Abstract

Large detection datasets have a long tail of lowshot classes with very few bounding box annotations. We wish to improve detection for lowshot classes with weakly labelled web-scale datasets only having image-level labels. This requires a detection framework that can be jointly trained with limited number of bounding box annotated images and large number of weakly labelled images. Towards this end, we propose a modification to the FRCNN [39] model to automatically infer label assignment for objects proposals from weakly labelled images during training. We pose this label assignment as a Linear Program with constraints on the number and overlap of object instances in an image. We show that this can be solved efficiently during training for weakly labelled images. Compared to just training with few annotated examples, augmenting with weakly labelled examples in our framework provides significant gains. Compared to few annotated examples, augmenting with weakly labelled examples in our framework provides significant gains. We demonstrate this on the LVIS dataset (3.5% gain in AP) as well as different lowshot variants of the COCO dataset. We provide a thorough analysis of the effect of amount of weakly labelled and fully labelled data required to train the detection model. Our DLWL framework can also outperform self-supervised baselines like omni-supervision [37] for lowshot classes.

1. Introduction

Object detection models have made drastic progress on standard datasets like COCO [31] and PASCAL VOC [11] with thousands of object instances per class. However, as we move towards larger datasets like LVIS [18], we encounter lowshot classes with fewer than ten bounding boxes. On the other hand, there are huge web-scale datasets [50] with image-level labels for large number of classes, but without any bounding boxes. It is lucrative to leverage this information to improve detection for lowshot classes.

Weakly supervised object detection (WSOD) [3, 62, 53] is a line of work that uses only image-level class labels to train detection models. However, the performance of these models are significantly lower than their fully supervised counterparts. In our work, we investigate a more practical middle-ground. We use the few bounding box annotations for lowshot classes in detection datasets in conjunction with a large number of weakly labelled images (Fig. 1). Intuitively, the bounding boxes from the fully supervised examples can lead to better localization compared to WSOD models. Similarly, the class supervision from weakly supervised examples could help in decreasing the classification error in object detection.

We propose a framework to improve Detection for Low-shot classes with Weakly Labelled (DLWL) data. To achieve this, we would like to enable a standard detection model like FRCNN [39] to leverage additional weakly labelled images to improve detection for these classes.

Figure 1: Large scale detection datasets have a long-tail distribution with many lowshot classes. We adopt FRCNN [39] to leverage additional weakly labelled images to improve detection for these classes.
each proposal during training can be used to identify its label. However, in the absence of additional constraints, this can lead to faulty label assignments. Hence, we introduce bounds on the spatial distribution and number of instances for an object in the image, and pose this as a Linear Program (LP). We also show, how this LP can be solved efficiently at every iteration. These estimated labels can then be used as pseudo ground-truth to train the model. With this simple modification to the label assignment process, the model can be trained with both forms of supervision without any other changes to the model architecture.

The idea of using additional data to improve detection has also been explored in past works [37, 12]. These approaches mine additional bounding boxes from a larger dataset using the noisy estimates from the initial low-shot model. On the other hand, we train jointly with all the data. Further, we allow the model to use object-specific constraints and progressively better predictions during training to infer bounding boxes for the weakly labelled images.

The main contributions of our work are two fold: (a) we enable FRCNN model to be trained with both forms of supervision by proposing a LP based framework that assigns labels to proposals in weakly labeled images, and (b) we present a thorough analysis of the effect of augmenting low-shot classes with weakly labelled examples. To this end, we present three sets of experiments. (a) We first demonstrate results on a simulated low-shot setup for COCO dataset. We observe a significant boost in mAP (> 5%) by augmenting low-shot classes (10 images per class) with weakly labelled images. We also observe a gain compared to self-training baselines [37]. Additionally, we show the effect of the amount of low-shot data as well as weakly labelled data. (b) We augment a real world low-shot dataset LVIS [18] with noisy weakly labelled examples from web-scale dataset YFCC100M [50] and observe a 3.5% gain for the rare classes without any additional annotations. (c) We also evaluate our model in an extreme setting without any bounding box labels and demonstrate comparable performance to the state-of-the-art WSOD models.

