Image-caption difficulty for efficient weakly-supervised object detection from in-the-wild data

Giacomo Nebbia
University of Pittsburgh
gin2@pitt.edu

Adriana Kovashka
University of Pittsburgh
kovashka@cs.pitt.edu

Abstract

In recent years, we have witnessed the collection of larger and larger multi-modal, image-caption datasets: from hundreds of thousands such pairs to hundreds of millions. Such datasets allow researchers to build powerful deep learning models, at the cost of requiring intensive computational resources. In this work, we ask: can we use such datasets efficiently without sacrificing performance? We tackle this problem by extracting difficulty scores from each image-caption sample, and by using such scores to make training more effective and efficient. We compare two ways to use difficulty scores to influence training: filtering a representative subset of each dataset and ordering samples through curriculum learning. We analyze and compare difficulty scores extracted from a single modality*—captions (i.e., caption length and number of object mentions) or images (i.e., region proposals' size and number)—or based on* alignment *of image-caption pairs (i.e., CLIP and concreteness). We focus on Weakly-Supervised Object Detection where image-level labels are extracted from captions. We discover that (1) combining filtering and curriculum learning can achieve large gains in performance, but not all methods are stable across experimental settings, (2) single-modality scores often outperform alignment-based ones, (3) alignment scores show the largest gains when training time is limited.*

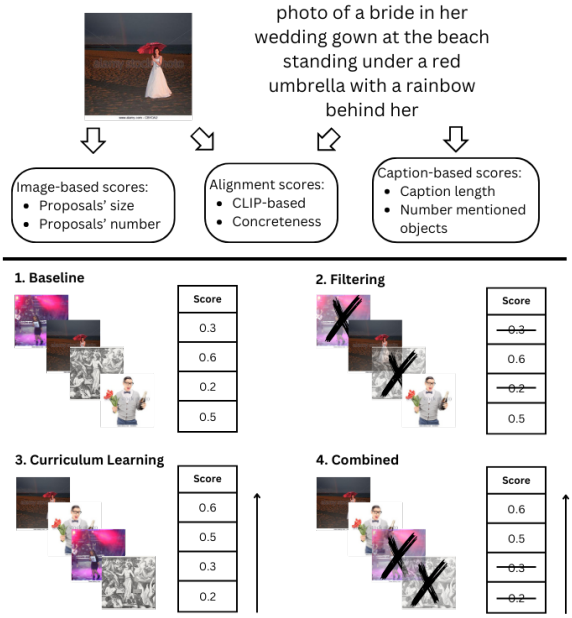

Figure 1. Top: we extract two image-caption alignment difficulty scores, two caption-based scores, and two image-based scores. Bottom: we train WSOD models by (1) randomly sampling images (baseline), (2) filtering out low-scoring samples, (3) using curriculum learning to sort images and sample high-scoring ones before low-scoring ones, and (4) combining (2) and (3) by applying curriculum learning to the top-scoring images only.

1. Introduction

The size of multi-modal, image-caption pair datasets has drastically increased in the past decade: from hundreds of thousands (e.g., COCO Captions [5]) to millions (e.g., Conceptual Captions [4, 27]) to hundreds of millions (e.g., data used to train models such as CLIP [21] or ALIGN [15]) to billions (e.g., LAION-5B [26]). While the collection of such datasets has allowed researchers to build computer vision models that achieve impressive performance on a variety of tasks [15, 21], training on such datasets becomes very resource-intensive. We aim to compare different ways to reduce such burden while preserving (or improving) models' performance.

When collecting such large numbers of images, clean, descriptive captions cannot be obtained through paid crowdsourcing (as done for smaller datasets [5]), but text that already accompanies images must be crawled instead (e.g., the alt-text HTML field [4, 27] or captions to images shared on Reddit [6]). We refer to such datasets as "in-the-wild" since the captioning process is not regulated and people captioning images are not paid to do so. Unlike crowdsourced captions, which provide a precise and thorough description

of the image, in-the-wild ones, e.g., ones on Reddit, aim to *complement* the image, thus their level of alignment or relevance to the image may vary; see Fig. 2. Overall, extracting supervision (labels) for the image from the caption can be more productive for some image-caption pairs than others.

In this work, we compare ways to reduce the computational burden of training computer vision models on large-scale, in-the-wild datasets, while focusing on productive supervision from image-text pairs. Specifically, we focus on Weakly Supervised Object Detection (WSOD) models trained on labels extracted from captions. In-the-wild datasets are not commonly used for this task due to reduced image-text alignment, but their potential is worth exploring due to their size. Our key idea is to extract difficulty scores from the image-caption pairs, and use this information to efficiently train WSOD models: such scores represent how easy/hard it would be for a model to learn from each image-caption pair. We leverage difficulty scores in three ways: (1) by filtering training samples to only include the "easiest" part of a dataset, (2) via curriculum learning (CL), where different samples are shown at different training stages, and (3) by combining filtering and CL. Fig. 1 shows a summaryof our approaches.

