Object Detection as a Positive-Unlabeled Problem

Yuewei Yang* Kevin J Liang* Lawrence Carin
Duke University

{yuewei.yang, kevin.liang, lcarin}@duke.edu

Abstract

As with other deep learning methods, label quality is important for learning modern convolutional object detectors. However, the potentially large number and wide diversity of object instances that can be found in complex image scenes makes constituting complete annotations a challenging task; objects missing annotations can be observed in a variety of popular object detection datasets. These missing annotations can be problematic, as the standard cross-entropy loss employed to train object detection models treats classification as a positive-negative (PN) problem: unlabeled regions are implicitly assumed to be background. As such, any object missing a bounding box results in a confusing learning signal, the effects of which we observe empirically. To remedy this, we propose treating object detection as a positive-unlabeled (PU) problem, which removes the assumption that unlabeled regions must be negative. We demonstrate that our proposed PU classification loss outperforms the standard PN loss on PASCAL VOC and MS COCO across a range of label missingness, as well as on Visual Genome and DeepLesion with full labels.

1. Introduction

The performance of supervised deep learning models is often highly dependent on the quality of the labels they are trained on [48, 43, 20]. Recent work [42] has implied the existence of "support vectors" in deep learning datasets: hard to classify examples that have an especially significant influence on a classifier's decision boundary. As such, ensuring that these difficult examples have the correct label would appear to be important to the final classifier.

Collecting completely accurate labels for object detection [14, 13, 30, 37, 35, 4, 36], however, can be challenging, much more so than it is for classification data. Unlike the latter, where there is a single label per image, the number of objects in an image is often variable, and objects can come in a large variety of shapes, sizes, poses, and settings, even within the same class. Worse, object detection scenes are often crowded, resulting in object instances that may be occluded. Given the requirement for tight bounding boxes and the sheer number of instances to label, constituting annotations can be very time-consuming. For example, just labeling instances, without localization, required ∼30K worker hours for the 328K images of MS COCO [29], and the airport checkpoint X-ray dataset used in [28], which required assembling bags, scanning, and hand labeling, took well over 250 person hours for 4000 scans over the span of several months. For medical datasets [32, 47, 26, 44], this becomes even more problematic, as highly trained (and expensive) radiologist experts or potentially invasive biopsies are needed to determine ground truth.

As a result of its time-consuming nature, dataset anno-

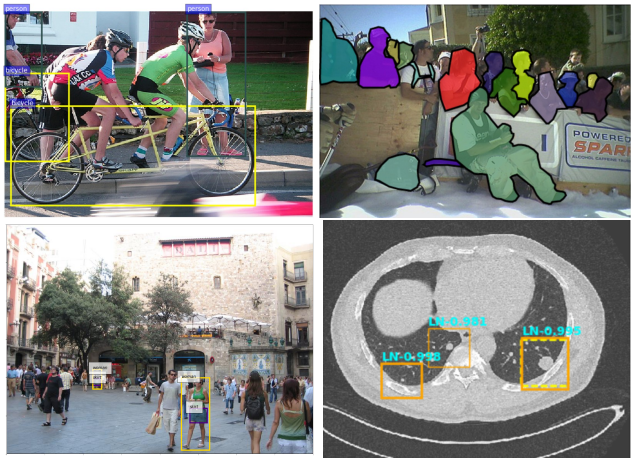


Figure 1: Because of inter- and intra-annotator inconsistencies and the inherent difficulty of instance labeling, the ground truth of object detection datasets can be incomplete. Example images and their ground truth labels shown for (clockwise from top left) PASCAL VOC [12] (missing people and bottles), MS COCO [29] (missing people), DeepLesion [47] (ground truth is the dotted line; two boxes on the left indicate two unlabeled nodules), and Visual Genome [22] (missing people, tree, clothing, etc.).

*Equal Contribution.

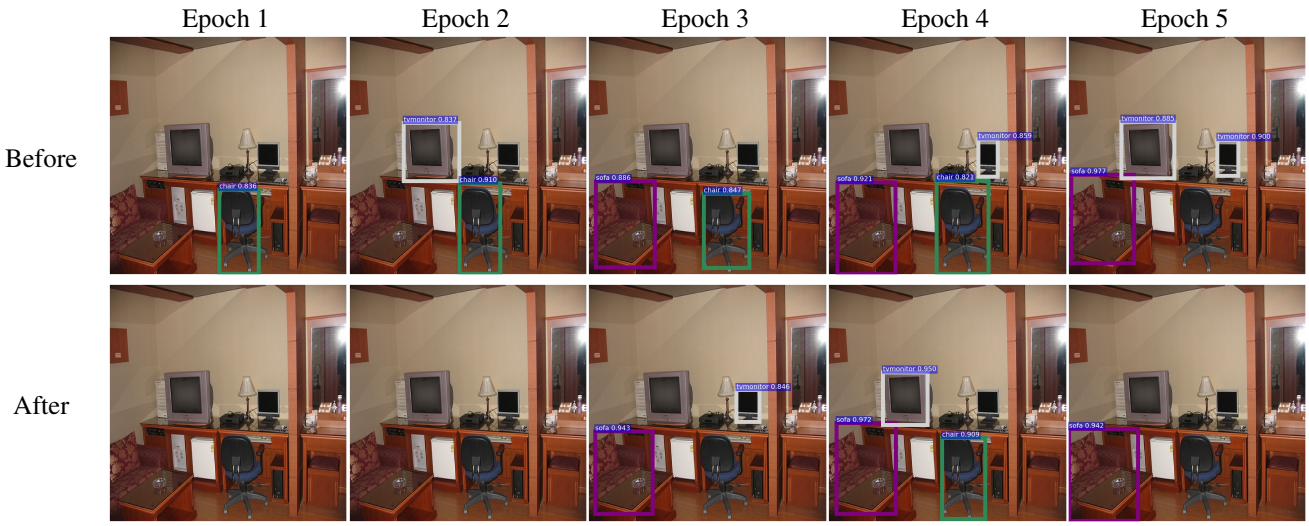


Figure 2: Detections on a PASCAL VOC train set image missing annotations throughout training: only the sofa in the lower left has a label. Each column shows detections directly before (top) and after (bottom) the model is trained on the image shown, for each epoch. While the sofa is consistently detected (purple box) after being learned, the unlabeled objects (2 monitors, a chair) are repeatedly found and then suppressed after being trained upon.

tations are often crowd-sourced when possible, either with specialized domain experts or Amazon's Mechanical Turk, significantly speeding up the data annotation process. In order to ensure consistency, dataset designers establish labeling guidelines and/or have multiple workers label the same image [12, 29]. Regardless, tough judgment-call instances, inter- and even intra-worker variability, and human error can still result in overall inconsistency in labeling, or missing instances entirely. This becomes especially exacerbated when trying to form a larger dataset, like OpenImages [24], which while extremely large, is incompletely labeled.

