D. Pathak, P. Krahenbuhl, and T. Darrell, "Constrained Convolutional Neural Networks for Weakly Supervised Segmentation," in *ICCV*, 2015.

[58] A. Quattoni and A. Torralba, "Recognizing indoor scenes," in *CVPR*, 2009.

[59] P. Mohapatra, C. Jawahar, and M. P. Kumar, "Efficient optimization for average precision svm," in *NIPS*, 2014.

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

[61] ——, "The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results."

[62] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common Objects in Context," in *ECCV*, 2014. [Online]. Available: http://mscoco.org

[63] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The Caltech-UCSD Birds-200-2011 Dataset," Tech. Rep., 2011.

[64] L.-J. Li, H. Su, E. P. Xing, and L. Fei-Fei, "Object Bank: A High-Level Image Representation for Scene Classification & Semantic Feature Sparsification," in *NIPS*, 2010.

[65] Y. Wei, W. Xia, J. Huang, B. Ni, J. Dong, Y. Zhao, and S. Yan, "CNN: single-label to multi-label," *CoRR*, vol. abs/1406.5726, 2014. [Online]. Available: http://arxiv.org/abs/1406.5726

[66] G. Sharma and B. Schiele, "Scalable nonlinear embeddings for semantic category-based image retrieval," in *ICCV*, 2015.

[67] Z. Wei and M. Hoai, "Region Ranking SVM for Image Classification," in *CVPR*, June 2016.

[68] P. Kulkarni, F. Jurie, J. Zepeda, P. Pérez, and L. Chevallier, "Spleap: Soft pooling of learned parts for image classification," in *ECCV*, 2016.

[69] Y. Gao, O. Beijbom, N. Zhang, and T. Darrell, "Compact Bilinear Pooling," in *CVPR*, 2016.

[70] T. Xiao, Y. Xu, K. Yang, J. Zhang, Y. Peng, and Z. Zhang, "The application of two-level attention models in deep convolutional neural network for fine-grained image classification," in *CVPR*, June 2015.

[71] M. Jaderberg, K. Simonyan, A. Zisserman *et al.*, "Spatial Transformer Networks," in *NIPS*, 2015.

[72] R. Wu, B. Wang, W. Wang, and Y. Yu, "Harvesting discriminative meta objects with deep cnn features for scene classification," in *ICCV*, 2015.

[73] M. Simon and E. Rodner, "Neural activation constellations: Unsupervised part model discovery with convolutional networks," in *ICCV*, December 2015.

[74] S. Huang, Z. Xu, D. Tao, and Y. Zhang, "Part-stacked cnn for fine-grained visual categorization," in *CVPR*, June 2016.

[75] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *CVPR*, 2015.

[76] S. Xie, R. Girshick, P. Dollr, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," Nov 2016. [Online]. Available: http://arxiv.org/pdf/1611.05431v1

[77] C. Szegedy, S. Ioffe, and V. Vanhoucke, "Inception-v4, inception-resnet and the impact of residual connections on learning," 2016.

[78] G. Gkioxari, R. Girshick, and J. Malik, "Contextual action recognition with r* cnn," in *CVPR*, 2015.

[79] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected crfs," in *ICLR*, 2015.

[80] A. J. Bency, H. Kwon, H. Lee, S. Karthikeyan, and B. S. Manjunath, "Weakly supervised localization using deep feature maps," in *ECCV*, 2016.

[81] B. Hariharan, P. Arbelaez, L. D. Bourdev, S. Maji, and J. Malik, "Semantic contours from inverse detectors." in *ICCV*, 2011.

Thibaut Durand received the Ph.D. in Computer Vision and Machine Learning from the University of Pierre et Marie Curie, France, in 2017. He received an M.Sc. in Electrical Engineering by ENSEA, France, and an M.Sc. degrees in computer science from the University of Cergy-Pontoise, France, in 2013. His research interests include computer vision and machine learning for image classification.

Nicolas Thome is full professor at Conservatoire Nationnal des Arts et Métiers (Cnam Paris). He received the Ph.D. degree in computer science from the University of Lyon, France in 2007, and has been associate professor at UPMC-Paris 6 from 2008 to 2016. His research interests include machine learning for computer vision, including applications for semantic understanding of multimedia data. He is involved in several French (ANR), European and international (Canada, Singapore, Brazil) research projects. He is being coordinator of an ANR project on weakly supervised learning for image retrieval in 2013-2017.

Matthieu Cord received the Ph.D. degree in computer science from the UCP, France, before working in the ESAT lab at KUL University, Belgium, and in the ETIS lab, France as Assistant Professor. He joined the Computer Science department LIP6, at UPMC Sorbonne Universities, Paris, in 2006 as full Professor. In 2009, he was nominated at the IUF (French Research Institute) for a 5 years delegation position. His research interests include Computer Vision, Image Processing, and Pattern Recognition. He developed several systems for content-based image and video retrieval, focusing on interactive learning-based approaches. He is also interested in Machine Learning for Multimedia processing, Digital preservation, and Web archiving. Prof. Cord has published a hundred scientific publications and participated in several international projects (European FP6 and FP7, Singapore, Brazil) on these topics. He is a member of the IEEE.