2. Related work

Augmenting with additional data: Semi-supervised approaches [28, 6, 40, 37] are widely used to train models with additional unlabelled data. Self-training methods uses predictions from an initial model to annotate additional data and then retrain the model [40, 37]. In particular, omni-supervision [37] showed that self-training can lead to modest performance gains in the high-shot regime. In low-shot regime, predictions from the initial model are noisy, and this could adversely affect self-training methods. We handle this noise through jointly training with weakly and fully labelled data with bounding box labels inferred during training, rather than only relying on initial predictions.

More recently, NOTE-RCNN [12] iteratively mined high-confidence examples from a collection of weakly labelled data. While their setup is similar to ours, they require multiple rounds of training. Other recent works [58, 51] utilize correlation between high-shot source classes and low-shot target classes to improve fine-grained detection. These approaches are geared towards selecting better examples and sharing information between classes. These ideas are complementary to our work and can be used in conjunction with our model.

YOLO9000 [38] also uses classification datasets by using the most confident prediction in each image as pseudo ground truth. We generalize this notion by allowing multiple object instances per image along with image-level constraints. Another related work [34] proposes a constrained convolutional network for weakly supervised segmentation with an alternative convex optimization based algorithm. Along similar lines, we present a simpler linear program with bounding box constraints for object detection.

Learning with weak supervision: Weakly supervised object detection has been studied extensively to train detection models only with image-level labels [7, 27, 23, 35, 44, 57, 33, 48, 45, 65, 16, 61, 46, 64, 63, 9, 43, 62]. Notably C-MIL [53] and Gao et al. [15] obtain good results on PASCAL VOC [11], by jointly training with an image classification and object detection loss. Prednet [1] introduces a new dissimilarity based objective function. These methods are focused on learning a model under stringent conditions where bounding box labels are completely absent. Our work provides extensive results for a more practical setup where at least a few bounding box labels are available. It is comparatively simpler and only requires a small change to FRCNN, compared to WSOD methods with specialized architectures.

Low-shot object localization: Many recent works [21, 32, 56, 55, 20, 24, 10, 41, 22] have also developed specialized techniques to improve low-shot object localization. For instance, [22] uses attention and context guided learning to improve low-shot semantic segmentation. Meta-learning with knowledge transfer has also been used to improve low-shot models [56]. Alternatively, we explore the addition of weakly labelled examples to improve low-shot detection.

3. Approach

FRCNN [39] is a widely used model for fully supervised object detection. It consists of a region proposal network (RPN) which generates object proposals for an image, followed by region of interest (ROI) components (ROI-align and ROI-head). ROI-align aggregates features from the proposals, while the ROI-head assigns class labels to the proposals and fine-tunes their co-ordinates. This is illustrated in Fig. 2 with solid lines. In this section, we generalize this FRCNN model to enable training with both fully la-
Figure 2: Overview of our DLWL framework that uses both weakly labelled and strongly labelled examples to train the FRCNN model. If the provided example has bounding box labels, then the standard green module with IoU matching is used to assign labels to proposals, else a linear program with constraints is used to infer the proposal labels as shown in the yellow module with dotted lines. The linear program is explained in more detail in Sec. 3.

belled and weakly labelled data. We refer to our framework as DLWL (Detection for Lowshot classes with Weakly Labelled data).

We consider a large-scale dataset like LVIS [18] composed of both highshot classes (with large number of bounding box annotations) and lowshot classes (with very few bounding box annotations). An inherent advantage with such a mixture of classes is that the large number of bounding boxes in the highshot classes can help to learn the class-agnostic parts of the network like the RPN, leading to overall better object detection. We augment such a dataset with additional weakly labelled examples for the lowshot classes.

