ScaleDet: A Scalable Multi-Dataset Object Detector

Yanbei Chen,   Manchen Wang,   Abhay Mittal,   Zhenlin Xu,
Paolo Favaro,   Joseph Tighe,   Davide Modolo
AWS AI Labs

Abstract

Multi-dataset training provides a viable solution for exploiting heterogeneous large-scale datasets without extra annotation cost. In this work, we propose a scalable multi-dataset detector (ScaleDet) that can scale up its generalization across datasets when increasing the number of training datasets. Unlike existing multi-dataset learners that mostly rely on manual relabelling efforts or sophisticated optimizations to unify labels across datasets, we introduce a simple yet scalable formulation to derive a unified semantic label space for multi-dataset training. ScaleDet is trained by visual-textual alignment to learn the label assignment with label semantic similarities across datasets. Once trained, ScaleDet can generalize well on any given upstream and downstream datasets with seen and unseen classes. We conduct extensive experiments using LVIS, COCO, Objects365, OpenImages as upstream datasets, and 13 datasets from Object Detection in the Wild (ODinW) as downstream datasets. Our results show that ScaleDet achieves compelling strong model performance with an mAP of 50.7 on LVIS, 58.8 on COCO, 46.8 on Objects365, 76.2 on OpenImages, and 71.8 on ODinW, surpassing state-of-the-art detectors with the same backbone.

1. Introduction

Major advances in computer vision have been driven by large-scale datasets, such as ImageNet [9] and OpenImages [22] for image classification, or Kinetics [6] and ActivityNet [2] for video recognition. Large-scale datasets are crucial for training recognition models that generalize well. However, the collection of massive annotated datasets is costly and time-consuming. This is especially prominent in detection and segmentation tasks that require detailed annotations at the bounding box or pixel level. To exploit more training data without extra annotation cost, recent works unify multiple datasets to learn from more visual categories and more diverse visual domains for detection [38, 40, 47, 51] and segmentation [25, 37].

To train an object detector across multiple datasets, we need to tackle several challenges. *First*, multi-dataset train-

Figure 1. Our scalable multi-dataset detector (ScaleDet) learns across datasets in a unified semantic label space by visual-textual alignment with label semantic similarities. At test time, ScaleDet can generalize on any given upstream and downstream dataset.

ing requires unifying the heterogeneous label spaces across datasets, as label definitions are dataset-specific. The labels from two datasets may indicate the same or similar objects. For example, "footwear" and "sneakers" are two different labels in OpenImages [24] and Objects365 [34], but refer to the same type of objects (see Figure 1). *Second*, the training setups may be inconsistent among datasets, as different data sampling strategies and learning schedules are often required for datasets of different sizes. *Third*, a multi-dataset model should perform better than single-dataset models on individual datasets. This is challenging due to the heterogeneous label spaces, the domain discrepancy across datasets, and the risk of overfitting to the larger datasets.

To resolve the above challenges, existing work resorts to manually relabelings class labels [25], or training multiple dataset-specific classifiers with constraints to relate labels across datasets [51]. However, these methods lack scalability. The manual relabeling effort and the model complex-

ity of training multiple classifiers grow rapidly as the number of datasets increases. We overcome this limitation with **ScaleDet**: a scalable multi-dataset detector (Figure 1). We propose two innovations: *a scalable formulation* to unify multiple label spaces, and *a novel loss formulation* to learn hard label and soft label assignments across datasets. While hard label assignment serves to disambiguate class labels in probability space, soft label assignment works as a regularizer to relate class labels in semantic similarity space. Unlike existing multi-dataset methods [25,37,38,40,47,51] that mostly generalize on seen datasets or seen classes, our method exploits vision-language learning to attain good generalization on both upstream and downstream datasets, where the downstream datasets can contain unseen classes and new domains. Our contributions are:

- We propose a novel scalable multi-dataset training recipe for object detection. Our method utilizes text embeddings to unify and relate labels with semantic similarities across datasets, and trains a single classifier via visual-textual alignment to learn hard label and soft label assignments.

- We conduct extensive experiments to demonstrate the compelling scalability and generalizability of ScaleDet in multi-dataset training. We show that ScaleDet can boost its performance as we increase the number of training datasets: LVIS [14], COCO [27], Objects365 [34] and OpenImages [24] (Sec 4.2). Furthermore, we show that ScaleDet achieves state-of-the-art performance on multiple benchmarks when compared to recent advanced detectors, e.g., Detic [49], UniDet [51] (Sec 4.3, Sec 4.4).

- We evaluate the transferablity of ScaleDet on the challenging "Object Detection in the Wild" benchmark (which contains 13 datasets) [26] to demontrate its competitive generalizability on downstream datasets (Sec 4.5).

2. Related Work

Multi-dataset training aims to leverage multiple datasets to train more generalizable visual recognition models, especially for tasks that require more expensive annotations, such as detection [38,40,47,51] and segmentation [25,37]. Existing methods for multi-dataset training can be categorized into two groups. The first group introduces special network components to adapt feature representations tailored to datasets [38,40]. For example, a domain attention module is designed to learn an adaptive multi-domain detector, which assigns different network activations to different domains [38]. The second group introduces new formulations to combine heterogeneous label spaces over multiple datasets [25,37,47,51]. For instance, the MSeg dataset [25] is created upon multiple semantic segmentation datasets by relabeling with Amazon Mechanical Turk based on a manually defined taxonomy of class labels. To avoid manual relabeling, a pseudo labeling strategy [47] is used to generate pseudo labels across datasets based on predictions from

dataset-specific detectors. Recent works use set theory with manually defined sets [37] or employ combinatorial optimization [51] to learn the label relations across datasets. Our approach is more related to the second group, but it offers a more scalable and generalizable training recipe that does not require training multiple dataset-specific classifiers [47, 51], nor manually designed rules to relate class labels [25, 37]. Furthermore, different from existing multi-dataset detectors [38,40,47,51], ScaleDet is also capable of generalizing to datasets that contain unseen classes.