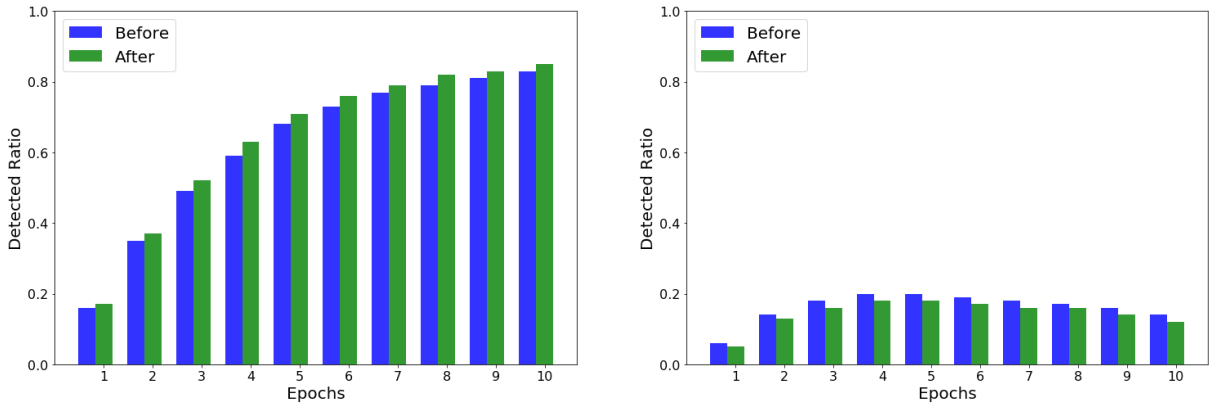
On the other hand, object detection algorithms often use the standard cross-entropy loss for object classification. As a result, implicit to this loss function is the assumption that any region without a bounding box does not contain an object; in other words, classification is posed as a positive-negative (PN) learning problem. While such an assumption may be reasonable for an appropriately accurate ground truth for each image, despite best efforts, this is often not the case in practice due to the previously outlined difficulties of instance annotation. As shown in Figure 1 for a wide array of common datasets, the lack of instance label does not always mean the absence of a true object.

While the result of this characterization constitutes a noisy label setting, it is not noisy in the same respect as is commonly considered for classification problems [48, 43, 20]. The presence of a positive label in object detection datasets are generally correct with high probability; it is the *lack* of a label that should not be interpreted with confidence as a negative (or background) region. Thus, given these characteristics common to object detection data, we propose recasting object detection as a positive-unlabeled (PU) learning problem [7, 5, 27, 11, 21]. With such a perspective, existing labels still implies a positive sample, but the lack of one no longer enforces that the region must be negative. This can mitigate the confusing learning signal that often occurs when training on object detection datasets.

In this work, we explore how the characteristics of object detection annotation lend themselves to a PU learning problem and demonstrate the efficacy of adapting detection model training objectives accordingly. We first illustrate with an empirical study the confusing effect missing labels have on the training process. We then perform a series of experiments to demonstrate the effectiveness of the PU objective on two popular, well-labeled object detection datasets (PASCAL VOC [12] and MS COCO [29]) across a range of label missingness, as well as two datasets with real incomplete labels (Visual Genome [22], DeepLesion [47]).

2. Example Forgetting in Object Detection

In a recent study of training dynamics of neural network classifiers, the authors of [42] defined a "forgetting event" as a training example switching from being classified correctly by the model to being classified incorrectly during training. It was found that certain examples were forgotten more frequently than others while others were never forgotten (termed "unforgettable"), with the degree of forgetting for individual examples being consistent across neural network architectures and random seeds. When visualized, the forgotten examples tend to have atypical or un-



(a) Labeled objects

(b) Unlabeled objects

Figure 3: Detection rates of objects before and after training on their corresponding images for (a) labeled instances and (b) instances with labels withheld during training.

common characteristics (e.g., pose, lighting, angle), relative to "unforgettable" examples. Interestingly, a significant number of "unforgettable" examples could be removed from the training set with only a marginal reduction in test accuracy, if the "hard" examples were kept. This implies that the "hard" examples play a role akin to support vectors in max-margin learning, while easier "unforgettable" examples have little effect on the final decision boundary.

Within the context of object detection datasets, we hypothesize that unlabeled object instances form a similar group of hard examples that are also learned and then forgotten throughout training. Unlike the inter-batch catastrophic forgetting in [42], however, where hard examples are learned while part of the current minibatch and then forgotten while learning other examples, unlabeled samples in object detection are learned from other examples and then *suppressed* after incurring misclassification losses during training (see Figure 2). Unlabeled instances strongly resemble positive examples throughout the rest of the dataset and indeed should be considered as such, but their lack of labels mean that the typical PN classification objective incentivizes learning them as negatives. Given that hard examples have a strong influence on classifier boundaries, having unlabeled examples trained as negatives may prove especially detrimental to training.

We perform a similar study as [42] and investigate forgetting events on PASCAL VOC [12] by tracking detection rates of labeled and unlabeled instances in the training set throughout learning. In particular, an object is considered detected if the detector produces a bounding box with intersection over union (IoU) of at least 0.5 and the classifier is at least 80% confident in the correct class. We track whether or not an object was detected directly before the image it belongs to is trained upon, and then again after the gradients have been applied. These indicator variables are then combined across objects for each epoch and reported as a percentage. While PASCAL VOC does naturally have unlabeled instances, we do not have access to these without a re-labeling effort. As such, we remove 10% of object annotations during training, but use them to calculate detection rates for this experiment.