We consider three ways of extracting difficulty scores from image-caption pairs: single-modality, from (1) captions alone or (2) images alone, and based on alignment (3) of the image-text pair. For the first, we consider caption length and number of mentioned objects of interest; for the second, average size and number of region proposals per image; and, for the last, CLIP score [21] and concreteness [13]. We argue that caption length can be considered a proxy for number of mentioned objects of interest: intuitively, the longer a caption, the more likely it is to mention an object. Number of mentions (and caption length) represents a difficulty score since the more mentioned objects, the denser the supervisory signal (i.e., more signal to learn from per image). The advantage of using caption length over the number of mentioned objects is that there is no need to define what such objects are, providing a more generalizable difficulty score that could be applied to other tasks beyond WSOD. In addition to caption-only scores, we consider two image-only scores: average size of region proposals and their number, which we regard as proxies for ground truth object size and number, which have been shown to capture useful information in previous work [29, 44], but are unavailable in in-the-wild datasets. Finally, we consider image-caption alignment: a way to quantify how well a caption describes its corresponding image. The assumption behind using alignment as a difficulty score is that better aligned captions should provide better supervisory signal. We test two ways to compute alignment: using the CLIP model to compute the cosine similarity between image and caption, and using the concreteness score, which captures how similar images whose captions use the same words are (the more similar such images, the more concrete the words). We compare the promise of using these scores for curriculum learning and filtering in a variety of experimental settings: across datasets, hyperparameters, and training schedules.

We show that: (1) single-modality scores outperform alignment-based scores when hyperparameter selection is suboptimal, but are less stable across datasets; (2) alignment scores boost performance especially when training resources, such as training time, are limited; (3) the CLIP alignment score shows the most consistent performance improvements across datasets, hyperparameters choices, and training schedules; (4) curriculum learning is generally more effective than filtering, but their combination can boost results further.

2. Related Work

Weakly Supervised Object Detection. While fully supervised object detection [12, 22] refers to the task of finding objects in an image when bounding boxes and their labels are available as supervision, in Weakly Supervised Object Detection (WSOD) only image-level labels are available at training time. One of the first successful approaches for WSOD is Weakly Supervised Deep Detection Networks (WSDDN) [3], from which numerous models originated [24, 32, 33, 38]. These approaches use Multiple-Instance Learning (MIL) to combine region proposal-level predictions and allow supervision from image-level labels. We use WSDDN in our experiments as it is the base model for so many WSOD frameworks. Some prior work [36] tests filtering (but not curriculum learning) using only dataset-specific alignment metrics (unlike CLIP) on small datasets/subsets (e.g., 64k images, 14 times smaller than the 907k we use for RedCaps), thus leading to conclusions that do not hold for large-scale, in-the-wild datasets (e.g., filtering never improves results).

As an alternative to MIL approaches, recent methods have started providing textual input to the model in addition to the image and using contrastive learning to train the model in a self-supervised way [9, 20, 40]. Different from our approach, such methods require ground-truth bounding boxes for training for the base classes, which we do not.

CLIP-based Approaches. CLIP [21] has become a popular image-text model included in many architectures [9, 18, 23, 28, 40]. Our use of CLIP differs from these since we use the CLIP model offline (not during training as either part of the architecture or as supervision), running our image-caption data through it before training starts. This makes our approach modular, where we can replace CLIP with any other multi-modal model's score without changing our model's architecture.

While we focus on sample selection and ordering, Pro-

posalCLIP [28] uses CLIP-based similarity scores offline to refine region proposals, which is orthogonal to our approach. Unlike our work, ProposalCLIP includes a multi-step refinement process and trains a neural network from CLIP-derived pseudo ground truth labels to achieve its goal.

Finally, CLIP has been used to filter out potentially mismatched image-caption pairs during data collection in the LAION datasets [25, 26], where pairs with CLIP cosine similarity below a threshold were removed. We formally evaluate the impact such choice makes, comparing performance for models trained with all data and with filtered data. In addition, we use curriculum learning to make use of all the collected data while making model training more effective and efficient.

Curriculum Learning. The idea of curriculum learning (CL) was introduced in [2] and inspired by how humans learn: from easy concepts to more difficult ones. CL posits that models should be trained with easy samples first, and the difficulty of samples should be increased throughout training. CL thus relies on two main components: a difficulty score to sort samples from easy to hard and a pacing function to decide how to choose samples during each training step [10, 37]. The choice of difficulty score allows researchers to introduce domain knowledge into the training process since they decide which characteristics of the data make a sample easy/hard. For computer vision applications, number of ground truth objects [44], object size [29], and human performance on visual tasks [31, 34] are among the scores that have been investigated. For WSOD applications, previous CL approaches focus on image features (such as number of objects) since image-level ground truth labels are assumed known. Our approach differs since we extract labels from captions (which may be noisy) and are the first, to the best of our knowledge, to use caption-based and image-caption pair-based difficulty scores, thus exploiting the image-caption relationship that is not leveraged in previous studies.

3. Methods

In this section, we introduce the difficulty scores we consider, explain the ways we use such scores to guide model training, and summarize the network architecture we experiment with.