Vision-language models (VLMs) employ vision-and-language learning to solve visual recognition problems. By bridging large-scale vision and language data, vision-language models (VLMs) such as VirTex [10], CLIP [31], ALIGN [19], FILIP [43], UniCL [42] and LiT [45], have shown remarkable performance on learning transferable visual representations that generalize well to downstream tasks. More recently, VLMs have been explored in segmentation [11, 41] and detection [3, 13, 20, 26, 46, 48, 49]. In detection, most of the VLMs utilize auxiliary semantic-rich vision-language datasets as pre-training data to build models capable of solving multiple tasks including open-vocabulary detection [44], as represented by MDETR [20], RegionCLIP [48], GLIP [26, 46], X-DETR [3], and Detic [49]. For instance, MDETR uses various vision-and-language datasets (e.g., Flickr30k [30], Visual genome [23]) to train a VLM that allows instance-wise detection guided by text query. GLIP [26] uses 27M grounding data to build a unified VLM for detection and grounding. X-DETR [3] trains a VLM on grounding and image-caption datasets to solve multiple instance-wise vision-language tasks in one model. Detic [49] uses detection and large-scale classification datasets to train a large-vocabulary detector. We exploit a pre-trained text encoder similar to other VLMs, e.g., RegionCLIP uses CLIP, while MDETR, X-DETER use RoBERTa. However, our model is trained directly on detection datasets, while these other VLMs use image captioning or grounding datasets to train jointly with detection datasets.

3. ScaleDet: A Scalable Multi-Dataset Detector

Our goal is to train object detectors that generalize better as we increase the number of datasets used for training. A critical challenge of multi-dataset training is unifying the heterogeneous label spaces across datasets by relating their semantic concepts. To tackle this challenge, we propose a simple yet effective recipe to train a scalable multi-dataset detector (ScaleDet, Figure 2). ScaleDet learns across multiple datasets by unifying different label sets to form a unified semantic label space (Figure 2 top), and is trained via hard label and soft label assignments for visual-textual alignment (Figure 2 bottom). In Sec 3.2 and Sec 3.3 we present the details of these two contributions, but first we discuss the preliminaries and problem formulation in Sec 3.1.
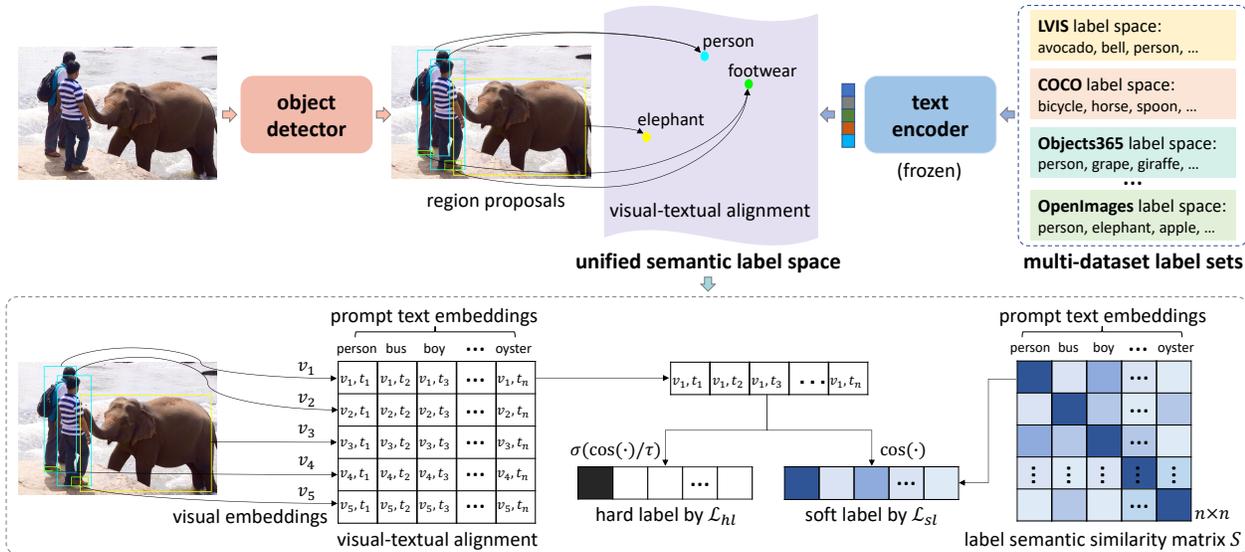
Figure 2. Approach overview. Our scalable multi-dataset detector (ScaleDet) uses text prompts to define labels, and learns across datasets by visual-textual alignment in the unified label space. Each visual feature of an object region proposal is assigned to one corresponding label via a hard label assignment (see the loss $\mathcal{L}_{hl}$ in Eq. (6)) and related to semantically-similar labels via a soft label assignment (see the loss $\mathcal{L}_{sl}$ in Eq. (7)). At test time, ScaleDet can perform object detection on any given label space of a seen or unseen detection dataset.

3.1. Preliminaries and problem formulation

Standard object detection. A typical object detector aims to predict the bounding box location $b_i \in \mathbf{R}^4$ and class label $c_i \in \mathbb{R}^n$ for any object that belongs to a predefined set of n classes. Given an image I, the image encoder (e.g., CNN or Transformer) of the detector extracts the box features and visual features, which are fed to a box regressor B and a visual classifier C. The detector is trained by minimizing a bounding box regression loss $\mathcal{L}_{bbox}$ and a classification loss $\mathcal{L}_{cls}$ to learn the predictions of bounding boxes and class labels given the box features and visual features, i.e.,

$$\mathcal{L}_{Det} = \mathcal{L}_{bbox} + \mathcal{L}_{cls}. \quad (1)$$

Existing object detectors generally adopt a one-stage [36, 52] or two-stage [15, 32] framework, which may contain additional loss terms in Eq. (1). For example, one-stage detectors use a regression loss to regress towards properties of object locations, such as centerness [36]. Two-stage detectors [15, 32] instead use a region proposal network with its own dedicated loss function that predicts the objectness of each box. In this work, we focus on reformulating the classification loss $\mathcal{L}_{cls}$ in Eq. (1) to solve the problem of multi-dataset training, built upon a two-stage detector [50].

Multi-dataset object detection. Given a set of K datasets $\{\mathcal{D}_1, \mathcal{D}_2, ..., \mathcal{D}_K\}$ with their label spaces $\{L_1, L_2, ..., L_K\}$, we aim at training a scalable multi-dataset detector that generalizes well on upstream and downstream detection datasets. In contrast to prior multi-dataset learners [25, 37, 51], which relate or merge similar labels across datasets to joint labels, we propose a simple yet scalable formula-

tion to unify labels without merging any labels explicitly. We exploit informative text embeddings from powerful pre-trained visual-language models to define and relate labels over the non-identical label spaces $\{L_1, L_2, ..., L_K\}$. We present our method in the following sections.