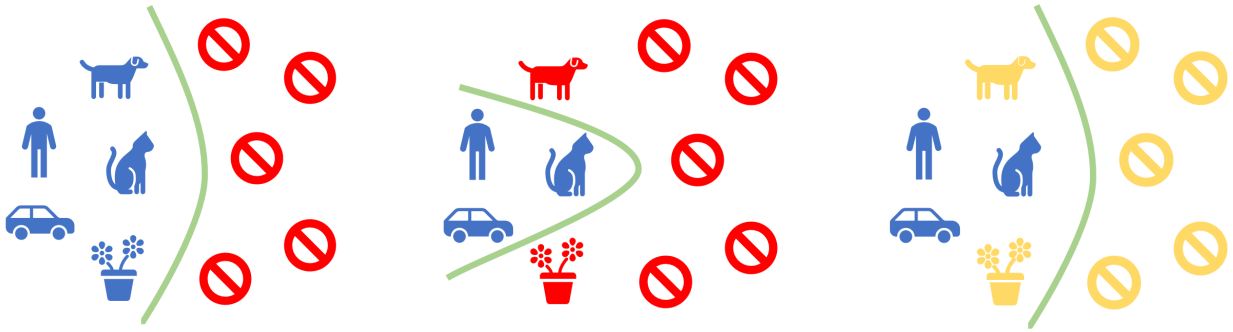
Detection rates for labeled and unlabeled objects over time are shown in Figure 3. As is expected, the model learns to detect a higher percentage of labeled instances over time, and objects are overall more likely to be detected immediately after the detector trains on them. Despite not having an explicit learning signal, unlabeled objects are still learned throughout training, but at a lower rate than labeled ones. In contrast with labeled objects, unlabeled object detections are discouraged with each PN gradient, leading to a dip in overall detection rates immediately after training. Despite this, overall detection rates of unlabeled objects grows through the first 5 epochs of training, implying a repeated cycle of learning unlabeled objects from other intra-class examples, forgetting them when explicitly trained against them, and then learning them again. Given the undesirability of this forced suppression of detected objects, we seek a method to remedy this behavior.

3. Methods

3.1. Faster R-CNN

In principle, the observed problem is characteristic of the data and is thus general to any object detection framework. However, in this work, we primarily focus on Faster R-CNN [37], a popular 2-stage method for which we provide a quick overview here.

As with other object detection models, given an input image X, the desired output of Faster R-CNN is a bounding box $B^{(i)} \in \mathbb{R}^4$ and class probabilities $c^{(i)} \in \mathbb{R}^k$ for each object (indexed by i) present, where k is the number



(a) Positive-Negative, assuming full labels (b) Positive-Negative, with missing labels (c) Positive-Unlabeled

Figure 4: A classifier (green) learns to separate proposals by "objectness". Models trained with a standard cross-entropy loss implicitly assume positive-negative (PN) learning: regions with bounding boxes are considered positive (blue), and any other proposed boxes are treated as negative (red). This is reasonable when labels are complete (a), but in reality, object detection datasets are inherently challenging to label, leading to missing annotations; this forces the classifier to exclude unlabeled objects from the positive class (b). We propose a positive-unlabeled (PU) approach (c), which considers non-labeled regions as unlabeled (yellow) rather than negative, allowing non-positive regions to be classified as positive. Best viewed in color.

of classes and the final classification decision is commonly $\operatorname{argmax} c^{(i)}$. Faster R-CNN does this in a 2-stage process. First, a convolutional neural network (CNN) [25] is used to produce image features h. A Region Proposal Network (RPN) then generates bounding box proposals $\hat{B}^{(i)}$ relative to a set of reference boxes spatially tiled over h. At the same time, the RPN predicts an "objectness" probability $\hat{c}^{(i)}$ for each proposal, learned as an object-or-not binary classifier. The second stage then takes the proposals with the highest scores, and predicts bounding box refinements to produce $B^{(i)}$ and the final classification probabilities $c^{(i)}$.

Of particular interest is how the classifier producing $\hat{c}^{(i)}$ is trained. Specifically, the cross-entropy loss $H(t, y)$ is employed, where $H(t, y)$ signifies the loss incurred when the model outputs t when the ground truth is y. In the RPN, this results in the following classification risk minimization:

$$R_{pn}^{RPN} = \pi_p \mathbb{E}[H(\hat{c}_p, +1)] + \pi_n \mathbb{E}[H(\hat{c}_n, -1)] \quad (1)$$

where π_p and π_n are the class probability priors for the positive and negative classes, respectively, and $\hat{c}_p^{(i)}$ and $\hat{c}_n^{(i)}$ are the predicted "objectness" probabilities for ground truth positive and negative regions. This risk is estimated with samples as:

$$\mathcal{L}_{pn}^{RPN} = \frac{\hat{\pi}_p}{N_p} \sum_{i=1}^{N_p} H(\hat{c}_p^{(i)}, +1) + \frac{\hat{\pi}_n}{N_n} \sum_{i=1}^{N_n} H(\hat{c}_n^{(i)}, -1) \quad (2)$$

where N_p and N_n are the number of ground truth positive and negative regions being considered, respectively, and the class priors are typically estimated as $\hat{\pi}_p = \frac{N_p}{N_p+N_n}$ and $\hat{\pi}_m = \frac{N_n}{N_p+N_n}$. Notably, this training loss treats all non-positive regions in an image as negative.

3.2. PU Learning

In a typical binary classification problem, input data $X \in \mathbb{R}^d$ are labeled as $Y \in \{\pm 1\}$, resulting in what is commonly termed a positive-negative (PN) problem. This implicitly assumes having samples from both the positive (P) and negative (N) distributions, and that these samples are labeled correctly (Figure 4a). However, in some instances, we only know the labels of the positive samples. The remainder of our data are *unlabeled* (U): samples that could be positive or negative. Such a situation is commonly called a positive-unlabeled (PU) setting, where the N distribution is replaced by an unlabeled (U) distribution (Figure 4c). Such a representation admits a classifier that can appropriately include unlabeled positive regions on the correct side of the decision boundary. We briefly review PN and PU risk estimation here.

