Spatial Consistency Loss for Training
Multi-Label Classifiers from Single-Label Annotations

Thomas Verelst¹* Paul K. Rubenstein² Marcin Eichner² Tinne Tuytelaars¹ Maxim Berman²
¹ESAT-PSI, KU Leuven, Belgium ²Apple
¹{firstname.lastname}@esat.kuleuven.be

Abstract

Multi-label image classification is more applicable “in the wild” than single-label classification, as natural images usually contain multiple objects. However, exhaustively annotating images with every object of interest is costly and time-consuming. We train multi-label classifiers from datasets where each image is annotated with a single positive label only. As the presence of all other classes is unknown, we propose an Expected Negative loss that builds a set of expected negative labels in addition to the annotated positives. This set is determined based on prediction consistency, by averaging predictions over consecutive training epochs to build robust targets. Moreover, the ‘crop’ data augmentation leads to additional label noise by cropping out the single annotated object. Our novel spatial consistency loss improves supervision and ensures consistency of the spatial feature maps by maintaining per-class running-average heatmaps for each training image. We use MS-COCO, Pascal VOC, NUS-WIDE and CUB-Birds datasets to demonstrate the gains of the Expected Negative loss in combination with consistency and spatial consistency losses. We also demonstrate improved multi-label classification mAP on ImageNet-1K using the ReaL multi-label validation set.

1. Introduction

In the last decade, computer vision has seen great progress thanks to the emergence of large-scale data-driven machine learning. With enough annotated data, machine perception has reached or exceeded human accuracy in many difficult tasks, in particular single-label image classification [43]. Yet obtaining large amounts of annotated data remains a challenge, especially in more granular object recognition tasks such as multi-label classification, object detection or instance segmentation. Exhaustively annotating all objects in images on a large scale is time-consuming, and error-prone. To reduce the annotation cost, some large-scale datasets such as OpenImages [28] only annotate a subset of the object classes for each image in the dataset. In this case, the annotation process yields a set of positive labels guaranteed to be in the image, a set of negative labels guaranteed to be absent from the image, and a set of unknown labels for which no information is provided.

A more extreme setting, which reduces the annotation effort substantially, is the annotation of a single positive label per image, with no negative labels. This type of annotation is sensible for a single-label classification task, where the single annotation is intended to represent the main object of interest. Yet, it is clear that most natural images contain more than one object. For example, it has been shown that the ImageNet dataset for image classification [10] contains images with multiple objects of the annotated cate-

*Work done during an internship at Apple.
gories [58], with an average of 1.22 positive labels per image. The usage of a "one-versus-all" cross-entropy loss in combination with this specific type of label noise can hurt the performance of the classifier. Regularization, either implicitly through, e.g., stochastic optimization or explicitly through the use of label smoothing techniques [47, 52], can improve the accuracy and might help the classifier to learn a useful mapping in spite of the inherent label noise.

Other work acknowledges that the images of single-label datasets such as ImageNet can contain more than one object in practice [41, 2, 58]. In such setting, a single-labeled dataset can be thought of as a weakly-labeled multi-label classification dataset, with a single positive annotation per image. A common strategy is to consider all unannotated labels as negatives [9] in combination with a binary cross-entropy loss, introducing label noise and incorrect supervision by treating the unannotated positive labels as negatives.

Our method builds a set of expected positive and expected negative labels, using robust label scores that are estimated by tracking exponential moving averages (EMAs) of the network outputs over training epochs. This way to get robust estimates is similar to ensembling methods [29]. The expected positives are then selected as the highest-scoring labels. Whilst akin to pseudo-labeling [30], we show that ignoring the expected positives in the binary-cross entropy loss is essential to achieve good results.

The score estimates naturally lead to the application of a consistency loss (CL), popular in weakly-supervised learning with unannotated data [29, 44, 45, 24], which further increases the supervision for unannotated labels. However, we observe that the single positive annotated label might also be a source of label noise when training classifiers in conjunction with image crops as a data-augmentation technique. Cropping an image risks removing the object corresponding to the ground truth annotation, misleading the optimization as illustrated in figure 1.

