(Supplementary) DLWL: Improving Detection for Lowshot classes with Weakly Labelled data

Vignesh Ramanathan
Facebook
vigneshr@fb.com

Rui Wang
Facebook
ruiw@fb.com

Dhruv Mahajan
Facebook
dhruvm@fb.com

1. Linear Porgram Optimization

We are interested in solving the following optimization problem during every training iteration for a weakly labelled image.

$$\mathbf{Y} = \text{argmax}_{\mathbf{Y}} \ \text{Tr}\left(\mathbf{S}_C^T \mathbf{Y}\right), \tag{1}$$
$$\text{s. t.} \quad \mathbf{Y1} = \mathbf{1},$$
$$0 \leq \mathbf{Y} \leq 1,$$
$$\sum_p y_{pc} = N_c, \qquad \forall \ c \leq C,$$
$$\sum_{p \in h_i} y_{pc} \leq 1, \qquad \forall \ c \leq C, \quad 1 \leq i \leq H.$$

With about 1000 proposals per image and with even $3-4$ different objects each with $2-3$ instances in the image, this linear program can become very time consuming to solve with typical LP solvers. Instead, we leverage the inherent separable nature of the variables to use ADMM to solve the equations. We first observe that in the absence of the constraint $\mathbf{Y1} = \mathbf{1}$, the optimization problem can be attacked by solving for each column of $\mathbf{Y}$ separately. And as we show later, solving for each column separately in absence of this constraint can be done very efficiently. We can now use this idea to define the augmented Lagrangian for the problem as follows:

$$\mathbf{Y} = \text{argmin}_{\mathbf{Y},\mathbf{z}} - \text{Tr}\left(\mathbf{S}_C^T \mathbf{Y}\right) + \mathbf{z}^T\left(\mathbf{Y1} - \mathbf{1}\right) + \frac{\rho}{2}\|\mathbf{Y1} - \mathbf{1}\|^2,$$
$$\text{s. t.} \quad 0 \leq \mathbf{Y} \leq 1,$$
$$\sum_p y_{pc} = N_c, \ \ \forall \ c \leq C,$$
$$\sum_{p \in h_i} y_{pc} \leq 1, \ \ \forall \ c \leq C, \quad 1 \leq i \leq H,$$

where $\mathbf{z}$ is a dual variable and ρ is the dual step-length (optimization parameter). We set ρ to be 1000 in our work. In line with the ADMM algorithm, we can now solve for each column of $\mathbf{Y}$ one at a time, then optimize for $\mathbf{z}$ and then repeat the process again till convergence. In practice we observe convergence within less than 5 steps. More specifically, let $\mathbf{y}_c$ be the c^{th} column of $\mathbf{Y}$ and s_c be the c^{th} column of $\mathbf{S}_C$.

At each iteration $k+1$, we solve for each column of $\mathbf{Y}$ sequentially starting from $\mathbf{y}_1^{k+1}$ to $\mathbf{y}_{C+1}^{k+1}$. In particular, we solve the following optimization problem:

$$\mathbf{y}_c^{k+1} = \text{argmin}_{\mathbf{y}_c} \ - \mathbf{s}_c \mathbf{y}_c + \frac{\rho}{2}\|\mathbf{y}_c + \mathbf{b}_c^k + \mathbf{z}^k\|^2, \tag{2}$$
$$\text{s. t.} \quad 0 \leq y_{pc} \leq 1,$$
$$\sum_p y_{pc} = N_c, \ \ \text{if} \ \ c \leq C$$
$$\sum_{p \in h_i} y_{pc} \leq 1, \ \ \forall \ \ 1 \leq i \leq H, \ \ \text{if} \ \ c \leq C$$

where $\mathbf{b}_c^k = \mathbf{Y^k 1} - \mathbf{1} - \sum_{i \leq c} \mathbf{y}_c^k + \sum_{i<c} \mathbf{y}_c^{k+1}$. Then finally we update $\mathbf{z}^{k+1}$ as follows:

$$\mathbf{z}^{k+1} = \mathbf{z}^k + \left(\mathbf{Y}^{k+1}\mathbf{1} - \mathbf{1}\right) \tag{3}$$

The overall optimization problem then reduces to solving Eq. 2 efficiently for each weakly labelled class in the image. We first look into the case when $c = C+1$. This does not involve the last two constraints in Eq. 2 and reduces to:

$$\mathbf{y}_{C+1}^{k+1} = \text{argmin}_{\mathbf{y}_{C+1}} - \mathbf{s}_{C+1}\mathbf{y}_{C+1} + \frac{\rho}{2}\|\mathbf{y}_{C+1} + \mathbf{b}_{C+1}^k + \mathbf{z}^k\|^2,$$
$$\text{s. t.} \quad 0 \leq y_{pC+1} \leq 1,$$

which is straightforward to solve, since the objective function and constraint separate out in terms of each individual variable in the vector $\mathbf{y}_{C+1}$. This has a deterministic solution obtained by minimizing the objective and projecting to the simplex $0 \leq y_{pC+1} \leq 1$.