3.1. Caption-based Difficulty Scores

We extract *caption length* (defined as the number of words in a caption) as our main caption-based difficulty score. Such score is simple to compute and is generalizable to tasks other than WSOD since it does not require knowledge of which categories of interest models are trained on. We argue that caption length could be a proxy for the *number of objects of interest mentioned* by a caption: the longer the caption, the more likely it is to mention objects of interest, and, consequently, the more supervision it can provide.

3.2. Image-based Difficulty Scores

We use *average size* of the region proposals per image and *number* of regions proposals per image as our image-only difficulty scores. We use Selective Search [35] and MCG [1] to compute region proposals (Sec. 4.2). We choose such scores as proxies for number and size of ground truth objects, which have been successfully used with CL in previous work [29, 44], but are unavailable for large, not manually annotated, in-the-wild datasets.

3.3. Image-caption Alignment Difficulty Scores

Image-caption alignment measures aim to quantify how well a caption describes an image. We hypothesize that higher alignment should translate to a stronger supervisory signal provided by the caption for WSOD. We consider two alignment measures: a CLIP score [21] and concreteness [13]. The former is a general-purpose score derived from a model trained on a wide variety of image-caption pairs (which has been linked to this model's robustness across datasets [7]), the latter is a dataset-specific method aiming to capture nuances in alignment that could be different from one dataset to another.

3.3.1 CLIP-based Alignment

The CLIP model [21] maximizes the cosine similarity between corresponding image-caption embeddings and minimizes the similarity between all non-corresponding image-caption pairs' embeddings in the batch. Given an image-caption input pair, we compute the cosine similarity between the CLIP text and image embeddings and consider it as our alignment score. Fig. 2 shows examples of image-caption pairs for the two in-the-wild datasets we use in our experiments, stratified according to their CLIP-based alignment. While we see that captions in top-scoring pairs closely describe the image, in bottom-scoring pairs, they do not. For instance, the bottom two pairs represent misalignment errors naturally occurring in in-the-wild datasets. Captions for intermediate-scoring pairs are more descriptive than bottom-scoring ones, but more vague than top-scoring pairs.

3.3.2 Concreteness

The idea behind concreteness is to measure how well-clustered images with shared words in their captions are in an image features space; if a word is tightly clustered, it is highly concrete [13]. In detail, let w_v represent the set of words associated with image v, and let V_w be the set of images associated with a word w. For image $v \in V_w$, we first

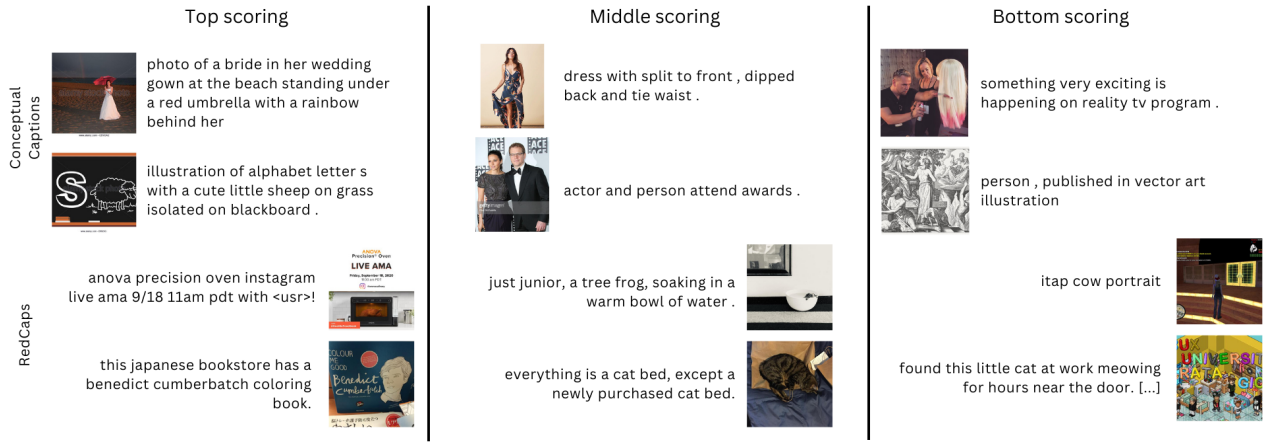

Figure 2. Examples of top- (left), middle- (center), and bottom- (right) scoring captions according to our CLIP-based alignment for the two in-the-wild datasets we consider: Conceptual Captions (top) and RedCaps (bottom).

measure how often v's nearest neighbors are also associated with w by computing the expected value of the Mutually Neighboring Images (MNI) of word w as:

$$\mathbb{E}_{P_{data}}[MNI_w^k] = \frac{1}{|V_w|} \sum_{v \in V_w} |NN^k(v) \cap V_w| \quad (1)$$

where $NN^k(v)$ represents the set of k Nearest Neighbors of v in image space. To correct for word frequency, concreteness is computed as the ratio of the value in Eq. 1 and the expected value computed under a random distribution of image data.

$$concreteness(w) = \frac{\mathbb{E}_{P_{data}}[MNI_w^k]}{\mathbb{E}_{P_{random}}[MNI_w^k]} \quad (2)$$

3.4. Guiding Model Training with Difficulty Scores

With the previously introduced metrics, we can assign a score to each image-caption pair: the higher the score, the easier learning from the pair is assumed to be. Here, we introduce three ways such scores can guide model training: (1) filtering, (2) curriculum learning, and (3) their combination.