More formally, let the training images be given by $X = X_f \cup X_w$ with both fully supervised ($X_f$) and weakly supervised ($X_w$) examples. We initially consider the standard form of weak supervision that provides only image-level class labels without bounding boxes. Later, we also discuss the use of the actual number of bounding boxes per class in an image as additional weak supervision.

### 3.1. Label assignment for weak examples

The ROI head in the FRCNN assigns class scores to each proposal generated from the RPN. This is in turn used to compute a classification loss during training. The main challenge while training with a weakly labelled image is that the class labels for the proposals are unknown. For a fully labelled image, these are obtained by aligning the proposals with the labelled bounding boxes in the image. This is illustrated as the (Intersection over Union) IoU matcher module in Fig. 2, where proposals having an IoU greater than a threshold with a labelled box are assigned the corresponding class-label. This alignment is not possible for weakly labelled images due to absence of bounding box labels. We solve this problem through an optimization based label assignment module (shown in yellow in Fig. 2).

One simple approach is to select the proposal with the highest score for each weakly labelled class as the positive bounding box for that class. However, if multiple instances of the object are present in the image, we run the risk of not associating labels with all instances. Also this does not exploit some intrinsic constraints in the image such as different instances of an object should not overlap. We overcome this by including these constraints during label assignment.

More formally, we consider an image $x \in X_w$, with $C$ weak labels. At any given iteration, let $S$ be the matrix of all class scores assigned by the ROI-head to the $P$ proposals. We consider a $P \times (C + 1)$ sub-matrix $S_C$ denoting the scores for $P$ proposals corresponding to the $C$ weak labels in the image and the background class $C + 1$, such that $s_{pc}$ is the score of the $p^{th}$ proposal for $c^{th}$ weak class in the image. We wish to assign labels to each of the proposals. Let this label assignment be denoted by the binary matrix $Y \in \{0, 1\}^{P \times C+1}$. In C-EM [5], the label for each proposal would be inferred by solving the following optimization problem:

$$Y = \arg\max_Y \quad \text{Tr} \left( S_C^T Y \right),$$

s. t. $Y_1 = 1, \quad \sum_p y_{pc} \geq 1, \quad \forall \ c \leq C$,

where $\text{Tr}(\cdot)$ represents the trace of a matrix. The first constraint ensures that every proposal is assigned a label and
the second constraint ensures that every weak label in the image is assigned to at least one proposal.

In order to improve the label assignment during training, we extend this optimization problem with additional constraints. In particular, we add constraints on the number of chosen boxes per class, as well as their spatial distribution. For every object class, we assume prior knowledge about the average number of boxes per image. This can often be obtained from the low-shot dataset or even set to a fixed number for all classes in a dataset as we show in the experiments section. For every class \( c \), let this number be denoted by \( N_c \). This leads to a stricter constraint in Eq. 1.

\[
\sum_p y_{pc} = N_c, \quad \forall \ c \leq C.
\]  

(2)

Additionally, we want to ensure that for every class, we chose boxes that do not overlap significantly. This ensures that multiple instances of the object are well spread out. To achieve this, we first cluster all the proposals in the image based on their IoU. In practice, we use agglomerative clustering with a threshold resulting in \( H \) clusters. The number of clusters are determined by the threshold and vary with the image. Let the clusters be denoted by \( \{h_1, \ldots, h_H\} \).

We can now add an additional constraint to ensure that every cluster has only one instance of an object:

\[
\sum_{p \in h_i} y_{pc} \leq 1, \quad \forall \ c \leq C, \quad 1 \leq i \leq H
\]  

(3)

The effect of label assignment with these constraints is shown in Fig. 3. As seen, the LP does not prohibit a cluster from containing instances of different objects.