Moving on to the case when $1 \leq c \leq C$, we first observe that if the objective in Eq. 2 was linear instead of quadratic, the problem would reduce to a simple knapsack problem

with a greedy solution. This leads to the idea of using a conditional gradient descent method like Frank Wolfe [2] to optimize the problem which replaces the objective with a first order linear approximation of the objective. In practice, we observe that using 1 or 2 steps of Frank-Wolfe [2] is sufficient to obtain a good solution for the problem. We initialize the Frank Wolfe algorithm by choosing the top N_c clusters with the highest scores for class c and setting $y_{pc} = 1$ for the highest scoring proposal in each of the clusters.

2. List of lowshot COCO classes

In our experiments on COCO, we had split COCO classes into a set of 10 lowshot classes and 70 highshot classes. The 10 lowshot classes were chosen at random and kept fixed for all experiments. The 10 classes are:*cow,motorcycle,knife,remote,hot dog,skateboard,dog,truck,cup,orange*.

3. Bootstrapping with weaker model

3.1. Annealing the bootstrapping weight

We anneal the weight λ over time to 0 gradually. We tried different annealing techniques and found exponential decay to work the best as shown below:

$$\lambda_t = \lambda_0 \frac{e - e^{\frac{t}{T}}}{e - 1},$$

where λ_t is the weight at iteration t, λ_0 is the initial weight and T is the total number of training iteration.

3.2. Bootstrapping for weakly supervised model

As explained in the main draft, for weakly supervised experiments we do not have an initial lowshot model to use for bootstrapping. Hence, we instead train an initial weakly supervised model WSDDN [1] and use the scores from this model for bootstrapping. However, unlike FRCNN we need to provide external proposals to train a WSDDN model. Hence we use Selective Search [4] proposals and MCG proposals [3] for PASCAL VOC and COCO14 datasets respectively to train the WSDDN model. Once the WSDDN model is trained, we extract 100 detections per image along with corresponding scores using WSDDN and the above proposals for all training images. The scores from these detections are then used to bootstrap training of the FRCNN model with our DWL framework.

Note that we do not need external proposals to train the FRCNN model. However, since there are no highshot classes and associated fully labelled images to guide the RPN part of the network, we *do not* set the RPN loss to zero while training the FRCNN unlike the lowshot setup. Rather the RPN is also trained with the bounding box labels inferred using our DWL method.

4. Augmenting with images from YFCC100M

For every "rare" class in LVIS, we use the following scheme to obtain weakly labelled images from YFCC100M:

1. Each class in LVIS is associated with a set of different synonyms. We use these synonyms to obtain images from YFCC100M which are tagged with one of the synonyms. Apart from the actual synonym, we also use plurals of each synonym to obtain images from YFCC100M.

2. We also use nearest neighbor based expansion to obtain additional images from YFCC100M. In particular for every annotated bounding box corresponding to a rare class in LVIS, we expand its height and width by 1.2 times and crop the resulting expanded bounding box. This provides an image of the rare class object with some additional context. We use each cropped bounding box to obtain 100 nearest neighbors from YFCC100M using cosine similarity. We only retain images which have a cosine distance less than 0.25. This results in thousands of images for each rare class.

3. We combine the images obtained from both steps above to form an expanded set of images for each rare class. For every rare class, we only retain 500 images from this set based on scores from an initial lowshot model. Specifically, we rank all images corresponding to a rare class by the highest detection score for the rare class obtained from the initial lowshot model. We then only retain the top 500 images for each rare class. This forms the final set of weakly labelled images. Note that the same image can be associated with multiple rare class labels. In addition to the rare class label, we also use the YFCC tags of each image corresponding to other LVIS classes to associate additional weak labels to the image.

References

[1] Hakan Bilen and Andrea Vedaldi. Weakly supervised deep detection networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2846–2854, 2016. 2

[2] Marguerite Frank and Philip Wolfe. An algorithm for quadratic programming. *Naval research logistics quarterly*, 3(1-2):95–110, 1956. 2

[3] Jordi Pont-Tuset, Pablo Arbelaez, Jonathan T Barron, Ferran Marques, and Jitendra Malik. Multiscale combinatorial grouping for image segmentation and object proposal generation. *IEEE transactions on pattern analysis and machine intelligence*, 39(1):128–140, 2016. 2

[4] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013. 2