3.2. Scalable unification of multi-dataset label space

As Figure 2 (top) shows, during training, a mini-batch of images is randomly sampled from multiple training sets and fed to the detector to obtain the visual features $\{v_1, v_2, ..., v_j\}$ of the region proposals for each image, where $v_i \in \mathbf{R}^D$ is a D-dimensional vector. Each visual feature v_i is matched to a set of text embeddings $\{t_1, t_2, ..., t_n\}$ by label assignment. Below, we detail how we define semantic labels with text prompts, and unify label spaces with label semantic similarities for multi-dataset training.

Define labels with text prompts. We represent each class label l_i with its text prompts, e.g., the label *"person"* can be denoted by a text prompt *"a photo of a person"*. We extract the prompt text embeddings from the text encoder of a pre-trained visual-language model (e.g., CLIP [31] or OpenCLIP [18]), and average embeddings of a set of predefined text prompts (known as *prompt engineering* [31]) to represent the label l_i as a semantic text embedding t_i.

Unify label spaces by concatenation. Given the text embeddings of class labels from all datasets, a crucial problem of multi-dataset training is to unify the non-identical label spaces $\{L_1, L_2, ..., L_K\}$, which can be solved by relating and merging similar labels into a unified label set [25]. However, without careful manual inspection, this incurs the

risk of propagating errors in model training due to the ambiguities in label definitions, e.g., labels *"boy"* and *"girl"* are similar but should not be merged. Thus, instead of merging labels across datasets, we unify different label spaces by concatenation:

$$L = L_1 \coprod ... \coprod L_K = \{l_{1,1}, l_{1,2}, ..., l_{K,1}, l_{K,2}, ...\}, \quad (2)$$

where $\coprod$ denotes the coproduct (i.e., the disjoint union of two label sets); $l_{k,i}$ is label i from dataset k (we omit k below). Besides its simplicity, our formulation of this unified semantic label space L maximally preserves the semantics of all labels, thus providing richer vocabulary for training.

Relate labels by semantic similarities. As we use text embeddings to represent class labels, we can relate the labels that share similar semantics, in our unified label space. For example, label *"person"* in LVIS [14] should be related to labels *"person"* and *"boy"* in OpenImages [24], as they are semantically similar. To uncover the label relations across datasets, we compute the semantic similarities using the prompt text embeddings. For a given class label l_i, its semantic similarities with respect to all the labels are derived with cosine similarities and normalized between 0 and 1:

$$\text{sim}(l_i, l_j) = \frac{\cos(t_i, t_j) - \alpha_i}{\beta_i - \alpha_i},$$
$$\text{where } \alpha_i = \min\{\cos(t_i, t_j)\}_{j=1}^n, \quad (3)$$
$$\beta_i = \max\{\cos(t_i, t_j)\}_{j=1}^n = \cos(t_i, t_i) = 1,$$

where $\text{sim}(l_i, l_j)$ is the semantic similarity between the text embeddings t_i, t_j of two labels l_i, l_j and it encodes their semantic relation, e.g., the labels *"person"* and *"boy"* are strongly related with a high similarity score, while *"person"* and *"avocado"* are weakly related with low similarity.

The label semantic similarity matrix S that encodes label relations among all n class labels can be written as:

$$S = \begin{bmatrix} 1 & \cdots & \text{sim}(l_1, l_n) \\ \vdots & \ddots & \vdots \\ \text{sim}(l_n, l_1) & \cdots & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{s}_1 \\ \vdots \\ \mathbf{s}_n \end{bmatrix}, \quad (4)$$

where S is an $n \times n$ matrix, and each row vector $\mathbf{s}_i$ encodes the semantic relations of label l_i with respect to all n class labels. With these label semantic similarities, we can introduce explicit constraints that allow the detector to learn on the unified semantic label space (Eq. (2)) with encoded label semantic similarities (Eq. (4)). Importantly, our formulas (Eq. (2), Eq. (4)) are computed offline, which does not add any computational cost for training and inference, nor requires a model reformulation when scaling up the number of training datasets.

3.3. Training with visual-language alignment

For training on the unified semantic label space $\{l_1, l_2, ..., l_n\}$, we align visual features with text embeddings $\{t_1, t_2, ..., t_n\}$ via the hard label and soft label assignments, as shown in Figure 2 (bottom) and detailed below.

Visual-language similarities. Given the visual feature v_i of an object region proposal, we first compute the cosine similarities between v_i and all text embeddings $\{t_1, t_2, ..., t_n\}$:

$$\mathbf{c}_i = [\cos(v_i, t_1), \cos(v_i, t_2), ..., \cos(v_i, t_n)]. \quad (5)$$

With these similarity scores, we can align the visual feature v_i to the text embeddings with the following loss terms.

Hard label assignment. Each visual feature v_i has its groundtruth label l_i, and thus can be matched to the text embedding t_i of l_i by hard label assignment as follows,

$$\mathcal{L}_{hl} = \text{BCE}(\sigma_{sg}(\mathbf{c}_i / \tau), l_i), \quad (6)$$

where $\text{BCE}(\cdot)$ is the binary cross-entropy loss, $\sigma_{sg}(\cdot)$ is the sigmoid activation, τ is a temperature hyperparameter. Eq. (6) ensures the visual feature v_i is aligned with the text embedding t_i. However, it does not explicitly learn the label relations across datasets. We introduce the soft label assignment to learn the semantic label relations.

Soft label assignment. As each label is semantically related to all class labels through the semantic similarities computed in Eq. (4), a visual feature can also be related to all text embeddings by using the semantic similarity scores. For this aim, we introduce the soft label assignment on the visual feature v_i:

$$\mathcal{L}_{sl} = \text{MSE}(\mathbf{c}_i, \mathbf{s}_i), \quad (7)$$

where $\text{MSE}(\cdot)$ is mean square error. $\mathbf{s}_i$ denotes the semantic similarities between the label l_i and all n class labels – the i_{th} row of the label semantic similarity matrix S in Eq. (3).