Let $p(x, y)$ be the underlying joint distribution of (X, Y), $p_p(x) = p(x|Y = +1)$ and $p_n(x) = p(x|Y = -1)$ be the distributions of P and N data, $p(x)$ be the distribution of U data, $\pi_p = p(Y = +1)$ be the positive class-prior probability, and $\pi_n = p(Y = -1) = 1 - \pi_p$ be the negative class-prior probability. In a PN setting, data are sampled from $p_p(x)$ and $p_n(x)$ such that $\mathcal{X}_p = \{x_i^p\}_{i=1}^{N_p} \sim p_p(x)$ and $\mathcal{X}_n = \{x_i^n\}_{i=1}^{N_n} \sim p_n(x)$. Let g be an arbitrary decision function that represents a model. The risk of g can be estimated from $\mathcal{X}_p$ and $\mathcal{X}_n$ as:

$$\hat{R}_{pn}(g) = \pi_p \hat{R}_p^+(g) + \pi_n \hat{R}_n^-(g) \quad (3)$$

$\hat{R}_p^+(g) = 1/N_p \sum_{i=1}^{N_p} \ell(g(x_i^p), +1)$ and $\hat{R}_n^-(g) = 1/N_n \sum_{i=1}^{N_n} \ell(g(x_i^n), -1)$, where ℓ is the loss function. In classification, ℓ is commonly the cross-entropy loss $H(t, y)$.

In PU learning, $\mathcal{X}_n$ is unavailable; instead we have unlabeled data $\mathcal{X}_u = \{x_i^u\}_{i=1}^{N_u} \sim p(x)$, where N_u is the number of unlabeled samples. However, the negative class empirical risk $\hat{R}_n^-(g)$ in Equation 3 can be approximated indirectly [9, 10]. Denoting $R_p^-(g) = \mathbb{E}_p[\ell(g(X), -1)]$ and $R_u^-(g) = \mathbb{E}_{X\sim p(x)}[\ell(g(X), -1)]$, and observing $\pi_n p_n(x) = p(x) - \pi_p p_p(x)$, we can replace the missing term $\pi_n R_n^-(g) = R_u^-(g) - \pi_p R_p^-(g)$. Hence, we express the overall risk without explicit negative data as

$$\hat{R}_{pu}(g) = \pi_p \hat{R}_p^+(g) + \hat{R}_u^-(g) - \pi_p \hat{R}_p^-(g) \tag{4}$$

where $\hat{R}_p^-(g) = 1/N_p \sum_{i=1}^{N_p} \ell(g(x_i^p), -1)$ and $\hat{R}_u^-(g) = 1/N_u \sum_{i=1}^{N_u} \ell(g(x_i^u), -1)$.

However, a flexible enough model can overfit the data, leading to the empirical risk in Equation 4 becoming negative. Given that most modern object detectors utilize neural networks, this type of overfitting can pose a significant problem. In [21], the authors propose a non-negative PU risk estimator to combat this:

$$\hat{R}_{pu}(g) = \pi_p \hat{R}_p^+(g) + \max\{0, \hat{R}_u^-(g) - \pi_p \hat{R}_p^-(g)\} \tag{5}$$

We choose to employ this non-negative PU risk estimator for the rest of this work.

3.3. PU Learning for Object Detection

3.3.1 PU Object Proposals

In object detection datasets, the ground truth labels represent positive samples. Any regions that do not share sufficient overlap with a ground truth bounding box are typically considered as negative background, but the accuracy of this assumption depends on every object within a training image being labeled, which may not be the case (Figure 1). As shown in Figure 4b, this results in the possibility of positive regions being proposed that are labeled negative during training, due to a missing ground truth label. Therefore, we posit that object detection more naturally resembles a PU learning problem than PN.

We recognize that two-stage detection naturally contains a binary classification problem in the first stage. In Faster R-CNN specifically, the RPN comprising the first stage assigns an "objectness" score, which is learned with a binary cross-entropy loss (Equation 2). As previously noted, the PN nature of this loss can be problematic, so we propose replacing it with a PU formulation. Combining Equations 2 and 5, we produce the following loss function:

$$\mathcal{L}_{pu}^{RPN} = \frac{\pi_p}{N_p} \sum_{i=1}^{N_p} H(\hat{c}_p^{(i)}, +1) + \max\left\{0, \frac{1}{N_u} \sum_{i=1}^{N_u} H(\hat{c}_u^{(i)}, -1) - \frac{\pi_p}{N_p} \sum_{i=1}^{N_p} H(\hat{c}_p^{(i)}, -1)\right\} \tag{6}$$

Such a loss function relaxes the penalty of positive predictions for unlabeled objects.

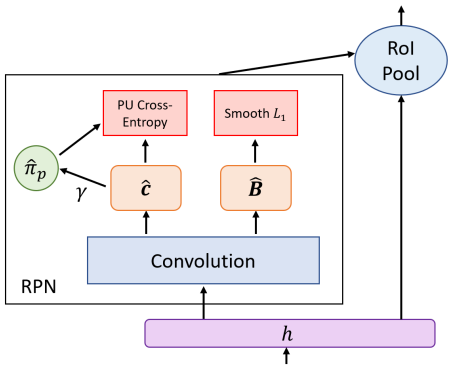


Figure 5: Faster R-CNN [37] Region Proposal Network (RPN) with the proposed positive-unlabeled cross-entropy loss. The estimate of the positive class prior $\hat{\pi}_p$ is updated with the objectness predictions $\hat{c}$, with momentum γ.

3.3.2 Estimating π_p

The PU cross-entropy loss in Equation 6 assumes the class-prior probability of the positive class π_p is known. In practice, this is not usually the case, so π_p must be estimated, denoted as $\hat{\pi}_p$. For object detection, estimating π_p is especially problematic because π_p is not static: as the RPN is trained, an increasing proportion of region proposals will (hopefully) be positive. While [21] showed some robustness to π_p misspecification, this was only on a fairly narrow range of $\pi_p \in [0.8\pi_p, 1.2\pi_p]$. During object detection performance, π_p starts from virtually 0 and grows steadily as the RPN improves. As such, any single estimate $\hat{\pi}_p$ poses the risk of being significantly off the mark during a large portion of training.