### 3.2. Bootstrapping with weaker model

A common problem when training with weakly labelled examples is that the model can get stuck in a bad local minima in the initial stages of training [51], since the prediction of the model is initially unreliable. We get around this problem by first training a low-shot model (\( M_{low} \)) without any weakly labelled data and using the labels predicted from (\( M_{low} \)) to augment the predictions from the current model. In other words, we replace the score matrix \( S \) with a weighted combination \( \lambda S + (1 - \lambda) S_{init} \), where \( S_{init} \) is a matrix obtained from the predictions of the low-shot model (\( M_{low} \)) and \( 0 \leq \lambda \leq 1 \).

In the initial stage, the score for low-shot classes can be quite low and \( S_{init} \) helps in bootstrapping the training. We also anneal \( \lambda \) to 0 as the model starts training, since the confidence of the model for low-shot classes increases over time. Please refer to supplementary material for the details on annealing.

In order to compute \( S_{init} \), we use 100 detections per image from (\( M_{low} \)). At a given iteration, for each proposal \( p \) in the image, we find the detection bounding box from the low-shot model with the highest overlap. If the overlap is higher than 0.7, we assign the class scores of the detection box to the \( p^{th} \) row in \( S_{init} \). If overlap is below this threshold, we set the value corresponding to background in the \( p^{th} \) row to 1.

Figure 3: An example illustrating label assignment with the linear program. The highest scoring proposals for two weak classes car and motorcycle are shown in red and blue respectively in the top-left image. The different clusters of proposals are show in the bottom-left image, where each color denotes a unique cluster. The final label assignment from the LP is shown in the right image.

### 3.3. Count based supervision

Weak supervision for detection typically refers to the setup, where we only know the set of object classes present in an image. Gao et al. [13] studied count based supervision as an alternative form of weak supervision, where the number of instances of each class in an image is annotated. This is cheaper to obtain compared to other forms of annotation like one-click [2] and can benefit weakly supervised detection. Interestingly, this weak supervision fits naturally into our framework. Instead of using a rough guess or prior from the dataset, if the exact count of an object is known, we could use this as a stricter constraint in Eq. 2. In other words, count supervision would provide us the true value of \( N_c \) in each image.

### 3.4. Training Details

**LP optimization:** In order to solve the optimization problem in Eq. 1 with additional constrains in Eqs. 2 and 3, we first relax the binary constraint. However, the resulting LP is very expensive to solve with standard LP solvers. This has to be carried out at every iteration when we encounter a weakly labelled image. We get over this problem by observing that the optimization separates into separable constraints that can be solved efficiently using ADMM [4] (shown in supplementary).

**RPN and bounding box regression:** We disable the RPN loss and bounding box regression losses for all weakly labelled examples. These losses are useful only for fully-supervised images, where the exact bounding box coordinates are known during training.
4. Experiments

The main contribution of our work is a simple but effective change to the FRCNN framework that enables training of an object detection model with a mixture of fully-supervised and weakly-supervised data. We show that this is particularly beneficial for datasets with low-shot classes. We first demonstrate this through controlled experiments on low-shot variants of the COCO [31] dataset. We use the 2017 version of the dataset unless otherwise specified. We also show the practical utility of our model by augmenting the rare classes of the LVIS [18] dataset with weakly labelled images mined from YFCC100M [50] without using any additional annotations. Finally, we show that even under the stricter weakly supervised regime (without any low-shot data), our model can achieve comparable performance to existing weakly supervised object detectors.

4.1. Experimental Set-up

**Implementation details:** We fix ImageNet [8] pre-trained ResNet-50 [19] with Feature Pyramid Network (FPN) [30] as the backbone for FRCNN model for all the low-shot experiments unless otherwise specified. We trained all the models for $90K$ iterations with a batch size of 16 and the standard learning rate schedule used in [18]. We resized images to a minimum edge size of 800 and used horizontal flip for data augmentation, unless otherwise stated. Also, as recommended in [18], we use square-root upsampling in all our low-shot experiments to handle the data imbalance across classes. We use $\lambda = 0.5$ and exponentially decay it to 0 by the end of training.