Remark. While hard label assignment (Eq. (6)) is imposed in probability space to disambiguate different class labels, soft label assignment (Eq. (7)) is imposed in semantic similarity space to assign each visual feature to the text embeddings with different semantic similarities, thus acting as a regularizer to relate similar class labels across datasets.

Training with semantic label supervision. With the hard label and soft label assignments, we train the detector to classify different region proposals by aligning visual features to text embeddings in the unified semantic label space. That is, the classification loss $\mathcal{L}_{cls}$ in Eq. (1) is replaced by

$$\mathcal{L}_{lang} = \mathcal{L}_{hl} + \lambda \mathcal{L}_{sl}, \quad (8)$$

where λ is a hyperparameter to balance the two loss terms. Eq. (8) uses language supervision to map images to texts, thus enabling zero-shot detection of unseen labels.

Overall objective. As we focus on reformulating the classification loss of our detector for multi-dataset training, we do not change the loss $\mathcal{L}_{bbox}$ in Eq. (1) and our overall objective for training ScaleDet is:

$$\mathcal{L}_{ScaleDet} = \mathcal{L}_{bbox} + \mathcal{L}_{lang}. \quad (9)$$

Once trained with $\mathcal{L}_{ScaleDet}$, ScaleDet can be deployed on any upstream or downstream datasets that contain seen or unseen classes. By replacing the unified label space L in Eq. (2) with the label space of any given test dataset, ScaleDet can compute the label assignment based on the visual-language similarities derived in Eq. (3). When the test dataset contains unseen classes, the evaluation setting is known as zero-shot detection [1] or open-vocabulary object detection [44]. To test on any given dataset, ScaleDet can either be directly evaluated or fine-tuned before evaluation.

4. Experiments

We detail our experiment setups in Sec 4.1. We analyze ScaleDet when training with an increasing number of datasets in Sec 4.2, and compare it to the state-of-the-art (SOTA) on standard benchmarks in Sec 4.3 and Sec 4.4. We evaluate the transferability of ScaleDet on downstream datasets in Sec 4.5, and conduct ablation study in Sec 4.6. We also provide qualitative visual results in *Supplementary*.

4.1. Experiment setups

Upstream datasets. For our multi-dataset training, we adopt the following four popular detection datasets as upstream datasets: (1) COCO [27] (C) contains 80 common object categories; (2) LVIS [14] (L) has a large number of 1203 object categories with a challenging long tail distribution; (3) Objects365 [34] (O365) has 365 object categories; (4) OpenImages detection (OID) [24] has 601 object categories in the sixth version. When we train ScaleDet on these four datasets, there are 2249 class labels in the unified label space (Eq. (2)) and 3.7M training images.

Downsteam datasets. To evaluate the transferability on diverse unseen datasets, we adopt the recent challenging "Object Detection in the Wild" (ODinW) benchmark [26]. It contains 13 public object detection datasets with very different application domains, to mimic diverse and challenging real-world scenarios. Some of these datasets capture seen classes in unseen domains, like people in thermal images, while others capture unseen classes, like "jellyfish" and "stingray" in aquariums (Figure 1, bottom right). To evaluate on ODinW, the models can be evaluated directly (direct transfer) or evaluated after fine-tuning (fine-tune transfer).

Evaluation metrics. For evaluation on *upstream* datasets, we use the standard mAP metric (i.e., mAP at IoU thresholds 0.5 to 0.95) on COCO, LVIS and Objects365. On OpenImages, we follow the official evaluation protocol that uses mAP@0.5 and an expanded semantic class hierarchy [24]. For evaluation on *downstream* datasets, we follow [26] and report the average mAP over 13 datasets in this section. We provide the detailed qualitative and quantitative results on individual datasets in *Supplementary*.

Implementation details. In our experiments, unless explic-

Model	Dataset(s)	L	C	O365	OID	mAP
baseline	L	33.1	37.0	15.2	41.5	31.7
	C	11.0	46.8	7.9	33.1	24.7
	O365	19.2	39.8	28.8	47.6	33.9
	OID	15.7	31.3	14.1	69.3	32.6
ScaleDet	L,C	33.3	44.9	15.9	43.7	34.5
	L,C,O365	36.5	47.0	**31.2**	44.9	39.9
	L,C,O365,OID	**36.8**	**47.1**	30.6	**69.4**	**46.0**

Table 1. **Training with an increasing number of datasets.** The best results on single-/multi-dataset training are in green/**bold**. Backbone: ResNet50. mAP is the mean AP across datasets.

itly stated, we use CenterNet2 [50] with a backbone pretrained on ImageNet21k [33]. We use prompt text embeddings from CLIP [31] or OpenCLIP [18] to encode class labels. For augmentation, we use large scale jittering [12] and efficient resize crop [49] with an input size of 640×640, 896×896 when using ResNet50 [16], Swin Transformer [28] as backbone. We use an input size of 800×1333 for testing. We use Adam optimizer [21] and train on 8 V100 GPUs. For multi-dataset training, we directly combine all datasets and use repeat factor sampling [14], without using any multi-dataset sampling strategy. More details about learning schedules of different tables are in *Supplementary*.

4.2. Training with a growing number of datasets

Evaluation setup. We setup the single-dataset *baseline* by training on individual datasets. We use text embeddings to represent the class labels for each dataset, and train the detector on each dataset by hard label assignment (Eq. (6)). To evaluate how ScaleDet scales as we increase the number of training datasets, we train it using the following incremental combinations: (1) LVIS+COCO, (2) LVIS+COCO+O365, (3) LVIS+COCO+O365+OID. These increase the dataset size from $218k \rightarrow 1.96M \rightarrow 3.7M$ and the number of class labels from $1283 \rightarrow 1648 \rightarrow 2249$. After training the baseline and ScaleDet, we replace the text embeddings in Eq. (5) to represent the labels of each dataset for evaluation. For example, to evaluate on COCO we set the text embeddings to represent the 80 classes of COCO and derive class predictions based on visual-textual similarities.

To make the evaluation efficient, for the results in Table 1, Figure 3 and Figure 4 we use a fixed input size of 640×640, a ResNet50 [16] backbone and text embeddings from CLIP [31]. We then explore using larger backbones (Swin Transformers [28]) and text embeddings from Open-CLIP [18] for the analysis of backbones in Table 2.