Thus, we extend the consistency loss in the spatial domain, introducing a spatial consistency loss (SCL). By taking EMAs of the spatial outputs of the network over consecutive training epochs, we obtain spatial heatmaps which localize objects in the image, beyond the single ground truth label. The SCL uses these spatial running averages as additional source of self-supervision which further improves the accuracy of the network.

The contributions of this work are as follows:

• Our expected negative (EN) scheme trains multi-label classifiers from single positive label annotations, by building a set of expected unannotated positives and expected negatives. Expected positives are Expected unannotated positives are ignored in the binary cross-entropy loss, which is essential for good performance;
• We introduce a spatial consistency loss (SCL) that extends CL in the spatial domain, improving the multi-label accuracy and acting in synergy with the ubiquitous “re-size+crop” data augmentation;
• We measure the gains stemming from our contributions on MS-COCO, Pascal VOC, NUS-WIDE and CUB-Birds in the single positive setting, as well as on ImageNet-1K evaluated using multi-label annotations.

2. Related work

Partial annotations. Collecting exhaustive multi-label classification annotations on a large number of classes and images can be intractable, which is why many large-scale datasets resort to partial annotations [34]. For instance, for each image in OpenImages [28] and LVIS [17], only a small fraction of the labels are annotated. Collecting a larger amount of partially labeled data can sometimes lead to better performance than a smaller set of fully-annotated data [13]. Partial labels can also occur naturally when training a model on the combination of several datasets with disjoint label spaces [56, 60].

Multi-label learning with missing labels can be framed as a transductive learning problem, where one aims to explicitly recover complete annotations that are consistent with the partial annotations provided [54]. Graph neural networks [53, 7, 13, 50, 35, 22, 31] or adversarial training [57] can be used to predict the missing labels from the annotated ones. Label co-occurrence analysis could be used to estimate the confidence of labels [3, 23]. A simple way to handle missing labels is to consider them as negatives [46, 4]. However, this deteriorates performance due to label noise. [25] shows that high-capacity models might memorize the noisy labels. Ignoring unannotated classes in the loss function can alleviate this issue [13], but this is inapplicable when the annotations only contain positives [9].

Training with a single positive label can be considered as a combination of single-label learning [39, 12, 21] and positive-unlabeled learning [11, 1]. Cole et al. [9] compare several baselines and propose a regularized online label estimation (ROLE) method that estimates the missing labels during training, by jointly optimizing a label estimator and image classifier. The output of one serves as ground-truth for the other, with the intuition that both are more likely to converge to the same solution. Other approaches reweight samples based on their loss values [59, 42]. Large Loss Matters [25] marks elements with large loss values as mislabeled and ignores or reweights those.

Semi-supervised learning. Semi-supervised learning uses a set of unlabeled data samples in addition to the fully-labeled samples, and is a special case of partial annotation [16]. One way to incorporate unlabeled samples in the training process is by encouraging consistency of predictions on these samples over different epochs or augmentations [44, 24]. Ladder networks [40] encourage
consistency between a standard branch and the denoised predictions of a corrupted branch. [29] proposes the Π-model, enforcing consistency between two perturbed versions of the same sample. In addition, they propose self-ensembling to build a consensus prediction by averaging outputs among different training epochs. Our consistency losses in sections 3.4 and 3.5 apply similar ideas directly on the training set, rather than on a held-out dataset of unlabeled images.

Other methods use pseudo-labeling to leverage unannotated images. [30] uses the highest-scoring class as the true label for unlabeled data. FixMatch combines pseudo-labels and consistency regularization [45]. However, pseudo-labels are prone to concept drift and confirmation bias, where early mislabeled samples lead to accumulating errors. Curriculum labeling [5] mitigates this using a refined training strategy. Noisy student [55] demonstrated state-of-the-art results on ImageNet [27] using self-training and distillation on a large set of unlabeled images, by iterative re-labling data and using increasingly larger student models. By contrast, we choose to ignore the labels that we identify as possible positives (section 3.3) rather than incorporating them in the positive annotations, avoiding concept drift.