3.4.1 Filtering

With this approach, we train models only on a subset of high-scoring, easy-to-learn-from image-caption pairs. This method is easy to implement and can significantly reduce the amount of resources needed to train a model, depending on how many samples one filters out. In the experiments, for each score, we take the top scoring (i.e., most aligned or easiest to learn from) 50% of each dataset and train a WSOD model on these subsets. With this approach, we view each sample as either informative (thus included) or noisy (thus filtered out). This binarization of the difficulty scores does not fully leverage the nuances in quantifying difficulty provided by difficulty scores, and thus may seem like an oversimplification. To remedy this issue, we experiment with curriculum learning, described next.

3.4.2 Curriculum learning

With curriculum learning (CL) [2], we train models on "easy" concepts first, and then build up to more complex ones. To do so, training samples must be scored by difficulty and a model is first trained with the easiest samples. Harder samples are progressively included in the training set as training progresses. Specifically, for each score, samples are grouped in four subsets representing the four quartiles according to a given difficulty score. WSOD models are trained on the top quartile only first and additional quartiles are sequentially added as training progresses. CL is thus able to leverage the nuanced order our proposed difficulty scores provide: all samples are used, but the model learns from them at different stages. CL can still provide computational advantages since early epochs do not include all data: by keeping the number of epochs fixed, fewer images are seen during training, thus making the training process more efficient. At the same time, since all data points are used at a certain point for training, the resulting training speedup is not as high as the one provided by the filtering approach.

3.4.3 Filtering and curriculum learning

Finally, we investigate the benefit of combining the two approaches. We apply filtering first and then train models us-

ing CL within the chosen subset. With this approach, we further improve training speed by not training a model on the whole filtered subset at each epoch. Specifically, we only consider the top-scoring 50% or 75% of each dataset, and we train the model using CL within the chosen subset. We consider 75% in addition to 50% (as used for filtering) because combining filtering and CL causes a more dramatic scarcity of data seen compared to filtering only (since CL trains on a subset of the already-filtered data in each epoch).

3.5. Weakly Supervised Deep Detection Network

3.5.1 Architecture

We use the Weakly Supervised Deep Detection Network (WSDDN) [3] model for WSOD (with some architectural changes, as done in previous work [36, 43]). WSDDN represents one of the first successful approaches for WSOD and has been used as the base network for future models [24, 32, 33, 38]. This model adapts a VGG16 [30] to make it trainable for WSOD: the last pooling layer is replaced by a region pooling layer which, given an image feature matrix and a region, returns a feature representation for that region. While WSDDN uses spatial pyramid pooling [11, 17], we use the common ROI pool [8]. Region-level features are then processed by two fully-connected layers with ReLU activation functions. After these layers, two streams branch out: a classification data stream, and a detection data stream. The two streams are then combined to obtain a score matrix $x^R \in \mathbb{R}^{C\times|R|}$, with C being the number of classes and $|R|$ the number of regions. Regions are then sorted by this score (for each class independently) and non-maxima suppression is applied to obtain the final list of class-specific detections for each image. To go from region-level scores to image-level class prediction scores y_c, we sum scores x^R for a given class across the regions, and the network is trained by computing an image-level loss as the sum of class-specific binary entropy losses. Finally, the WSDDN paper introduces a spatial regularizer, which we omit.

3.5.2 Extracting Objects from Captions

To leverage captions as supervision to train WSOD models, we extract mentions of objects of interest (i.e., COCO categories) from the captions. To do so, we apply `ExactMatch` (also previously used in [7, 42]), where we extract verbatim mentions of such objects. [42] report 89% precision and 62% recall of the labels extracted from COCO Captions, but in our setting these values are likely lower due to the variable and reduced image-text alignment in the datasets from which we extract training data, Conceptual Captions and RedCaps. Note no image-level labels are provided in these datasets.

4. Experimental Validation

4.1. Datasets

We focus our experiments on two in-the-wild datasets, Conceptual Captions and RedCaps. We also conduct a limited set of experiments on COCO Captions, to compare and contrast findings about the impact of filtering and curriculum learning on in-the-wild versus more traditional, descriptive captions. We use COCO for evaluation of WSOD models trained with supervision from captions in these three datasets.

Conceptual Captions [27] (CC) includes 3M image-caption pairs crawled from the Internet where captions are extracted from the `alt-text` HTML field. This dataset represents one of the first in-the-wild datasets since captions are not generated ad hoc after the collection of the dataset, but they are still pre-processed to ensure as high quality a dataset as possible (e.g., removal of captions with high token repetition rate or with high noun rate or capitalized-word ratio). Because we extract mentions of objects of interest from the captions to use as supervision, we exclude all image-caption pairs for which `ExactMatch` extracts no mention, yielding a dataset of 416,642 images.