Effect on upstream datasets. Table 1 shows the evaluation on upstream datasets when increasing the number of datasets, from which we can observe the followings. *First*, scaling up the number of training datasets consistently leads to better model performance, e.g., the mAP on
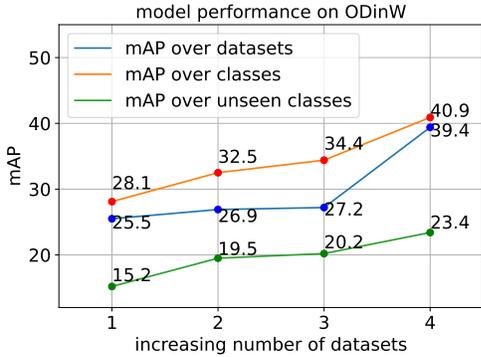
Figure 3. Direct transfer results on ODinW benchmark when scaling up the number of training datasets. Number 1-4 in x-axis means using L, L+C, L+C+O365, L+C+O365+OID for training. Details of unseen classes, qualitative results are in *Supplementary*.
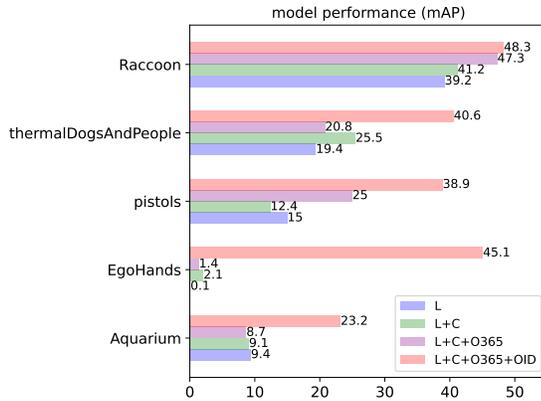
Figure 4. Direct transfer to different datasets on ODinW benchmark when scaling up the number of training datasets.

LVIS is improved from 33.1, 33.3, 36.5, to 36.8 when increasing the number of training sets from 1 to 4. *Second*, multi-dataset training with ScaleDet is generally better than single-dataset training (baseline), e.g., the mAP on LVIS, COCO, O365, OID are greatly improved from 33.1, 46.8, 28.8, 69.3 (baseline) to 36.8, 47.1, 30.6, 69.4 when training on all datasets. This suggests that ScaleDet learns well across heterogeneous label spaces, diverse domains of different datasets, and does not overfit to any specific datasets.

Effect on downstream datasets. Figure 3 shows the results of direct transfer on the ODinW benchmark. Notably, scaling up the number of training datasets for ScaleDet significantly improves its accuracy (mAP) on downstream datasets, for all settings (i.e., over datasets, over classes and over unseen classes). In Figure 4 we further visualize the performance of ScaleDet on some of the downstream datasets in ODinW. These datasets either contain unseen classes or come from the visual domains that are very different from those used for training. Importantly, ScaleDet performs well on both scenarios. For example, *thermalDogsAndPeople* contains novel thermal images, yet multi-dataset training with ScaleDet is capable of boosting its per-

	Model	Text	L	C	O365	OID	mAP
1	ScaleDet-R	CLIP	36.8	47.1	30.6	69.4	46.0
2	ScaleDet-T	CLIP	42.7	51.2	35.9	73.6	50.9
3	ScaleDet-T	OpenCLIP	43.9	51.4	36.3	74.1	51.4
4	ScaleDet-B	CLIP	50.4	55.6	43.7	75.7	56.3
5	ScaleDet-B	OpenCLIP	**50.7**	55.4	43.8	**76.2**	56.5
6	ScaleDet-B*	OpenCLIP	50.6	**58.8**	**46.8**	75.9	**58.0**

Table 2. Different backbones for multi-dataset training. R: ResNet50. T, B: Swin-Tiny, Swin-Base transformer. *: further fine-tune the model on each dataset after multi-dataset training.

formance considerably from 19.4 mAP (single-dataset) to 40.6 (all 4). Furthermore, *Raccoon* requires the localization of an unseen class, yet again ScaleDet improves zero-shot detection from 39.2 mAP (single-dataset) to 48.3 (all 4). Overall, Figure 3 and Figure 4 show how ScaleDet can utilize the synergistic benefits of different upstream datasets to boost its generalization on downstream datasets.

Effect of different backbones. We now test ScaleDet with different backbones and text embeddings (Table 2). Enhancing the discriminative power of the image backbone from ResNet50 [16] to Swin-Base Transformer [28] significantly boosts the mAP from 46.0 to 56.3 (row 1 vs 4). Changing the text embeddings from CLIP to OpenCLIP instead helps mostly when using weaker backbones (e.g., +0.5 mAP with ScaleDet-T, row 2 vs 3), as opposed to stronger ones (e.g., +0.2 mAP with ScaleDet-B, row 4 vs 5). Finally, with fine-tuning ScaleDet can be further improved from 56.5 to 58.0 in mAP (row 5 vs 6). Overall, all these results suggest that ScaleDet can leverage the complementary power of scalable training sets and strong backbones to achieve competitive model performance.

4.3. Comparison to SOTA multi-dataset detectors

In this section, we provide an apple-to-apple comparison of ScaleDet against the two latest state-of-the-art (SOTA) multi-dataset detectors: UniDet [51] and Detic [49].

Comparison to UniDet. In Table 3, we follow the settings of UniDet [51] and train ScaleDet on the same datasets (COCO, O365, OID), using the same ResNet50 backbone, SGD optimizer and standard data augmentation in the Detectron2 codebase [39]. In UniDet, multiple dataset-specific classifiers are trained, while ScaleDet is trained with semantic labels by one classifier. Table 3 shows the comparison on multi-dataset training. ScaleDet outperforms UniDet, yielding an mAP of 47.7 vs 45.4. Moreover, ScaleDet has better improved margins for multi-dataset training over single-dataset training (row 4,3 vs 2,1). The improved margin in mAP is 2.6 points (47.7 vs 45.1) by ScaleDet vs 1.1 points (45.4 vs 44.3) by UniDet. These results show the benefit of learning in a unified semantic label space, as opposed to training multiple dataset-specific classifiers (UniDet).