**Data augmentation and instance discrimination.** Our CL and SCL losses enforce consistency of the network across subsequent training epochs, which favors invariance of the network outputs to the data augmentation. This can be connected to recent trends of self-supervised learning for instance discrimination, ensuring that the embeddings of data-augmented versions of an instance are closer in embedding space than the embeddings of different instances [48, 36, 18, 20, 6]. In the fully annotated multi-label image classification setting, [15] encourages consistency of the spatial activations of the network among two data augmentations of an image, akin to a spatial extension of the Π-model [29]. In the semi-supervised single-label setting, our SCL of section 3.5 uses a similar idea of encouraging consistency of the spatial class outputs, but uses a temporal ensemble over the different training epochs to do so, rather than directly comparing the outputs of data-augmented copies during a single training iteration.

### 3. Method

#### 3.1. Problem statement

We state the problem of multi-label classification with partially annotated labels similarly to [9]. Our goal is to learn a mapping from an image $x_n$ to the indicator vector $y_n \in \{0, 1\}^L$ of the classes contained in the image, $L$ being the number of classes. We use a dataset $(x_n, z_n)_{n=1}^N$, where each input image $x_n$ has a partial annotation $z_n \in \{0, 1\}^L$. The positive labels encoded by 1 are contained in the image; the negative labels 0 are absent from the image; missing labels encoded by $\emptyset$ can be either present or absent. In the single positive setting, there is a single positive label $i$ for each image such that $z_{ni} = 1$; all other labels $j \neq i$ are supposed unknown ($z_{nj} = 0$).

Given an image $x_n$, a neural network classifier predicts $L$ label probabilities $f_n \in [0, 1]^L$. At training time, the network parameters are optimized to minimize the empirical risk on the training set, measured with a loss function $L$. A common multi-label classification loss is the binary cross entropy (BCE) loss

$$L_{\text{BCE}}(f_n) = -\frac{1}{L} \sum_{i=1}^{L} [z_{ni} = 1] \log(f_{ni}) + [z_{ni} = 0] \log(1 - f_{ni}) \quad (1)$$

with $[\cdot] \in \{0, 1\}$ the Iverson bracket equal to 1 iff. the condition holds. With incomplete annotations, missing labels (where $z_{ni} = \emptyset$) are ignored in eq. (1) and thus not penalized. Although natural, this modeling is not suited for training with only positive annotated labels, such as the single positive setting that we consider. In such a setting, nothing prevents the network from predicting all $L$ classes regardless of the input, as there is no penalty for false positives.

#### 3.2. Assume-negative loss (AN)

One simple strategy to handle single-positive labels is to assume that all unknown labels are negatives. This leads to the assume-negative (AN) loss function [9]

$$L_{\text{AN}}(f_n) = -\frac{1}{L} \sum_{i=1}^{L} [z_{ni} = 1] \log(f_{ni}) + [z_{ni} \in \{0, \emptyset\}] \log(1 - f_{ni}) \quad (2)$$

In this case, unobserved labels (where $z_{ni} = \emptyset$) are considered as negatives. This is justifiable since the number of objects present in an image is typically small, leading to only a few false negatives in the supervision, weighed against many true negatives supervised correctly. However, the false negatives of the AN loss can have a large impact on the accuracy. Our interpretation is that the network is penalized strongly by the binary cross-entropy loss when predicting high scores for missing positive labels. Therefore, the missing positive labels in AN lead to a large incorrect supervision that can dominate the contribution to the loss from the true negatives.

#### 3.3. Expected-negative loss (EN)

We design a strategy to ignore the large incorrect contributions of noisy labels in the Assume Negative loss, by tracking a set of samples that we expect to be negatives for each class. To this effect, we build robust score estimates for
each unannotated label, and consider high-scoring labels as expected positives and other labels as expected negatives. We use a hyperparameter $K$ which sets the number of expected positive labels per image. For a training set of size $N$, the expected number of ground truth positives with class $i$ is given by

$$p_i = KN \cdot \frac{\sum_{n=1}^{N} [z_{ni} = 1]}{N} = K \sum_{n=1}^{N} [z_{ni} = 1],$$

assuming that the class distribution of annotated labels $\sum_{n=1}^{N} [z_{ni} = 1]/N$ is similar to the unknown true distribution $\sum_{n=1}^{N} Y_{ni}/N$.