**Evaluation:** We report the standard COCO metrics like AP (averaged over IoU thresholds) and AP$_{50}$.

**Baselines:** We refer to our model as DLWL and compare results with the following models:

- **lowshot-only:** We train a model on fully supervised data only with available bounding box annotations.
- **omni-weak:** We also compare our model with omni-supervision approach [37]. To enable fair comparison, we use a slight variant which makes use of the weak labels. In particular, for a given class, we first select the subset of images from the weakly labelled dataset associated with that class and only use this subset to generate additional bounding box annotations using the lowshot-only model. We use the same strategy as [37] to identify a per-class threshold that results in the same average number of bounding boxes per class in the weakly labelled dataset as the original low-shot dataset. If this threshold does not yield bounding boxes in a weakly labelled image, we use the highest scoring detection for that class in the image. We then train a FRCNN model with original plus new annotations. ¹

1 We also tried another variant of this model where we used only one an-

4.2. COCO low-shot experiments

**Dataset Construction:** We split the 80 COCO classes into a set of 70 highshot classes and 10 lowshot classes (chosen randomly, list in supp. section). We create a variant of COCO-train dataset where we have large number of images for the highshot classes and only a few images from the lowshot classes. In particular, we create the following two subsets of COCO-train data:

- **COCO-N-strong:** In this subset, we retain only $N$ training images from each of the 10 lowshot classes, and all images from highshot classes that do not contain any instance of a lowshot class. Note that since both lowshot and highshot classes can co-occur together in an image, we might exclude some images having highshot objects too. We vary the value of $N$ to create different $N$-shot subsets.

- **COCO-N-weak:** The remaining images from COCO-train that are excluded from COCO-N-strong are used to form a weakly labelled dataset. In this subset, we only retain image-level labels without bounding boxes for the images. Note that this subset is dominated by objects belonging to the 10 lowshot classes. We refer to this as COCO-N-weak.

We evaluate on 5000 images in the COCO validation set.

**Varying amounts of full-supervision:** We now explore the effect of varying the amount of lowshot data. For each value of $N$, we use the corresponding N-shot split COCO-N-strong as the fully-supervised training dataset. We augment this data with weakly labelled examples from COCO-N-weak for the models that use weakly labelled data. We fix $N_c$ to be the average number of bounding boxes for class $c$ in the fully supervised lowshot dataset COCO-N-strong. Note that in practice, this does not correspond to an integer value for $N_c$. We do stochastic rounding during training for each sample, so that across multiple iterations the average converges to the fractional value. For the specific case of $N = 0$, which is the weakly supervised setup, we set $N_c$ to 3 by computing the mean of average number of bounding boxes for the 70 high shot classes.

From Fig. 4, we immediately observe that addition of small number of fully labelled examples ($N = 10$) leads to a huge improvement compared to using only weakly labelled data ($N = 0$) for all the models. This shows the benefit of training with at least a handful of fully labelled examples. In Fig. 5 we see better localization as the amount of supervision increases.

Compared to lowshot-only, we show significant gains for different $N$ values (8.8% for $N = 10$). Our model trained with only $N = 20$ surpasses the lowshot-only model trained using $N = 200$ examples.

We also compare our approach with the strong omni-weak baseline which also leverages weakly labelled data.

¹We also tried another variant of this model where we used only one an-
In the lowshot regime of $N < 50$, our model outperforms "omni-weak" by large margins. For $N = 10$, our approach achieves 17.1% AP compared to 13.1% achieved by "omni-weak". The predictions from "lowshot-only" model used by "omni-weak" to get pseudo ground truth annotations are very noisy in the lowshot regime that impacts its performance. On the other hand, additional constraints used by our approach and dynamic label assignment during the training helps to rectify the faulty label assignments to proposals. In the highshot regime of $N \geq 50$, we observe diminishing returns from our model compared to "omni-weak". The performance of "lowshot-only" model improves with higher values of $N$, which in turn leads to better performance for the "omni-weak" model as well.