	Model	Dataset(s)	COCO	O365	OID	mAP
1	UniDet	single	42.5	24.9	65.7	44.3
2		multiple	**45.5**	24.6	66.0	45.4
3	**ScaleDet**	single	42.1	26.5	66.6	45.1
4		multiple	**45.5**	**27.9**	**69.6**	**47.7**

Table 3. Comparison to UniDet [51] on multi-dataset training with COCO, O365 and OID. Backbone: ResNet50 pre-trained on ImageNet1k. "single": training with one single training set same as the test set. "multiple": multi-dataset training with COCO,O365,OID. mAP is the mean AP across datasets.

	Model	Datasets	LVIS	COCO	mAP
1	Detic [49]	L,C	33.0	43.9	38.4
2	**ScaleDet**	L,C	**33.3**	**44.9**	**39.1**
3	Detic [49]	L,C,IN21k	35.4	42.4	38.9
4	**ScaleDet**	L,C,O365	36.5	47.0	41.7
5	**ScaleDet**	L,C,O365,OID	**36.8**	**47.1**	**41.9**

Table 4. Comparison to Detic [49] on multi-dataset training with LVIS (L) and COCO (C). Backbone: ResNet50. IN21k: ImageNet21k [9]. In row 1-2, both Detic and ScaleDet are trained with only LVIS and COCO, thus providing a like-to-like comparison. In row 3, 4, 5, the training data size is 12.6M, 1.96M, 3.7M respectively. mAP is the mean AP across datasets.

Comparison to Detic. In Table 4, we follow the settings of Detic [49] and perform multi-dataset training on LVIS and COCO, using a ResNet50 backbone. In Detic, the unified label space for LVIS and COCO contains 1203 class labels, obtained by merging two label sets with *wordnet synsets*, while ScaleDet 'flattens' their labels (1203+80) to 1283. Table 4 (row 1 and 2) presents the comparison. ScaleDet outperforms Detic by 1 point on COCO (44.9 *vs.* 43.9) and 0.7 mAP on average. Next, we compare ScaleDet and Detic with more training sets (Table 4, row 3-5). ScaleDet is trained using more detection datasets (O365,OID), while Detic uses 14M additional classification images (ImageNet21) as a large-scale weakly annotated dataset for detection. Our results show that, although Detic is trained with far more data and class labels than ScaleDet (12.6M images and 22k classes after data cleaning vs 3.7M images and 2k classes), ScaleDet still surpasses Detic's performance by a large margin of 3.0 mAP points (41.9 *vs.* 38.9). These results show the importance of learning from multiple detection datasets and the effectiveness of ScaleDet in doing so.

4.4. Comparison to SOTA detectors on COCO

In this section, we compare ScaleDet to different types of SOTA detectors on standard COCO benchmark. The different models cover four major types: (1) single-dataset detectors; (2) detectors trained with vision-and-language understanding datasets, e.g., UniT [17] is trained on 7 tasks

	Model	Model Type	mAP
1	Faster RCNN [32]		37.9
2	Mask RCNN [15]		39.8
3	CenterNet [52]	single-dataset	40.2
4	CascadeRCNN [4]	detection	41.6
5	DETR [5]		42.0
6	CenterNet2 [50]		42.9
7	UniT [17]	detection +	42.3
8	RegionCLIP [48]	understanding	42.7
9	Detic [49]	detection + classification	42.4
10	UniDet [51]	multi-dataset	45.5
11	**ScaleDet**	detection	**47.1**

Table 5. Comparison on COCO using ResNet50 as backbone. Note: "understanding", "classification" mean training with vision-and-language understanding, and classification datasets respectively. All methods include COCO dataset for training.

	Model	Model Type	mAP
1	Faster RCNN-T [32]	single-dataset	46.0
2	DyHead-T [8]	detection	49.7
3	CascadeRCNN-T [4]		50.4
4	GLIP-T [26]	detection +	55.2
5	GLIPv2-T [46]	understanding	55.5
6	GLIPv2-B [46]		**58.8**
7	Detic-B [49]	detection + classification	54.9
8	**ScaleDet-B**	multi-dataset detection	**58.8**

Table 6. Comparison on COCO using Swin Transformer as backbone. T, B: Swin-Tiny, Swin-Base Transformer. Note: GLIPv2-B uses 20.5M data: 3.7M detection data (COCO, O365, OID, Visual genome [23], ImageNetBoxes [9]), 16.8M grounding and caption data (GoldG [20], CC15M [7, 35], SBU [29]); while ScaleDet-B uses 3.5M data (LVIS,COCO,O365,OID).

over 8 datasets and RegionCLIP [48] is trained with 3M image-text pairs from Conceptual Caption [35]; (3) detector trained with image classification dataset, i.e., Detic [49] is trained with ImageNet21k [9]; (4) multi-dataset detector trained with only detection datasets, e.g., UniDet [51].

We train our ScaleDet using LVIS, COCO, O365, OID and present the results in Table 5, where all models are trained using a ResNet50 backbone. ScaleDet obtains the best performance among 11 models. Moreover, compared to detectors trained with large vision-and-language or classification datasets (row 7-9), ScaleDet gives much better performance, even though it learns a much smaller number of concepts – ScaleDet achieves 47.1 by "only" learning 2249 labels, while RegionCLIP and Detic achieve 42.7 and 42.4 by learning 6790 and 22047 concepts, respectively.

In Table 6, we compare ScaleDet to competitors using Swin Transformers as backbone. With this strong back-

Model	Model Type	#Data	ODinW	
			direct	fine-tune
GLIP-T [26]	detection + understanding	5.5M	46.5	64.9
GLIPv2-T [46]		5.5M	48.5	66.5
GLIPv2-B [46]		20.5M	**54.2**	69.4
Detic-R [49]	detection + classification	12.6M	29.4	64.4
Detic-B [49]		12.6M	38.7	70.1
ScaleDet-R	detection	3.6M	39.4	68.5
ScaleDet-T		3.6M	44.3	70.4
ScaleDet-B		3.6M	47.3	**71.8**

Table 7. Results of direct and fine-tune transfer on ODinW. R, T, B: ResNet50, Swin-Tiny, Swin-Base Transformer as backbone. Metric: mAP over 13 datasets. Note: "understanding", "classification" mean training with vision-and-language understanding, and classification datasets besides detection data.

bone, ScaleDet demonstrates SOTA performance with high data-efficiency among all models. Among all competitors, the recent GLIPv2-B [46] is the only model performing on-par with ScaleDet (58.8), but using almost an order of magnitude more training data than ScaleDet (20.5M *vs.* 3.5M).