The score estimates, which are used to determine the $p_i$ most likely unannotated positive labels, are obtained by keeping running-average estimates per label, similarly to consistency losses [29, 44, 45, 24]. Over consecutive training epochs, the network sees different data-augmented versions of an image; keeping running averages of the model outputs on these different augmentations leads to more robust label estimates. At training epoch $t$, the estimated scores $s^t_n$ are updated with the network outputs $f^t_n$ as an EMA

$$s^t_n = \mu s^{t-1} + (1 - \mu)f^t_n$$

with $\mu$ the momentum. The scores $s^0_n$ are initialized to 1 for the positive label, i.e. $s^0_{ni} = 1$ if $z_{ni} = 1$, and 0 otherwise.

At the beginning of each epoch $t$, we identify the top-$p_i$ instances for each class $i$ among the running-average score estimates $(s^t_{ni})_{n=1...N}$ as likely to correspond to positive ground-truth labels. We set $z^t_{ni} \in \{0, 1\}$, where 1 is an indicator for expected positive labels and 0 for expected negative labels. In the first training epoch, we initialize $z^0_{ni} = 1$ if $z_{ni} = 1$ and 0 otherwise.

We show in Sec. 4.2 that simply considering expected positives as positives leads to unsatisfactory results, possibly due to label drift of those pseudo-labels, where early mislabeled samples lead to accumulating errors [5]. Our expected negative (EN) only applies a binary-cross entropy loss on annotated positives and the set of expected negatives, ignoring the expected positive labels in the loss. This leads to the following loss function:

$$\mathcal{L}_{EN}(f_n) = -\frac{1}{L} \sum_{i=1}^{L} [z_{ni}=1] \log(f_{ni}) + [z_{ni}=0] \log(1-f_{ni}).$$

Contrary to the AN loss, $\mathcal{L}_{EN}$ does not assume all unannotated labels to be negatives, but only the ones that are not part of the expected positive samples.

### 3.4. Consistency loss (CL)

As the Expected Negative loss builds robust targets for unannotated samples, we experiment with using these targets as additional supervision. This leads to a consistency loss, which is commonly used in semi-supervised methods with unannotated samples [29, 44, 45, 24].

The consistency loss (CL) is given by the $\ell_1$-distance between the predicted $s^t_n$ and the running averages $f^t_n$:

$$\mathcal{L}_{CL}(f^t_n) = \|f^t_n - s^{t-1}_n\|_1.$$  

### 3.5. Spatial consistency loss (SCL)

Even though the running averages $s^t_n$ provide robust label scores, they lead to an additional source of label noise when training multi-label classifiers, as objects might be cropped out the frame when using the prevalent ‘crop’ augmentation during training. For this reason, we extend the running averages in the spatial dimension, using score heatmaps to track the average scores per spatial position of the image. This spatial consistency loss (SCL) ensures consistency over multiple predictions, even when the image is being cropped randomly.

We consider a typical classifier network architecture with a convolutional backbone, an average pooling operation over the features and a fully connected classification layer. To obtain spatially localized class-specific predictions, we modify the network architecture by (i) interpreting the fully connected layer as a $1 \times 1$ convolution, and (ii) applying it before the pooling operation rather than after.

Assuming square input images for the sake of exposition, this modification produces spatial score maps $F_n \in [0, 1]^{G \times G}$, with $G \times G$ the spatial dimensions of the feature map. Applying the fully-connected layer to every spatial location of the feature map increases the computations at training time. However, due to the distributive property, the order of the average pooling and the $1 \times 1$ convolution layers can be reversed without affecting the network outputs, as explained in appendix J. Consequently, our modification causes no computational penalty during inference.

For each image $n$, we keep score heatmaps $H^t_n \in [0, 1]^{W \times W}$ which contain running averages of the output score maps $F^t_n$ at epoch $t$. The heatmap size $W$ is a multiple of $G$, allowing to store details in the heatmaps at a finer resolution than the score maps; in practice, we use $W = 2G$. When feeding the input $x_n$ to the network, we record the spatial transformation $T^t_n$ used in the data augmentation, such as cropping and flipping. Given this transformation, only the visible part of the heatmaps $H^t_n$ is updated with an EMA: the score maps $F^t_n$ are resized with bilinear interpolation to fit the cropped region, and flipped if needed. Heatmap regions that are cropped out of the input are not updated. Similar to the CL method, the heatmaps are initialized to 1 for the annotated ground truth and 0 for the other classes.