RedCaps [6] includes 12M image-caption pairs collected from 350 manually selected subreddits to ensure the inclusion of photographs (rather than, for instance, memes) and to limit the number of people (which are present in the majority of images in other common manually curated datasets [5]). RedCaps differs from CC as it applies minimal caption pre-processing and instead relies on the nature of Reddit to produce high quality data: captions are generated from human users to accompany each image. The dataset maintains its in-the-wild characteristics though because users are not bound to caption images with their description. In addition, the limited amount of pre-processing makes RedCaps potentially more likely to include noisy data (e.g., misaligned pairs as shown in Fig. 2). The reduced amount of human involvement in curating this dataset (via preprocessing) and its larger size make it even more appealing than CC as a source of supervision for WSOD. As done for CC, we exclude images whose captions do not contain mentions of objects of interest as extracted by `ExactMatch`, resulting in a total of 907,339 images.

We also use COCO Captions [5] to train WSOD models and verify if results for in-the-wild datasets apply to older, manually curated datasets as well. COCO Captions includes an average of 5 captions per image in COCO Objects. Such captions were crowd-sourced from human annotators. As they were collected with specific instructions to human annotators, these captions differ from in-the-wild ones, which naturally co-occur with images. After excluding images whose captions include no mentions of objects of interest, 95,110 images remain in the dataset.

Since the previously introduced datasets do not include annotated samples, we evaluate our WSOD models on COCO Objects [19], which includes 118,287 training images and 5,000 validation images.

4.2. Implementation

We use PyTorch 1.7.1 to implement our models. We extract region proposals using Selective Search [35] for CC and RedCaps, and using MCG [1] for COCO training and evaluation. We empirically found MCG proposals to work better than SS for COCO, but they are not available for CC and RedCaps. All models are trained on 2 Titan X Nvidia GPUs with total batch size of 4.

4.2.1 Curriculum and training schedule

For curriculum learning, we use a step pacing function over the course of 4 (partial) epochs. Training starts with a *first partial epoch* containing images in the top quartile only, based on one of the difficulty metrics. Then the second top-scoring one is added, thus in the *second partial epoch* the model is trained with half of the data (the "easier" half based on difficulty metrics). In the *third partial epoch*, the third quartile is added, and in the *fourth epoch*, the model is trained with all the data. The only exception to this scheme is that used for number of mentioned objects: due to the limited number of samples with more than one mentioned object, we train the model for 3 epochs with images whose captions mention more than one object, and with all images for a final epoch. For filtering, models are trained for 4 epochs as well (each epoch containing the same 50% of the dataset). Filtering experiments using number of mentions are an exception, as they are trained twice as long (i.e., 8 epochs) given the low number of captions with more than one object mention. The filtering+CL method is also trained for 4 epochs. Baselines are trained for 3 epochs to ensure that baseline models do not train on fewer images than those used for CL: given that we add a quarter of the data at each epoch, CL models are trained on 2.5 times (0.25+0.5+0.75+1) the original size of each dataset. Baseline models see all data in each epoch. Overall, baselines are trained with 3x the number of images in the original data, CL methods with 2.5x that size, filtering methods with 2x, and combination methods with 1.25x or 1.875x (depending on whether we keep the easiest 50% or 75% of the data in each of 4 epochs). All models trained on COCO are trained for 7 epochs due to the smaller training set size.

4.3. Main analysis on in-the-wild datasets

Curriculum learning uses all data but uses it more efficiently by only training on a subset of the data in early epochs. Thus, we test curriculum learning for each of the six introduced difficulty scores (i.e., CLIP-based alignment, concreteness, caption length, number of mentioned objects, proposal number, proposal size). For a subset of these difficulty scores, we also test filtering and the combination of filtering and CL. Specifically, we combine the best-performing filtering method and two well-performing CL methods. As our baseline, we train WSDDN models on RedCaps and CC without filtering or CL.

Table 1 reports results for our filtering and CL strategies on CC and RedCaps, using a learning rate of 1e-3. We notice how CL outperforms filtering approaches (italic results, indicating methods that outperform the baseline). This is true for eight of the twelve methods tested for CL, and only one of the four tested for filtering. Looking at CLIP and concreteness specifically, results are better (or comparable) with CL than filtering.

	CC	RedCaps
Baseline	2.2	1.8
Filtering (keep 50% of data)		
CLIP	**1.5**	***2.6***
Concreteness	0.7	1.4
Curriculum learning		
Caption length	n/a	*2.5*
Num. mentioned objects	1.5	***4.4***
Proposal size	***2.9***	*3.0*
Proposal number	1.7	*3.1*
CLIP	*2.4*	*2.5*
Concreteness	*2.5*	1.6
Filtering + curriculum learning		
Keep 50% of data		
CLIP + Caption length	0.9	*3.6*
CLIP + Proposal size	0.9	*1.9*
Keep 75% of data		
CLIP + Caption length	***5.8***	***5.2***
CLIP + Proposal size	*2.5*	*3.4*

Table 1. Main filtering and CL results: mAP@0.5 (in percentage) on COCO val 2017 for WSDDN models trained on CC and RedCaps with the two approaches individually and combined. **Bold**: highest performance per group. *Italic*: methods outperforming the baseline. n/a: model training failed to converge.