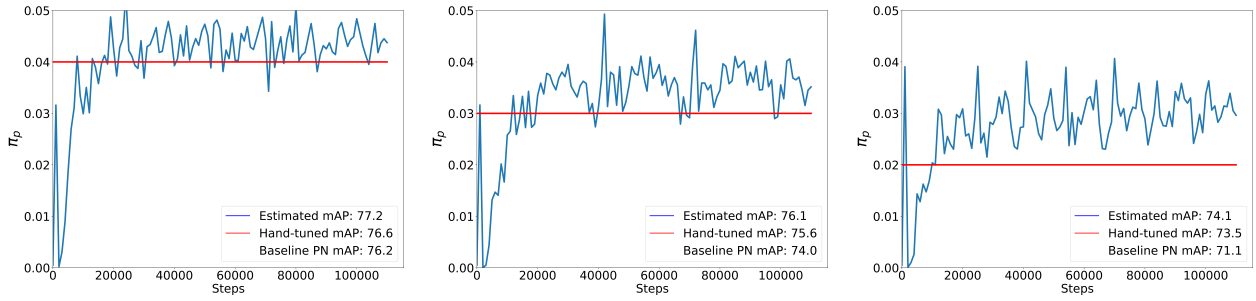
To address this, we recognize that the RPN of Faster R-CNN is already designed to infer the positive regions of an image, so we count the number of positive regions produced by the RPN and use it as an estimator for π_p:

$$\hat{\pi}_p = \frac{N_p^{RPN}}{N^{RPN}} \tag{7}$$

where N^{RPN} is the total number of RPN proposals that are sampled for training, and N_p^{RPN} being those with classifier confidence of at least 0.5. Note that this estimation of π_p comes essentially for free. Given that Faster R-CNN is trained one image at a time and the prevalence of objects varies between images, we maintain an exponential moving average with momentum γ in order to stabilize $\hat{\pi}_p$ (see Figure 5). This estimate $\hat{\pi}_p$ is then used in the calculation of the loss $\mathcal{L}_{pu}^{RPN}$ and its gradients.

4. Related Work

Like many machine learning problems, the formulation of most object detection frameworks are designed fully supervised [14, 13, 30, 37, 35, 4, 36]: it is assumed that there



(a) $\rho = 0.4$ (b) $\rho = 0.5$ (c) $\rho = 0.6$

Figure 6: Positive class prior $\hat{\pi}_p$ estimated during training of Faster R-CNN on PASCAL VOC versus from hand-tuning π_p as a hyperparameter, for instance label missingness proportion $\rho = \{0.4, 0.5, 0.6\}$.

exists a dataset of images where every object is labeled and such a dataset is available to train the model. However, as discussed above, collecting such a dataset can be expensive. Because of this, methods that can learn from partially labeled datasets have been a topic of interest for object detection. What "partially labeled" constitutes can vary, and many types of label missingness have been considered.

Weakly supervised object detection models [3, 33, 38] assume a dataset with image level labels, but not any instance labels. These models are somewhat surprisingly competent at identifying approximate locations of objects in an image without any object specific cues, but have a harder time with providing precise localization. This is especially the case when there are many of the same class of object in close proximity to each other, as individual activations can blur together, and the lack of bounding boxes makes it difficult to learn precise boundaries.

Other approaches consider settings where bounding boxes are available for some classes (e.g., PASCAL VOC's 20 classes) but not others (e.g., ImageNet [6] classes). LSDA [18] does this by modifying the final layer of a CNN [23] to recognize classes from both categories, and [41] improves upon LSDA by taking advantage of visual and semantic similarities between classes. OMNIA [34] proposes a method merging datasets that are each fully annotated for their own set of classes, but not each other's.

There are also approaches that consider a single dataset, but the labels are undercomplete across all classes. This setting most resembles what we consider in our paper. In [8], only 3-4 annotated examples per class are assumed given to start; additional pseudo-labels are generated from the model on progressively more difficult examples as the model improves. Soft-sampling has also been proposed to re-weight gradients of background regions that either have overlap with positive regions or produce high detection scores in a separately trained detector [45]; experiments were done on PASCAL VOC with a percentage of annotations discarded and on a subset of OpenImages [24].

5. Experiments

5.1. Hand-tuning Versus Estimation of π_p

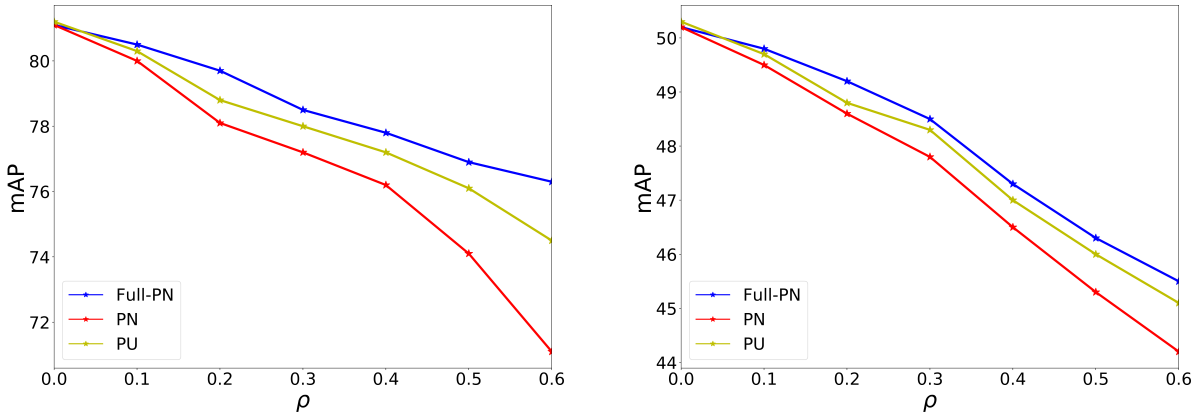
As discussed in Section 3.3.2, PU risk estimation requires the prior π_p. We experiment with two ways of determining π_p. In the first method (*Hand-Tuned*), we treat π_p as a constant hyperparameter and tune it by hand. In the second (*Estimated*), we infer π_p from our network as described in Equation 7, setting momentum γ to 0.9. We compare the estimate $\hat{\pi}_p$ inferred automatically with the hand-tuned π_p that yielded the highest mAP on PASCAL VOC. To see how our estimate changes in response to label missingness, when assembling our training set, we remove each annotation from an image with probability ρ, giving us a dataset with $1 - \rho$ proportion of the total labels, and then do our comparison for $\rho = \{0.4, 0.5, 0.6\}$ in Figure 6.