The spatial consistency loss (SCL) is the $\ell_1$-distance between the score heatmap and the network output. The input augmentation transformation $T^t_n$ is first applied on the
running-average heatmap. The result is then rescaled to match the dimensions of \( F_n^t \). The SCL is given by

\[
L_{SCL}(F_n^t) = \|F_n^t - \text{resize}(T_n^t(H_n^{t-1})))\|_1.
\]

In our experiments, we use the EN loss in combination with the CL or SCL, with a weighting parameter \( \gamma \):

\[
\mathcal{L} = \mathcal{L}_{EN} + \gamma \mathcal{L}_{SCL}.
\]

4. Experiments

4.1. Results and comparison

Dataset, setup and metrics. We use MS-COCO 2014 [33], Pascal VOC 2012 [14], NUS-WIDE [8] and Caltech-UCSD Birds-200-2011 (CUB) [49] as benchmarks for multi-label classification. In order to test our contributions, we use the code shared by [9] to simulate a single-positive annotated setting, and reproduce their train, validation and test samples. The validation and test splits are fully annotated, and the training samples have a single label by randomly picking a single ground-truth positive label per image. Details are in appendix K.

We report the mean average precision (mAP) on the test split, using the epoch corresponding to the best validation mAP. The ResNet-50 [19] model from torchvision [38] is trained at a resolution of \( 448 \times 448 \), as in [9]. We use random crop augmentations (area scale 0.25 to 1) and random horizontal flip; details and ablation on the scale are provided in appendix A. We use the Adam optimizer [26] and batch size of 8. With ImageNet-1k pretraining [43], the final linear layer is trained for 5 epochs with learning rate \( 10^{-3} \), followed by 25 epochs of finetuning of the whole network with a learning rate of \( 10^{-5} \) and cosine annealing. When trained from scratch, the model is trained for 100 epochs with learning rate \( 10^{-4} \) and cosine annealing.

We compare with related work ROLE [9] and Large Loss Matters (LL) [25]. Additionally, we retrain the following baselines with our training setup: Assume Negative (AN), AN with label smoothing (LS) where the optimal label smoothing parameter selected among \{0.1, 0.2\}, and Weak Assume Negative (WAN) [9] which down-weights negatives in the loss. We use the codebase shared by [9] to report the performance of ROLE with our setup. Comparison with [59] is in appendix B as it uses a different data split, which also includes partial labeling experiments where 40% or 75% of the positives are labeled instead of only a single positive.

SCL/CL implementation details. Given \( 448 \times 448 \) inputs, the network outputs \( 14 \times 14 \) score maps. Score heatmaps are stored with size \( 28 \times 28 \) in 8-bit unsigned integer format. After linear pretraining, we use CL and SCL in combination with EN according to eq. (8). The EMA momentum is set to \( \mu=0.8 \). Loss weight \( \gamma \) is searched in \{0.1, 1\}, and we test the best model based on validation results. No other experiment-specific hyperparameter searching is done, in contrast to related work [9, 25]. We set the expected number of positives \( K \) based on validation set annotations (see appendix K): 2.9 for MS-COCO, 1.5 for VOC, 1.9 for NUS-WIDE and 31.5 for CUB.

Results. Table 1 compares our method to other baselines and related work [9, 25]. The results show that the Expected Negative (EN) loss outperforms assume-negative (AN), by avoiding penalization of unannotated positive labels. As EN uses the EMA scores to determine ignored labels, it is simple to combine with a consistency loss (CL). The SCL further improves the results thanks to localized self-supervision, significantly outperforming related work Large Loss Matters [25] on all datasets except VOC, and ROLE [9] on all datasets except NUS-WIDE (although scoring lower when reproduced with our setup).

4.2. Analysis and ablation

Ablation experiments are performed on MS-COCO with ImageNet pretraining, with the same setup as in section 4.1; we report the best results on the validation split.