We also report the performance for the 70 highshot classes at different values of $N$ in Tab. 1. We note that the AP for these classes doesn’t change much with the addition of weakly labelled data for lowshot classes, compared to the fully-supervised "lowshot-only" model.

**Effect of amount of weakly labelled data:** Since the main focus of the work is to leverage weakly labelled images, it is important to analyze the effect of number of such images required. To do so, we fix COCO-10-strong as the fully supervised dataset, augment it with different number of weakly labelled images from COCO-10-weak and then train our model. COCO-10-weak consists of 10k images not present in COCO-10-strong.

Fig. 7 shows the results as we vary the number of weakly labelled examples in increments of 1000. We observe that the performance increases rapidly in the beginning and then saturates around 8k images. Hence, adding up to two orders of magnitude more weakly labelled data still improves the detection performance for the lowshot classes.

**Effect of $N_c$:** We recommend setting $N_c$ to the average number of object instances per image for each lowshot class in the COCO-N-strong dataset. We use $N_c = \text{avg}$ to depict this setting. We now experiment with different strategies for deciding $N_c$. The simplest approach is to fix it to the same integer value for all the classes. We use COCO-N-strong as fully-supervised dataset and augment it with COCO-N-weak. Tab. 2 shows results for $N = 0$ and 10.

We observe that performance first increases and then decreases as we vary $N_c$. Note that at $N_c = 1$, the model is trained without the constraint in Eq. 2 since only one instance per class is chosen. This leads to a performance drop, showing the importance of constraints introduced in our framework. By increasing $N_c$, there is a higher chance for one of the estimated bounding boxes to cover the true object instance in the image, leading to higher recall. However, at very high values this also leads to lower precision due to increase in false positives. Hence, the value of $N_c$ needs to be close to the true count of object instances for each class in an image. Our strategy ($N_c = \text{avg}$) of estimating $N_c$ from the lowshot dataset itself works the best.

**Count supervision:** The choice of $N_c$ can lead to a significant variation in model performance. Hence, we look into a form of weak supervision where $N_c$ is known for every image. We assume we know the count of each class for all the images in the weakly labelled datasets COCO-N-weak. This can directly be used to bound the label assignment to proposals in Eq. 2. We show results at different $N$-shot values with this added supervision in Tab. 3. We observe a nominal gain of 0.7% – 1.6% at low values of $N$. This is fairly cheap and can lead to good gains in the lowshot regime.

### 4.3. Augmenting LVIS with weakly labelled data

The recently released LVIS [18] dataset clearly highlights the need for better lowshot object detection models. The dataset has more than a 100 “rare” classes which have less than 10 bounding box annotations in the training dataset. We attempt to improve the performance of this model by augmenting the rare classes with additional weakly labelled images. Unlike the controlled settings for COCO in Sec. 4.2, we do not have a clean source of...
### Figure 5: Sample images with detections from our model and omni-weak model at $N = 5$ and $N = 10$. We see better localization as $N$ increases for both models.

### Figure 6: Effect of varying the amount of weakly labelled data used in addition to the lowshot COCO-10-strong dataset to train our model.

### Figure 7: Samples from YFCC100M for a few rare classes that are part of our weakly labelled dataset. For each image, we show the detections from the initial lowshot-only model in blue. The images that are noisy and do not have any instance of the corresponding object are highlighted in red. We notice a significant fraction of noisy images as well as missing or wrongly localized objects from the initial model.

weakly labelled images for LVIS. Hence, we look to another dataset, YFCC100M [50] to augment the rare classes.