In all tested settings of ρ, the estimation $\hat{\pi}_p$ increases over time before stabilizing. Such a result matches expectations, as when an object detection model is first initialized, its parameters have yet to learn good values, and thus the true proportion of positive regions π_p is likely to be quite low. As the model trains, its ability to generate accurate regions improves, resulting in a higher proportion of regions being positive. This in turn results in a higher true value of π_p, which our estimate $\hat{\pi}_p$ follows. As the model converges, π_p (and $\hat{\pi}_p$) stabilizes towards the true prevalence of objects in the dataset relative to background regions. Interestingly, the final value of $\hat{\pi}_p$ settles close to the value of π_p found by treating the positive class prior as a static hyperparameter, but consistently above it. We hypothesize that this is due to a single static value having to hedge against the early stages of training, when π_p is lower.

We use our proposed method of auto-inferring π_p for the rest of our experiments, with $\gamma = 0.9$, rather than hand-tuning it as a hyperparameter.

5.2. PU versus PN on PASCAL VOC and MS COCO

We investigate the effect that incomplete labels have on object detection training for the popular datasets PASCAL VOC [12] and MS COCO [29], using Faster R-CNN [37]



(a) PASCAL VOC

(b) MS COCO

Figure 7: mAP at IoU 0.5 (AP_{50}) on (a) PASCAL VOC and (b) MS COCO, for a range of label missingness ρ.

with a ResNet101 [17] convolutional feature extractor. In order to quantify the effect of missing labels, we artificially discard a proportion ρ of the annotations. We compare three settings, each for a range of values of ρ. Given that the annotations are the source of the learning signal, we keep the number of total instances constant between settings for each ρ as follows:

- *PN*: We remove a proportion of labels from every image in the dataset, such that the total proportion of removed labels is equal to ρ, and all images are included in the training set. We then train the detection model with a PN objective, as is normal.
- *Full-PN*: We discard a proportion ρ of entire images and their labels, resulting in a dataset of fewer images, but each of which retains its complete annotations.
- *PU*: We use the same images and labels as in *PN*, but instead train with our proposed PU objective.

A comparison of mean average precision (mAP) performance at IoU 0.5 for these 3 settings on PASCAL VOC and MS COCO is shown in Figure 7. As expected, as ρ is increased, the detector's performance degrades. Focusing on the results for *PN* and *Full-PN*, it is clear that for an equal number of annotated objects, having fewer images that are more thoroughly annotated is preferable to a larger number of images with less thorough labels. On the other hand, considering object detection as a PU (*PU*) problem as we have proposed allows us to improve detector quality across a wide range of label missingness. While having a more carefully annotated set (*Full-PN*) is still superior, the PU objective helps close the gap. Interestingly, there is a small gain (PASCAL VOC: +0.2, MS COCO: +0.3) in mAP at full labels ($\rho = 0$), possibly due to better learning of objects missing labels in the full dataset.

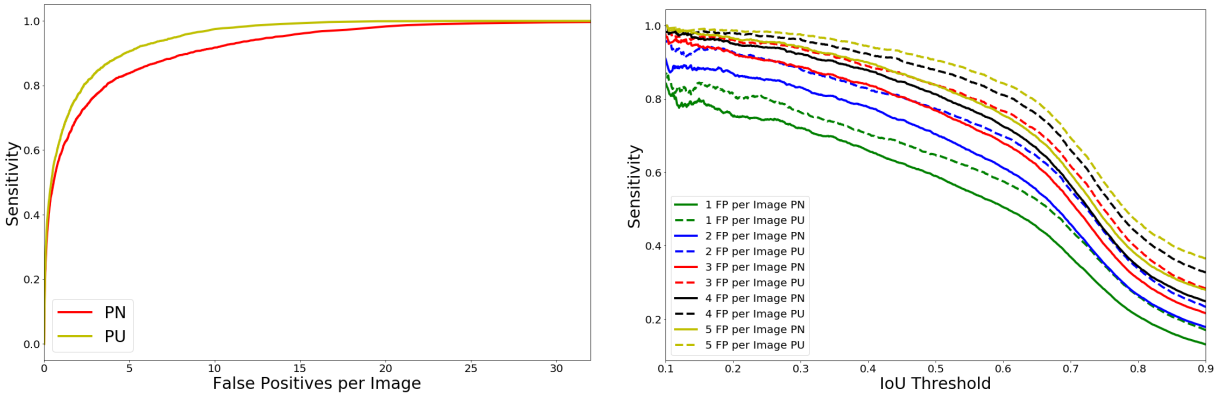
	AP_{25}		AP_{50}		AP_{75}	
Weighted?	Y	N	Y	N	Y	N
PN	12.09	22.79	9.11	17.35	2.46	9.98
PU	**13.83**	**25.56**	**10.44**	**19.89**	**4.52**	**11.79**

Table 1: Detector performance on Visual Genome, with full labels, at various IoU thresholds.

5.3. Visual Genome

Visual Genome [22] is a scene understanding dataset of objects, attributes, and relationships. While not as commonly used as an object detection benchmark as PASCAL VOC or MS COCO, Visual Genome is popular when relationships or attributes of objects are desired, as when Faster R-CNN is used as a pre-trained feature extractor for Visual Question Answering [1, 2]. Given the large number of classes (33,877) and the focus on scene understanding during the annotation process, the label coverage of all object instances present in each image is correspondingly lower than PASCAL VOC or MS COCO. In order to achieve its scale, the labeling effort was crowd-sourced to a large number of human annotators. As pointed out in [12], even increasing from 10 classes of objects in PASCAL VOC2006 to the 20 in VOC2007 resulted in a substantially larger number of labeling errors, as it became more difficult for human annotators to remember all of the object classes. This problem is worse by several orders of magnitude for Visual Genome. While the dataset creators implemented certain measures to ensure quality, there still are many examples of missing labels. In such a setting, the proposed PU risk estimation is especially appropriate, even with all included labels.