Overall, our results in Table 5 and Table 6 show that ScaleDet offers a data-efficient training recipe that can learn from fewer visual concepts and less training data, but still achieves state-of-the-art performance on COCO.

4.5. Comparison of SOTA on ODinW

We evaluate the transferablity on the "Object Detection in the Wild" (ODinW) benchmark and compare ScaleDet with SOTA pre-trained detectors that are capable of both direct and fine-tune transfer given any downstream detection datasets. Table 7 shows the comparison on ODinW among 3 types of detectors: (1) GLIP [26], GLIPv2 [46], (2) Detic [49], and (3) ScaleDet. ScaleDet is trained with the smallest training data, yet it achieves the best results in fine-tune transfer, surpassing even GLIPv2-B by 2.4 points (71.8 *vs.* 69.4). On direct transfer, GLIPv2-B is however stronger; we conjecture that this is thanks to its massive visual-language training data (20.5M), which likely covers the unseen concepts in the downstream datasets. Furthermore, when comparing ScaleDet-R, ScaleDet-B to Detic-R, Detic-B, we find the results of ScaleDet are significantly better, e.g., ScaleDet-R outperforms Detic-R by 10.0 points (39.4 *vs.* 29.4) on direct transfer and by 4.1 (68.5 *vs.* 64.4) on fine-tune transfer. These observations show the efficacy of ScaleDet in transferring to downstream datasets.

4.6. Ablation study

Finally, we ablate the components of our ScaleDet. As described in Sec. 3, ScaleDet is trained with two loss terms: the hard label assignment $\mathcal{L}_{hl}$ (Eq. (6)), and the soft label assignment $\mathcal{L}_{sl}$ (Eq. (7)). $\mathcal{L}_{hl}$ aims to assign each visual feature to one corresponding label, while $\mathcal{L}_{sl}$ works as a

Model	Datasets	L	C	O365	OID	mAP
$\mathcal{L}_{hl}$	L,C	33.1	44.6	15.7	43.2	34.2
$\mathcal{L}_{hl}+\mathcal{L}_{sl}$		33.3	44.9	15.9	43.7	**34.5**
$\mathcal{L}_{hl}$	L,C,O365,OID	36.7	46.7	30.8	69.1	45.8
$\mathcal{L}_{hl}+\mathcal{L}_{sl}$		36.8	47.1	30.6	69.4	**46.0**

(a) Ablation on upstream datasets.

Model	Datasets	ODinW	
	Metric	mAP	mAP_C
$\mathcal{L}_{hl}$	L,C	26.9	32.1
$\mathcal{L}_{hl}+\mathcal{L}_{sl}$		**26.9**	**32.5**
$\mathcal{L}_{hl}$	L,C,O365,OID	37.9	39.8
$\mathcal{L}_{hl}+\mathcal{L}_{sl}$		**39.4**	**40.9**

(b) Ablation on downstream benchmark ODinW by direct transfer.

Table 8. Ablation study on upstream datasets (a) and downstream datasets (b). mAP is the mean AP over datasets; mAP_C is the mean AP over all classes; better mAP of each setup is in **bold**.

regularizer to relate labels across datasets. Table 8a shows our ablation study on 4 upstream datasets and Table 8b shows our ablation of direct transfer on downstream benchmark ODinW. We find that training with two loss terms ($\mathcal{L}_{hl}+\mathcal{L}_{sl}$) leads to better overall results on both upstream and downstream datasets. For instance, in Table 8a, when training with two datasets (L+C), $\mathcal{L}_{hl}+\mathcal{L}_{sl}$ gives consistent better results on 4 upstream datasets compared to $\mathcal{L}_{hl}$. In Table 8b, when using $\mathcal{L}_{hl}+\mathcal{L}_{sl}$ to train on four training sets (L+C+O365+OID), the mAP is greatly improved by 1.5 points (39.4-37.9) and mAP_C is improved by 1.1 points (40.9-39.8), as compared to using only $\mathcal{L}_{sl}$. This suggests that $\mathcal{L}_{sl}$ is valuable to improve the model's generalizability. Overall, these results indicate that $\mathcal{L}_{hl}$ works effectively for label assignment within the upstream dataset, while $\mathcal{L}_{sl}$ takes a complementary role that is more evident when transferring to downstream datasets.

5. Conclusion and Future Work

We present ScaleDet, a simple yet scalable and effective multi-dataset training recipe for detection. ScaleDet learns across multiple datasets in a unified semantic label space, optimized by hard label and soft label assignments to align visual and text embeddings. ScaleDet achieves the new state-of-the-art performance on multiple upstream datasets (LVIS, COCO, Objects365, OpenImages) and downstream datasets (ODinW). As the key success of ScaleDet lies in learning in a unified semantic label space, our formulation can also generalize to other vision tasks such as image classification and semantic segmentation. We leave the unified design of a generic multi-dataset multi-task foundation model as promising and exciting future endeavors.

References

[1] Ankan Bansal, Karan Sikka, Gaurav Sharma, Rama Chellappa, and Ajay Divakaran. Zero-shot object detection. In *ECCV*, 2018. 5

[2] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*, 2015. 1

[3] Zhaowei Cai, Gukyeong Kwon, Avinash Ravichandran, Erhan Bas, Zhuowen Tu, Rahul Bhotika, and Stefano Soatto. X-detr: A versatile architecture for instance-wise vision-language tasks. In *ECCV*, 2022. 2

[4] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *CVPR*, 2018. 7

[5] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 7

[6] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017. 1

[7] 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 *CVPR*, 2021. 7

[8] Xiyang Dai, Yinpeng Chen, Bin Xiao, Dongdong Chen, Mengchen Liu, Lu Yuan, and Lei Zhang. Dynamic head: Unifying object detection heads with attentions. In *CVPR*, 2021. 7

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 1, 7

[10] Karan Desai and Justin Johnson. Virtex: Learning visual representations from textual annotations. In *CVPR*, 2021. 2

[11] Zheng Ding, Jieke Wang, and Zhuowen Tu. Open-vocabulary panoptic segmentation with maskclip. *arXiv preprint arXiv:2208.08984*, 2022. 2

[12] Golnaz Ghiasi, Yin Cui, Aravind Srinivas, Rui Qian, Tsung-Yi Lin, Ekin D Cubuk, Quoc V Le, and Barret Zoph. Simple copy-paste is a strong data augmentation method for instance segmentation. In *CVPR*, 2021. 5