**YFCC100M as weakly labelled dataset:** YFCC100M has 100M images along with the noisy hashtags which can be treated as weak image-level labels. For each rare class, we use the names\(^2\) associated with it to find the matching hashtags. We then use the images tagged with the matched hashtags to augment the rare class with weakly labelled data. However, a good fraction of classes have no corresponding tags. Hence, we also use the nearest neighbors to gather additional examples. Specifically, we use the cropped bounding boxes (expanded with additional context similar to [17]) from each rare class to retrieve at most 1000 nearest neighbors from YFCC100M. We then include these retrieved images in the weakly labelled set also. Please refer to supplementary material for details.

However, this set of weakly labelled images can be very noisy due to erroneous tags and nearest neighbors returning irrelevant images particularly for small objects. In order to reduce the noise, we use the original model trained on the LVIS dataset (lowshot-only) to filter out images that have no detections above a threshold of 0.001 for the rare classes. After this filtering, we retain a maximum of 500 images per class that have the highest detection scores. Note that this is a very low threshold and is aimed at getting a higher recall. The resulting images are used to build a weakly labelled dataset YFCC100M-weak. This dataset still has significant amount of noise (Fig. 7).

**Image deduplication:** We use nearest neighbors approach to remove all images from YFCC100M-weak, that were near duplicates of images from the LVIS validation set to avoid

\(^2\)Each LVIS class is a WordNet synset with multiple associated names.
<table>
<thead>
<tr>
<th>Backbone</th>
<th>Method</th>
<th>AP-r</th>
<th>AP</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResNet-50</td>
<td>lowshot-only</td>
<td>10.84</td>
<td>21.88 ± 0.24</td>
</tr>
<tr>
<td>ResNet-50</td>
<td>lowshot mask* [18]</td>
<td>11.15</td>
<td>23.32 ± 0.21</td>
</tr>
<tr>
<td>ResNet-50</td>
<td>omni-weak [37]</td>
<td>12.88</td>
<td>21.85 ± 0.22</td>
</tr>
<tr>
<td>ResNet-50</td>
<td>DLWL</td>
<td>14.21</td>
<td>22.14 ± 0.16</td>
</tr>
<tr>
<td>ResNet-101-32x8d</td>
<td>lowshot-only</td>
<td>12.73</td>
<td>23.75 ± 0.41</td>
</tr>
<tr>
<td>ResNet-101-32x8d</td>
<td>omni-weak [37]</td>
<td>16.03</td>
<td>24.73 ± 0.73</td>
</tr>
<tr>
<td>ResNet-101-32x8d</td>
<td>DLWL</td>
<td>17.36</td>
<td>25.07 ± 0.10</td>
</tr>
</tbody>
</table>

Table 4: Performance of our model in comparison to different baselines on LVIS dataset after augmenting with weakly labelled data from YFCC-100M. AP-r is the average precision for rare classes and AP is the average precision for all classes. * Note that lowshot mask uses a mask-RCNN with segmentation masks as additional supervision unlike other methods in the table.

corruption between train and test data.

**Evaluation:** We report the average (and standard deviation) of AP-r and AP over 3 training runs. AP-r is the average precision for rare classes and AP is the average precision for all classes, following the convention from [18].

**Results:** From Tab. 4, we observe that adding additional weakly labelled examples provides a gain for both omni-weak and our model. Further, we see that our model which leverages image-level constraints to handle noise during training outperforms omni-weak. More interestingly, the performance of our model on the rare classes surpasses that of mask-RCNN which uses additional supervision from segmentation masks, thus clearly demonstrating the power of leveraging weakly labelled data. Extending our approach to mask-RCNN is an interesting future direction.

### 4.4. Additional weakly supervised experiments

The main focus of our work is to provide a model that can be jointly trained with both weakly labelled and fully labelled examples, and is not specifically geared towards stand alone weakly supervised detection. However, we present results on weakly supervised benchmarks through some simple modifications to our setup.