Our CLIP score outperforms (or performs on par with) concreteness and results in more similar performance across datasets (especially for CL). This is compatible with the observation that CLIP is a general-purpose tool trained on a variety of images, while concreteness was developed for manually curated datasets (i.e., COCO and Flickr30K [14]), which may make it less suitable for in-the-wild datasets.

We next compare CL with alignment scores with single-modality caption-based and image-based scores. These are effective at boosting performance (mAP@0.5=2.9% for

proposal size on CC, and mAP@0.5=4.4% for number of mentioned objects on RedCaps). However, the boost they provide is not reliable, with a score helping on one dataset but not on the other (e.g., number of mentioned objects, proposal number). Such unreliability is further exemplified by results using caption length on CC, where training diverges.

Finally, we report results for the combination of the best filtering strategy (CLIP) and two CL strategies (i.e., proposal size and caption length). When keeping 50% of the data followed by CL, performance increases (compared to the corresponding row under CL) for caption length on RedCaps (mAP@0.5=3.6% vs 2.5%), but otherwise decreases, especially on CC. Sub-optimal performance for CC was expected since the filtering approach under-performs the baseline. Overly reducing the size of the training set is expected to lead to decreased performance, and our results indicate we have reached that point with CC (but not for RedCaps, which is twice as big as CC). To prove this, we repeat the combination experiments with 75% of the data (instead of 50%), reporting substantially higher performance than the baseline for our proposed filtering and CL on both CC and RedCaps (mAP@0.5=0.9% to 5.8% and from 3.6% to 5.2% for CL with caption length, respectively). This combination achieves the best results across all methods in Table 1. These results show how combining filtering and CL could make training not only more efficient but more effective, too.

4.4. Analysis on hyperparameter choice and training schedule

Next, we investigate the recent claim that CL may only benefit training of models with the Adam optimizer [16] when suboptimal hyperparameters are chosen [39]. For this reason, we test 1e-4 as the learning rate and re-run our experiments on in-the-wild datasets using this decreased (and more optimal) learning rate. Gains from curriculum learning with suboptimal hyperparameters are valuable since hyperparameter search can be costly.

Finally, following previous work reporting a beneficial impact on performance for CL when training resources are limited [41], we repeat select experiments with optimized learning rate, but shorter training time: 1 epoch for the baseline and 2 epochs for filtering and CL (which see the data the same number of times or fewer times than the baseline: 0.5+0.5 or 0.25+0.5, respectively).

4.4.1 Curriculum learning and learning rate

We report results for experiments with lower learning rate on CC and RedCaps in Table 2. We observe that curriculum learning and filtering still provide gains over the baseline results. In the case of filtering, three of the four methods outperform the baseline, compared to just one in Table 1. Gains

	CC	RedCaps
Baseline	10.3	10.3
Filtering (keep 50% of data)		
CLIP	***10.5***	*10.6*
Concreteness	10.3	***11.0***
Curriculum learning		
Caption length	9.6	***10.9***
Num. mentioned objects	9.0	9.7
Proposal size	9.4	***10.9***
Proposal number	9.8	*10.6*
CLIP	***10.8***	*10.4*
Concreteness	9.6	10.3
Filtering (50%) + curriculum learning		
CLIP + Caption length	***10.7***	***10.8***
CLIP + Proposal size	10.2	*10.6*

Table 2. Decreased learning rate from 1e-3 to 1e-4. mAP@0.5 on COCO val 2017 for WSDDN models trained on CC, RedCaps.

	CC	RedCaps
Baseline	9.7	9.4
Filtering		
CLIP	***10.5***	*10.9*
Concreteness	***10.5***	***11.0***
Curriculum learning		
CLIP	***10.5***	***11.5***
Concreteness	9.6	*10.9*

Table 3. Shorter training schedule (lr=1e-4, showing mAP@0.5).

are more pronounced on RedCaps, and using CLIP for CL. However, the best-performing single-modality methods using CL now fail to outperform the baseline (e.g., mAP@0.5=2.9% for proposal size in Table 1 and 2.2% for the baseline on CC, vs 9.4% in Table 2 and 10.3% for the baseline). This result partly confirms the claims advanced in [39] for NLP applications, but in the new context of computer vision. Note that CL using CLIP outperforms the baseline using both learning rates.

4.4.2 Shorter training schedule

Table 3 reports results for select experiments with optimized learning rate and reduced training time. Comparing Table 3 and Table 2, we observe that absolute gains over the baseline are increased in Table 3. For example, CL using CLIP gains 0.8 percentage points (=10.5-9.7) over the baseline on CC, and 2.1 on RedCaps, in Table 3. In Table 2, these gains are 0.5 and 0.1, respectively. This verifies previous claims that CL is most beneficial when training resources are lim-

	100%	50%
GT labels*	16.5*	6.2*
Baseline	10.7	2.0
Curriculum learning - alignment		
CLIP	*10.9*	*3.8*
Concreteness	***13.7***	***5.6***
Curriculum learning - GT info		
Num. objects	***13.0***	*5.8*
Obj. size	6.6	***6.0***

Table 4. Training on COCO using 100% and a random 50% subset: baseline and CL mAP@0.5 (in percentage) on COCO val 2017. The * denotes upper bound.

ited [41], a result that has not been shown in the context of vision-language training or WSOD before.