We train ResNet101 Faster R-CNN using both PN and the proposed PU risk estimation on 1600 of the top object



(a) (b)

Figure 8: Lesion sensitivity versus (a) false positive rate and (b) IoU threshold for different false positive (FP) allowances per image. We compare the baseline Faster R-CNN variant in [47] trained with a PN objective versus the proposed PU objective.

classes of Visual Genome, as in [2]. We evaluate performance on the classes present in the test set and report mAP at various IoU thresholds $\{0.25, 0.50, 0.75\}$ in Table 1. We also show mAP results when each class's average precision is weighted according to class frequency, as done in [2]. The PASCAL VOC and MS COCO results in Figure 7 indicate that we might expect increasing benefit from utilizing a PU loss as missing labels become especially prevalent, and for Visual Genome, where this is indeed the case, we observe that PU risk estimation outperforms PN by a significant margin, across all settings.

5.4. DeepLesion

The recent progress in computer vision has attracted increasing attention towards potential health applications. To encourage deep learning research in this direction, the National Institutes of Health (NIH) Clinical Center released DeepLesion [47], a dataset consisting of 32K CT scans with annotated lesions. Unlike PASCAL VOC, MS COCO, or Visual Genome, labeling cannot be crowd-sourced for most medical datasets, as accurate labeling requires medical expertise. Even with medical experts, labeling can be inconsistent; lesion detection is a challenging task, with biopsy often necessary to get an accurate result. Like other datasets labeled by an ensemble of annotators, the ground truth of medical datasets may contain inconsistencies, with some doctors being more conservative or aggressive in their diagnoses. Due to these considerations, a PU approach more accurately characterizes the nature of the data.

We re-implemented the modified version of Faster R-CNN described in [47] as the baseline model and compare against our proposed model using the PU objective, making no other changes. We split the dataset into 70%-15%-15% parts for training, validation, and test. Following [47], we report results in terms of free receiver operating characteristic (FROC) and sensitivity of lesion detection versus intersection-over-union (IoU) threshold for a range of allowed false positives (FP) per image (Figure 8). In both cases, we show that switching from a PN objective to a PU one results in gains in performance.

6. Conclusion and Future Work

Having observed that object detection data more closely resembles a positive-unlabeled (PU) problem, we propose training object detection models with a PU objective. Such an objective requires estimation of the class probability of the positive class, but we demonstrate how this can be estimated dynamically with little modification to the existing architecture. Making these changes allows us to achieve improved detection performance across a diverse set of datasets, some of which are real datasets with significant labeling difficulties. While we primarily focused our attention on object detection, a number of other popular tasks share similar characteristics and could also benefit from being recast as PU learning problems (e.g., segmentation [39, 31, 16], action detection [40, 19, 15]).

In our current implementation, we primarily focus on applying the PU objective to the binary object-or-not classifier in Faster R-CNN's Region Proposal Network. A natural extension of this work would be to apply the same objective to the second stage classifier, which must also separate objects from background. However, as the second stage classifier outputs one of several classes (or background), the classification is no longer binary, and requires estimating multiple class priors $\{\pi_c\}_{c=1}^{k}$ [46], which we leave to future work. Such a multi-class PU loss would also allow extension to single-stage detectors like SSD [30] and YOLO [35, 36]. Given the performance gains already observed, we believe this to be an effective and natural improvement to the object detection classification loss.

References

[1] Aishwarya Agrawal, Jiasen Lu, Stanislaw Antol, Margaret Mitchell, C. Lawrence Zitnick, Dhruv Batra, and Devi Parikh. VQA: Visual Question Answering. *International Conference on Computer Vision*, 2015. 7

[2] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering. *Conference on Computer Vision and Pattern Recognition*, 2018. 7, 8

[3] Hakan Bilen and Andrea Vedaldi. Weakly Supervised Deep Detection Networks. *Conference on Computer Vision and Pattern Recognition*, 2016. 6

[4] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-FCN: Object Detection via Region-based Fully Convolutional Networks. *Advances In Neural Information Processing Systems*, 2016. 1, 5

[5] Francesco De Comité, François Denis, Rémi Gilleron, and Fabien Letouzey. Positive and Unlabeled Examples Help Learning. *Algorithmic Learning Theory*, 1999. 2

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. ImageNet: A large-scale hierarchical image database. *Conference on Computer Vision and Pattern Recognition*, 2009. 6

[7] François Denis. PAC Learning from Positive Statistical Queries. *Algorithmic Learning Theory*, 1998. 2

[8] Xuanyi Dong, Liang Zheng, Fan Ma, Yi Yang, and Deyu Meng. Few-shot Object Detection. *Transactions on Pattern Analysis and Machine Intelligence*, 2018. 6

[9] Marthinus Du Plessis, Gang Niu, and Masashi Sugiyama. Convex formulation for learning from positive and unlabeled data. In *International Conference on Machine Learning*, pages 1386–1394, 2015. 5

[10] Marthinus C Du Plessis, Gang Niu, and Masashi Sugiyama. Analysis of learning from positive and unlabeled data. In *Advances in neural information processing systems*, pages 703–711, 2014. 5

[11] Charles Elkan and Keith Noto. Learning Classifiers from Only Positive and Unlabeled Data. *International Conference on Knowledge Discovery and Data Mining*, 2008. 2

[12] Mark Everingham, Luc Van Gool, Christopher K.I. Williams, John Winn, and Andrew Zisserman. The Pascal Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision*, 2010. 1, 2, 3, 6, 7

[13] Ross Girshick. Fast R-CNN. *International Conference on Computer Vision*, 2015. 1, 5

[14] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *Conference on Computer Vision and Pattern Recognition*, 2014. 1, 5

[15] Chunhui Gu, Chen Sun, David A. Ross, Carl Vondrick, Caroline Pantofaru, Yeqing Li, Sudheendra Vijayanarasimhan, George Toderici, Susanna Ricco, Rahul Sukthankar, Cordelia Schmid, and Jitendra Malik. AVA: A Video Dataset of Spatio-Temporally Localized Atomic Visual Actions. *Conference on Computer Vision and Pattern Recognition*, 2018. 8

[16] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. *International Conference on Computer Vision*, 2017. 8

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. *Conference on Computer Vision and Pattern Recognition*, 2016. 7

[18] Judy Hoffman, Sergio Guadarrama, Eric Tzeng, Jeff Donahue, Ross B. Girshick, Trevor Darrell, and Kate Saenko. LSDA: Large Scale Detection Through Adaptation. *Advances In Neural Information Processing Systems*, 2014. 6