[13] Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. Open-vocabulary object detection via vision and language knowledge distillation. In *ICLR*, 2021. 2

[14] Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *CVPR*, 2019. 2, 4, 5

[15] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017. 3, 7

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 5, 6

[17] Ronghang Hu and Amanpreet Singh. Unit: Multimodal multitask learning with a unified transformer. In *ICCV*, 2021. 7

[18] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. Openclip. *https://doi.org/10.5281/zenodo.5143773*, 2021. 3, 5

[19] 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 *ICML*, 2021. 2

[20] Aishwarya Kamath, Mannat Singh, Yann LeCun, Gabriel Synnaeve, Ishan Misra, and Nicolas Carion. Mdetr-modulated detection for end-to-end multi-modal understanding. In *ICCV*, 2021. 2, 7

[21] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[22] Ivan Krasin, Tom Duerig, Neil Alldrin, Vittorio Ferrari, Sami Abu-El-Haija, Alina Kuznetsova, Hassan Rom, Jasper Uijlings, Stefan Popov, Shahab Kamali, Matteo Malloci, Jordi Pont-Tuset, Andreas Veit, Serge Belongie, Victor Gomes, Abhinav Gupta, Chen Sun, Gal Chechik, David Cai, Zheyun Feng, Dhyanesh Narayanan, and Kevin Murphy. Openimages: A public dataset for large-scale multi-label and multi-class image classification. *Dataset available from https://storage.googleapis.com/openimages/web/index.html*, 2017. 1

[23] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *IJCV*, 2017. 2, 7

[24] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, et al. The open images dataset v4. *IJCV*, 2020. 1, 2, 4, 5

[25] John Lambert, Zhuang Liu, Ozan Sener, James Hays, and Vladlen Koltun. Mseg: A composite dataset for multi-domain semantic segmentation. In *CVPR*, 2020. 1, 2, 3

[26] Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, et al. Grounded language-image pre-training. In *CVPR*, 2022. 2, 5, 7, 8

[27] 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 *ECCV*, 2014. 2, 5

[28] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021. 5, 6

[29] Vicente Ordonez, Girish Kulkarni, and Tamara Berg. Im2text: Describing images using 1 million captioned photographs. In *NeurIPS*, 2011. 7

[30] Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *ICCV*, 2015. 2

[31] 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 *ICML*, 2021. 2, 3, 5

[32] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015. 3, 7

[33] Tal Ridnik, Emanuel Ben-Baruch, Asaf Noy, and Lihi Zelnik-Manor. Imagenet-21k pretraining for the masses. *arXiv preprint arXiv:2104.10972*, 2021. 5

[34] Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li, and Jian Sun. Objects365: A large-scale, high-quality dataset for object detection. In *ICCV*, 2019. 1, 2, 5

[35] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *ACL*, 2018. 7

[36] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *ICCV*, 2019. 3

[37] Jasper Uijlings, Thomas Mensink, and Vittorio Ferrari. The missing link: Finding label relations across datasets. In *ECCV*, 2022. 1, 2, 3

[38] Xudong Wang, Zhaowei Cai, Dashan Gao, and Nuno Vasconcelos. Towards universal object detection by domain attention. In *CVPR*, 2019. 1, 2

[39] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019. 6

[40] Hang Xu, Linpu Fang, Xiaodan Liang, Wenxiong Kang, and Zhenguo Li. Universal-rcnn: Universal object detector via transferable graph r-cnn. In *AAAI*, 2020. 1, 2

[41] Jiarui Xu, Shalini De Mello, Sifei Liu, Wonmin Byeon, Thomas Breuel, Jan Kautz, and Xiaolong Wang. Groupvit: Semantic segmentation emerges from text supervision. In *CVPR*, 2022. 2

[42] Jianwei Yang, Chunyuan Li, Pengchuan Zhang, Bin Xiao, Ce Liu, Lu Yuan, and Jianfeng Gao. Unified contrastive learning in image-text-label space. In *CVPR*, 2022. 2

[43] Lewei Yao, Runhui Huang, Lu Hou, Guansong Lu, Minzhe Niu, Hang Xu, Xiaodan Liang, Zhenguo Li, Xin Jiang, and Chunjing Xu. Filip: Fine-grained interactive language-image pre-training. In *ICLR*, 2021. 2

[44] Alireza Zareian, Kevin Dela Rosa, Derek Hao Hu, and Shih-Fu Chang. Open-vocabulary object detection using captions. In *CVPR*, 2021. 2, 5

[45] Xiaohua Zhai, Xiao Wang, Basil Mustafa, Andreas Steiner, Daniel Keysers, Alexander Kolesnikov, and Lucas Beyer. Lit: Zero-shot transfer with locked-image text tuning. In *CVPR*, 2022. 2

[46] Haotian* Zhang, Pengchuan* Zhang, Xiaowei Hu, Yen-Chun Chen, Liunian Harold Li, Xiyang Dai, Lijuan Wang, Lu Yuan, Jenq-Neng Hwang, and Jianfeng Gao. Glipv2: Unifying localization and vision-language understanding. In *NeurIPS*, 2022. 2, 7, 8

[47] Xiangyun Zhao, Samuel Schulter, Gaurav Sharma, Yi-Hsuan Tsai, Manmohan Chandraker, and Ying Wu. Object detection with a unified label space from multiple datasets. In *ECCV*, 2020. 1, 2

[48] Yiwu Zhong, Jianwei Yang, Pengchuan Zhang, Chunyuan Li, Noel Codella, Liunian Harold Li, Luowei Zhou, Xiyang Dai, Lu Yuan, Yin Li, et al. Regionclip: Region-based language-image pretraining. In *CVPR*, 2022. 2, 7

[49] Xingyi Zhou, Rohit Girdhar, Armand Joulin, Phillip Krähenbühl, and Ishan Misra. Detecting twenty-thousand classes using image-level supervision. In *ECCV*, 2022. 2, 5, 6, 7, 8

[50] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Probabilistic two-stage detection. *arXiv preprint arXiv:2103.07461*, 2021. 3, 5, 7

[51] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Simple multi-dataset detection. In *CVPR*, 2022. 1, 2, 3, 6, 7

[52] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019. 3, 7