4.5. Analysis on COCO

While we focus on in-the-wild datasets, we also test whether our conclusions hold on traditional, manually curated, "clean" datasets like COCO Captions. Table 4 reports results for WSDDN models trained on labels extracted from the full (i.e., 100%) COCO Captions data and on a random 50% subset. First, we report both a baseline result, and an upper-bound result with ground-truth labels (rather than those extracted from captions, as the baseline does). Following previous work, we expect CL to be the most beneficial when training data is limited [41]. Indeed, we observe larger gains under the 50% setting (e.g., mAP@0.5=2.0% for the baseline to 3.8% for CLIP and 5.6% for concreteness, vs 10.7% to 10.9% and 13.7% in the 100% setting). This is the first time the impact of CL under a limited-data setting is reported for WSOD from caption supervision. Note that concreteness outperforms CLIP-based alignment, which is reasonable since concreteness was developed for COCO [13].

Our focus is on using weak labels, but, for comparison, we also test CL with ground-truth bounding box information. CL with number of objects and object size outperforms CL using alignment very slightly, or even underperforms (in the 100% setting). CL with concreteness achieves performance similar to CL with number of objects, highlighting the promise of alignment-based CL. Finally, CL with concreteness almost closes the gap between training with caption-extracted labels and with GT (mAP@0.5=5.6% vs. 6.2%) in the 50% setting.

5. Discussion and Conclusions

In this work, we compared ways to train WSOD models with caption-extracted supervision on in-the-wild datasets in a more efficient and effective way using curriculum learning, which has not been evaluated in this setting before. We tested a diverse spectrum of difficulty scores. We showed the benefit of image-caption alignment to boost performance, with improvements consistent across choices of hyperparameters, training schedules, and datasets. We report how such benefit is more pronounced when training time or data is limited, or for sub-optimal choices of learning rate, compatibly with recent claims in the NLP field [39].

Our study has some limitations: first, we used verbatim mentions of objects of interest to extract labels from captions. This method's low recall [42] shows we are discarding image-caption pairs with supervisory signal. We could use a different extraction method including a list of synonyms for class names or train a model to extract labels from captions. Additionally, we only considered a step pacing function; alternative pacing functions may lead to better results, although previous work has shown that the choice of pacing function does not seem to significantly impact performance [37].

Acknowledgement

This work was supported by National Science Foundation Grant No. 2006885.

Ethical Statement

Our study aims to improve training of deep learning models, which may exacerbate biases learned from datasets. While the use of in-the-wild datasets should mitigate bias-related concerns, the authors of RedCaps report on Reddit's inherent bias [6]; for instance, Reddit users skew male, young, college educated, and white.

References

[1] Pablo Arbeláez, Jordi Pont-Tuset, Jonathan T Barron, Ferran Marques, and Jitendra Malik. Multiscale combinatorial grouping. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 328–335, 2014. 3, 6

[2] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48, 2009. 3, 4

[3] 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, 5

[4] Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3558–3568, 2021. 1

[5] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick.

Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015. 1, 5

[6] Karan Desai, Gaurav Kaul, Zubin Aysola, and Justin Johnson. RedCaps: Web-curated image-text data created by the people, for the people. In *NeurIPS Datasets and Benchmarks*, 2021. 1, 5, 8

[7] Alex Fang, Gabriel Ilharco, Mitchell Wortsman, Yuhao Wan, Vaishaal Shankar, Achal Dave, and Ludwig Schmidt. Data determines distributional robustness in contrastive language image pre-training (clip). In *International Conference on Machine Learning*, pages 6216–6234. PMLR, 2022. 3, 5

[8] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. 5

[9] Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. Open-vocabulary object detection via vision and language knowledge distillation. In *International Conference on Learning Representations*, 2022. 2

[10] Guy Hacohen and Daphna Weinshall. On the power of curriculum learning in training deep networks. In *International Conference on Machine Learning*, pages 2535–2544. PMLR, 2019. 3

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 37(9):1904–1916, 2015. 5

[12] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 2

[13] Jack Hessel, David Mimno, and Lillian Lee. Quantifying the visual concreteness of words and topics in multimodal datasets. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2018. 2, 3, 8

[14] Mark J. Huiskes, Bart Thomee, and Michael S. Lew. New trends and ideas in visual concept detection: the mir flickr retrieval evaluation initiative. In *Multimedia Information Retrieval*, 2010. 6

[15] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning*, pages 4904–4916. PMLR, 2021. 1

[16] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 7

[17] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06)*, pages 2169–2178. IEEE, 2006. 5

[18] Feng Liang, Bichen Wu, Xiaoliang Dai, Kunpeng Li, Yinan Zhao, Hang Zhang, Peizhao Zhang, Peter Vajda, and Diana Marculescu. Open-vocabulary semantic segmentation with mask-adapted clip. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7061–7070, 2023. 2