[19] Haroon Idrees, Amir R. Zamir, Yu-Gang Jiang, Alex Gorban, Ivan Laptev, Rahul Sukthankar, and Mubarak Shah. The THUMOS challenge on action recognition for videos "in the wild". *Computer Vision and Image Understanding*, 2017. 8

[20] Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. MentorNet: Learning Data-Driven Curriculum for Very Deep Neural Networks on Corrupted Labels. *International Conference on Machine Learning*, 2018. 1, 2

[21] Ryuichi Kiryo, Gang Niu, Marthinus C Du Plessis, and Masashi Sugiyama. Positive-Unlabeled Learning with Non-Negative Risk Estimator. *Advances In Neural Information Processing Systems*, 2017. 2, 5

[22] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei. Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations. *International Journal Computer Vision*, 2017. 1, 2, 7

[23] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet Classification with Deep Convolutional Neural Networks. *Advances In Neural Information Processing Systems*, 2012. 6

[24] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Tom Duerig, and Vittorio Ferrari. The Open Images Dataset V4: Unified image classification, object detection, and visual relationship detection at scale. *arXiv preprint*, 2018. 2, 6

[25] Y. Lecun, B. Boser, J.S Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Computation*, 1989. 4

[26] Rebecca Sawyer Lee, Francisco Gimenez, Assaf Hoogi, Kanae Kawai Miyake, Mia Gorovoy, and Daniel L. Rubin. A curated mammography data set for use in computer-aided detection and diagnosis research. *Nature Scientific Data*, 2017. 1

[27] Fabien Letouzey, François Denis, and Rémi Gilleron. Learning from Positive and Unlabeled Examples. *Algorithmic Learning Theory*, 2000. 2

[28] Kevin J Liang, Geert Heilmann, Christopher Gregory, Souleymane Diallo, David Carlson, Gregory Spell, John Sigman, Kris Roe, and Lawrence Carin. Automatic Threat Recognition of Prohibited Items at Aviation Checkpoints with X-Ray Imaging: a Deep Learning Approach. *Proc SPIE, Anomaly Detection and Imaging with X-Rays (ADIX) III*, 2018. 1

[29] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C Lawrence Zitnick, and Piotr Dolí. Microsoft COCO: Common Objects in Context. *European Conference on Computer Vision*, 2014. 1, 2, 6

[30] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. SSD: Single Shot MultiBox Detector. *European Conference on Computer Vision*, 2016. 1, 5, 8

[31] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully Convolutional Networks for Semantic Segmentation. *Conference on Computer Vision and Pattern Recognition*, 2015. 8

[32] Inês C. Moreira, Igor Amaral, Inês Domingues, Ant´ónio Cardoso, Maria Jo˜ão Cardoso, and Jaime S. Cardoso. INbreast: Toward a Full-field Digital Mammographic Database. *Academic Radiology*, 2012. 1

[33] Maxime Oquab, Léon Bottou, Ivan Laptev, and Josef Sivic. Is object localization for free? Weakly-supervised learning with convolutional neural networks. *Conference on Computer Vision and Pattern Recognition*, 2015. 6

[34] Alexandre Rame, Emilien Garreau, Hedi Ben-Younes, and Charles Ollion. OMNIA Faster R-CNN: Detection in the wild through dataset merging and soft distillation. *arXiv preprint*, 2018. 6

[35] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You Only Look Once: Unified, Real-Time Object Detection. *Conference on Computer Vision and Pattern Recognition*, 2016. 1, 5, 8

[36] Joseph Redmon and Ali Farhadi. YOLO9000: Better, Faster, Stronger. *Conference on Computer Vision and Pattern Recognition*, 2017. 1, 5, 8

[37] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *Advances in Neural Information Processing Systems*, 2015. 1, 3, 5, 6

[38] Mrigank Rochan and Yang Wang. Weakly supervised localization of novel objects using appearance transfer. *Conference on Computer Vision and Pattern Recognition*, 2015. 6

[39] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. *Medical Image Computing and Computer Assisted Intervention*, 2015. 8

[40] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. T. *Technical Report CRCV-TR-12-01, University of Central Florida*, 2012. 8

[41] Yuxing Tang, Josiah Wang, Boyang Gao, Emmanuel Dellandrea, Robert Gaizauskas, and Liming Chen. Large Scale Semi-Supervised Object Detection Using Visual and Semantic Knowledge Transfer. *Conference on Computer Vision and Pattern Recognition*, 2016. 6

[42] Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J. Gordon. An Empirical Study of Example Forgetting during Deep Neural Network Learning. *International Conference on Learning Representations*, 2019. 1, 2, 3

[43] Andreas Veit, Neil Alldrin, Gal Chechik, Ivan Krasin, Abhinav Gupta, and Serge Belongie. Learning From Noisy Large-Scale Datasets With Minimal Supervision. *Conference on Computer Vision and Pattern Recognition*, 2017. 1, 2

[44] Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald M. Summers. ChestX-ray8: Hospital-Scale Chest X-Ray Database and Benchmarks on Weakly-Supervised Classification and Localization of Common Thorax Diseases. *Conference on Computer Vision and Pattern Recognition*, 2017. 1

[45] Zhe Wu, Navaneeth Bodla, Bharat Singh, Mahyar Najibi, Rama Chellappa, and Larry S. Davis. Soft Sampling for Robust Object Detection. *British Machine Vision Conference*, 2019. 6

[46] Yixing Xu, Chang Xu, Chao Xu, and Dacheng Tao. Multi-positive and unlabeled learning. In *IJCAI*, pages 3182–3188, 2017. 8

[47] Ke Yan, Xiaosong Wang, Le Lu, and Ronald M. Summers. DeepLesion: Automated Deep Mining, Categorization and Detection of Significant Radiology Image Findings using Large-Scale Clinical Lesion Annotations. *arXiv preprint*, 2017. 1, 2, 8

[48] Chiyuan Zhang, Samy Bengio, Google Brain, Moritz Hardt, Benjamin Recht, Oriol Vinyals, and Google Deepmind. Understanding Deep Learning Requires Re-thinking Generalization. *International Conference on Learning Representations*, 2017. 1, 2