Spatial heatmaps. Some qualitative examples of spatial heatmaps are with in fig. 2. We show heatmaps for the positive annotated class, as well as selected heatmaps for unannotated classes. The heatmaps exhibit localization of many objects in the image absent from the single-label ground truth. Figure 3 shows the progress during training. Figure 4 compares heatmaps with and without \( L_{SCL} \) (setting \( \gamma=0 \)), and shows that SCL localizes objects more precisely, avoiding false predictions for negative classes. Appendix H presents another example and appendix I contains uncurated heatmaps, showing the observations hold in general.

Bias towards single-positive predictions. Figure 5a shows the distributions of the top-1 scores, per method, over all validation images. An extended version with top-4 scores is in supplementary material (appendix F). In contrast to the fully annotated baseline, the single-positive dataset in combination with AN loss leads to low-scoring predictions. The EN + SCL loss (eq. (8)) reduces the number of false negative labels and leads to a distribution more akin to the fully annotated case.

In table 2, we compare strategies to avoid bias towards single-positive predictions. The EN loss in eq. (5) ignores expected positive samples. In contrast, the expected positive loss \( L_{EFP} \) uses those as additional positives in the super-
Table 1. Mean average precision (mAP) obtained on the test set of Pascal VOC 2012 [14] and MS-COCO 2014 [33], NUS-WIDE [8] and CUB [49]. ImageNet-1K [43] pretraining warms up the linear layer for 5 epochs. Results indicated with † are reported by related work.

\[
\mathcal{L}_{EP}(f_n) = -\frac{1}{L} \sum_{i=1}^{L} \left[ z_{ni} = 1 \lor z_{ni}^t = 1 \right] \log(f_{ni}) \\
+ \left[ z_{ni}^t = 0 \right] \log(1 - f_{ni}).
\]

We find \( \mathcal{L}_{EP} \) to perform poorly; we believe incorrect expected-positives disturb the training progress by introducing concept drift. We also compare the EN loss with the expected positive regression loss \( \mathcal{L}_{EPR} \) of [9], which regresses the sum of the predicted probabilities towards the estimated number of positives \( K \). Generally, \( \mathcal{L}_{EN} \) in combination with \( \mathcal{L}_{CL} \) or \( \mathcal{L}_{SCL} \) performs best among competing methods.

**EMA momentum parameter.** Figure 5b compares the validation mAP for values of \( \mu \). With \( \mu = 1.0 \), heatmaps are not updated by the predictions. On the validation set, the value we use in our experiments \( \mu = 0.8 \) corresponds to an optimum between updating the heatmaps and building accurate object localizations.
Hyperparameter K. Figure 5c explores different values for the hyperparameter $K$. The optimal value is $K=2.5$. In our experiments, we simply use 2.9 as determined on the validation set statistics. Figure 5d compares values of $K$ when restricting the evaluation to images containing $1, 2, \ldots, 7$ true positive labels. We see that $K$ tunes the tendency of the classifier to predict more or less positives.

The supplementary material further includes a study of the $\ell_1, \ell_2, \ell_{JSD}$ distance functions and weights $\gamma$ in appendix E, the crop augmentation in appendix C and the improvement of SCL for small object sizes in appendix D.

4.3. Multi-label classification on ImageNet-1K

We apply our method to train a multi-label classifier on ImageNet-1K [10], for which multi-label ground truth is not available. This single-label dataset has 1.2 million training and 50K validation images. As in section 4.1, we use a ResNet-50 network pretrained on ImageNet. We compare the accuracies obtained when finetuning with AN loss (eq. (2)), and EN loss combined with CL or SCL (eq. (8)). We use an Adam optimizer [26] with weight decay $10^{-4}$. The linear classification layer is trained for 5 epochs with learning rate $10^{-4}$ before finetuning the whole network for 25 epochs with cosine learning rate decay. We use the standard crop and flip augmentations from [19]. We use $224\times224$ inputs, leading to score maps of size $7\times7$ and heatmaps of size $14\times14$ in the SCL. To limit the memory usage, we only keep heatmaps for the 10 top-scoring classes after the warmup stage in the SCL (details in appendix G).