**Bootstrapping:** Unlike the lowshot case (Sec. 3.2), we do not have an initial fully supervised model for bootstrapping. Hence we first train another weakly supervised model WSDDN [3] augmented with contextual pooling [25], and use its predictions for bootstrapping. We chose WSDDN due to its simplicity; using more sophisticated models could lead to better performance. Refer to supplementary for details.

**Datasets:** We report results for the PASCAL VOC07 as well as COCO14 (2014 version of COCO[31]) datasets. We train on the train-split and evaluate on the full val-split, when reporting results for COCO14.

**Training Details:** We use VGG-16 backbone for all experiments. Following the settings in [47], the WSDDN model used for bootstrapping was trained with MCG [36] proposals for PASCAL VOC07 and selective search [52] proposals for COCO14. Once WSDDN is trained, we use the detections from this model for initializing $S_{init}$ and train a FR-CNN using our method. In line with previous works, as a last-step we also retrain a FR-CNN using the predictions of our model as pseudo ground-truth. For PASCAL-VOC07, we trained all models for 20 epochs with an initial learning rate of $5e^{-3}$ which was dropped to $5e^{-4}$ after 10 epochs. We used the same learning rate schedule as described in Sec. 4 for COCO14. During training, we used scale jitter with 5 different scales and horizontal flip.

**Results:** The results are shown in Tab. 5 for PASCAL VOC07 and COCO14. We see that the performance of our model is comparable to state-of-the-art weakly supervised methods. Dedicated weakly supervised models [26, 53, 15] have specialized architectures and loss functions for fine tuning predictions on the weakly labelled examples. These methods are complimentary to the simpler change to FR-CNN proposed in our work. We also note that more recent models [14, 59, 29, 42] have reported a considerable gain in detection performance by guiding the network with additional segmentation signals from weakly supervised segmentation or superpixel straddling. Since our aim is to provide a simple approach to leverage weakly labelled examples in a standard FR-CNN model, the use of segmentation signals is beyond the scope of this work.

<table>
<thead>
<tr>
<th>Model</th>
<th>AP_{50}</th>
</tr>
</thead>
<tbody>
<tr>
<td>OCR [45]</td>
<td>47.0</td>
</tr>
<tr>
<td>PCL [47]</td>
<td>48.8</td>
</tr>
<tr>
<td>MELM [54]</td>
<td>47.3</td>
</tr>
<tr>
<td>Weak-RPN [49]</td>
<td>50.4</td>
</tr>
<tr>
<td>Yang et al. [59]</td>
<td>51.5</td>
</tr>
<tr>
<td>PGE [26]</td>
<td>52.1</td>
</tr>
<tr>
<td>C-MIL [53]</td>
<td>52.3</td>
</tr>
<tr>
<td>Gao et al. [13]</td>
<td>52.6</td>
</tr>
<tr>
<td>Prednet [1]</td>
<td>52.9</td>
</tr>
<tr>
<td>DLWL</td>
<td>52.0</td>
</tr>
</tbody>
</table>

(a) PASCAL VOC07 results

Table 5: Weakly supervised object detection results for PASCAL-VOC07 and COCO14 with VGG-16 backbone.

5. Conclusion

We introduced a framework to improve Detection for Lowshot classes with Weakly Labelled data (DLWL). We showed how this can be used to train FR-CNN models with both weakly supervised and fully supervised images, by extending the proposal label assignment process in FR-CNN to handle both forms of supervision. We formulated the label assignment for weakly labelled images as a Linear Program (LP). The LP imposed constraints on the number of instances of an object in an image and ensured non-overlap of multiple instances of the same object. We demonstrated the effectiveness of our approach on the LVIS dataset and lowshot variants of COCO dataset. For future work, we could extend to other forms of weak supervision such as one point annotation. Another interesting direction would be to train mask-RCNN in our framework as well.
References


[46] Abhilash Srikantha and Juergen Gall. Weak supervision for detecting object classes from activities. Computer Vision and Image Understanding, 156:138–150, 2017. 2