[19] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 6

[20] Matthias Minderer, Alexey Gritsenko, Austin Stone, Maxim Neumann, Dirk Weissenborn, Alexey Dosovitskiy, Aravindh Mahendran, Anurag Arnab, Mostafa Dehghani, Zhuoran Shen, et al. Simple open-vocabulary object detection with vision transformers. *arXiv preprint arXiv:2205.06230*, 2022. 2

[21] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 1, 2, 3

[22] 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*, 28, 2015. 2

[23] Shuhuai Ren, Aston Zhang, Yi Zhu, Shuai Zhang, Shuai Zheng, Mu Li, Alex Smola, and Xu Sun. Prompt pre-training with twenty-thousand classes for open-vocabulary visual recognition. *arXiv preprint arXiv:2304.04704*, 2023. 2

[24] Zhongzheng Ren, Zhiding Yu, Xiaodong Yang, Ming-Yu Liu, Yong Jae Lee, Alexander G. Schwing, and Jan Kautz. Instance-aware, context-focused, and memory-efficient weakly supervised object detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2, 5

[25] Christoph Schuhmann, Robert Kaczmarczyk, Aran Komatsuzaki, Aarush Katta, Richard Vencu, Romain Beaumont, Jenia Jitsev, Theo Coombes, and Clayton Mullis. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. In *NeurIPS Workshop Datacentric AI*, number FZJ-2022-00923. Jülich Supercomputing Center, 2021. 3

[26] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade W Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa R Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. LAION-5b: An open large-scale dataset for training next generation image-text models. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022. 1, 3

[27] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2556–2565, 2018. 1, 5

[28] Hengcan Shi, Munawar Hayat, Yicheng Wu, and Jianfei Cai. Proposalclip: unsupervised open-category object proposal generation via exploiting clip cues. In *CVPR*, 2022. 2, 3

[29] Miaojing Shi and Vittorio Ferrari. Weakly supervised object localization using size estimates. In *European Conference on Computer Vision*, pages 105–121. Springer, 2016. 2, 3

[30] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 5

[31] Petru Soviany, Claudiu Ardei, Radu Tudor Ionescu, and Marius Leordeanu. Image difficulty curriculum for generative adversarial networks (cugan). In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 3463–3472, 2020. 3

[32] Peng Tang, Xinggang Wang, Xiang Bai, and Wenyu Liu. Multiple instance detection network with online instance classifier refinement. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2843–2851, 2017. 2, 5

[33] Peng Tang, Xinggang Wang, Song Bai, Wei Shen, Xiang Bai, Wenyu Liu, and Alan Yuille. Pcl: Proposal cluster learning for weakly supervised object detection. *IEEE transactions on pattern analysis and machine intelligence*, 42(1): 176–191, 2018. 2, 5

[34] Radu Tudor Ionescu, Bogdan Alexe, Marius Leordeanu, Marius Popescu, Dim P Papadopoulos, and Vittorio Ferrari. How hard can it be? estimating the difficulty of visual search in an image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2157–2166, 2016. 3

[35] 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. 3, 6

[36] Mesut Erhan Unal, Keren Ye, Mingda Zhang, Christopher Thomas, Adriana Kovashka, Wei Li, Danfeng Qin, and Jesse Berent. Learning to overcome noise in weak caption supervision for object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 2, 5

[37] Xin Wang, Yudong Chen, and Wenwu Zhu. A survey on curriculum learning. *TPAMI*, 44(9):4555–4576, 2021. 3, 8

[38] Yuting Wang, Ricardo Guerrero, and Vladimir Pavlovic. D2f2wod: Learning object proposals for weakly-supervised object detection via progressive domain adaptation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 22–31, 2023. 2, 5

[39] Lucas Weber, Jaap Jumelet, Paul Michel, Elia Bruni, and Dieuwke Hupkes. Curriculum learning with adam: The devil is in the wrong details. *arXiv preprint arXiv:2308.12202*, 2023. 7, 8

[40] Size Wu, Wenwei Zhang, Sheng Jin, Wentao Liu, and Chen Change Loy. Aligning bag of regions for open-vocabulary object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15254–15264, 2023. 2

[41] Xiaoxia Wu, Ethan Dyer, and Behnam Neyshabur. When do curricula work? In *ICLR*, 2021. 7, 8

[42] Keren Ye, Mingda Zhang, Adriana Kovashka, Wei Li, Danfeng Qin, and Jesse Berent. Cap2det: Learning to amplify weak caption supervision for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 5, 8

[43] Zhaoyang Zeng, Bei Liu, Jianlong Fu, Hongyang Chao, and Lei Zhang. Wsod2: Learning bottom-up and top-down objectness distillation for weakly-supervised object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8292–8300, 2019. 5

[44] Dingwen Zhang, Junwei Han, Long Zhao, and Deyu Meng. Leveraging prior-knowledge for weakly supervised object detection under a collaborative self-paced curriculum learning framework. *International Journal of Computer Vision*, 127(4):363–380, 2019. 2, 3