We report the top-1 validation accuracy on the ImageNet validation set. We also use the relabeled multi-label annotations of ReaL [2], containing annotations for 46837 validation images, with $K=1.22$ positive labels per image on average. On the Real set, we report the top-1 accuracy [2]

$$\text{top-1}_{\text{Real}} = \frac{1}{N} \sum_{n=1}^{N} \left[ \arg\max_{y} \left( f_n \right) \in \{ i \mid y_{ni} = 1 \} \right],$$

as well as the mean average precision (mAP), and subsets of images having $k = \{1, 2, 3, 4\}$ labels. We report all metrics at the end of the finetuning.

The results are detailed in table 3. Finetuning with AN already improves the single-label top-1 accuracy of the network, as observed by previous work [52] and gives a significant boost in multi-label mAP metric. We observe further improvement in the multi-label metrics when adding CL and SCL losses. We note that these methods bring the most improvements over AN when looking at the mAP over images with $k = 1$ or $k = 2$ labels, which constitute 96% of the validation set. This is to be expected given the value of the hyperparameter $K = 1.2$ for this dataset, which favors images with 1 or 2 labels over images with more labels.

4.4. Limitations of the method

Spatial heatmaps stored in 8-bit unsigned integer format use $NLW^2$ bytes of memory, which is around 8 GB for MS-COCO ($N=112K$, $L=81$, $W=28$). For larger datasets, memory constraints can be alleviated by keeping top-$k$ heatmaps after pretraining as we do in section 4.3, or by offloading the heatmaps to disk with asynchronous I/O.

Like [9] our experiments use an oracle value of the number of expected positives per image $K$ set using statistics from annotated samples. This value is dependent on the data collection procedure of the dataset: for instance, ImageNet mostly contains images with one object, whereas MS-COCO images contain many objects. Therefore, some calibration of this value is to be expected depending on the dataset and of the properties desired from the classifier.
Figure 4. Comparison of heatmaps generated in the final training epoch with and without spatial consistency loss.

![Heatmaps](image)

Figure 5. Ablations on MS-COCO validation set with ImageNet-pretrained ResNet-50.

<table>
<thead>
<tr>
<th></th>
<th>top-1 IN-val</th>
<th>top-1 ReaL</th>
<th>mAP ReaL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>k = all</td>
<td>k = 1</td>
</tr>
<tr>
<td>ResNet-50</td>
<td>76.1</td>
<td>46,837</td>
<td>46,837</td>
</tr>
<tr>
<td>ResNet-50 + AN</td>
<td>76.9</td>
<td>83.0</td>
<td>81.4</td>
</tr>
<tr>
<td>ResNet-50 + EN with CL</td>
<td>77.1</td>
<td>83.4</td>
<td>81.7</td>
</tr>
<tr>
<td>ResNet-50 + EN with SCL</td>
<td>77.1</td>
<td>83.9</td>
<td>82.3</td>
</tr>
</tbody>
</table>

Table 3. We finetune ResNet-50 with AN, consistency loss (CL) or spatial consistency loss (SCL). We report top-1 validation accuracy on ImageNet-val (single-label) and on ReaL (multi-label); as well as mean average precision (mAP) on ReaL. mAP is reported on all images (k = all), or on subsets of images with k = 1, 2, 3, 4+ annotated labels.

5. Conclusion

We studied the problem of training a multi-label classifier using only a single-positive label per image, improving the accuracy using spatial consistency losses. In addition, we showed that standard training strategies result in a bias towards negative predictions and proposed a method to build a set of expected-positive labels, which are not penalized in the training loss.

While we have focused our efforts on the ubiquitous single-positive labeled setting, our work can be naturally extended to other partial annotation settings. Besides image crops, other data-augmentations such as affine transformations or masking could be similarly leveraged to enforce consistency of the neural network’s feature maps across training epochs. Finally, we note that an extension of our approach may also be beneficial in other data modalities making use of data augmentations similar to random cropping or masking, such as word deletion in text classification [51], or frequency masking with audio data [37].

Acknowledgement

This work was partially done during an internship at Apple and partially funded by KU Leuven C1-project Macchina.
References


