Dataset Preparation for Arbitrary Object Detection: An Automatic Approach based on Web Information in English

Shucheng Li
National Key Lab for Novel Software Technology, Nanjing University
Nanjing, Jiangsu, China
shuchengli58@gmail.com

Boyu Chang
National Key Lab for Novel Software Technology, Nanjing University
Nanjing, Jiangsu, China
cby19680713@163.com

Bo Yang
National Key Lab for Novel Software Technology, Nanjing University
Nanjing, Jiangsu, China
1961882079@qq.com

Hao Wu
National Key Lab for Novel Software Technology, Nanjing University
Nanjing, Jiangsu, China
hao.wu@nju.edu.cn

Sheng Zhong
National Key Lab for Novel Software Technology, Nanjing University
Nanjing, Jiangsu, China
zhongsheng@nju.edu.cn

Fengyuan Xu*
National Key Lab for Novel Software Technology, Nanjing University
Nanjing, Jiangsu, China
fengyuan.xu@nju.edu.cn

ABSTRACT

Automatic dataset preparation can help users avoid labor-intensive and costly manual data annotations. The difficulty in preparing a high-quality dataset for object detection involves three key aspects: relevance, naturality, and balance, which are not addressed by existing works. In this paper, we leverage information from the web, and propose a fully-automatic dataset preparation mechanism without any human annotation, which can automatically prepare a high-quality training dataset for the detection task with English text terms describing target objects. It contains three key designs, i.e., keyword expansion, data de-noising, and data balancing. Our experiments demonstrate that the object detectors trained with auto-prepared data are comparable to those trained with benchmark datasets and outperform other baselines. We also demonstrate the effectiveness of our approach in several more challenging real-world object categories that are not included in the benchmark datasets.

CCS CONCEPTS

• **Information systems** → *Web mining*; • **Computing methodologies** → *Object detection.*

KEYWORDS

dataset preparation, web information retrieval, object detection

ACM Reference Format:
Shucheng Li, Boyu Chang, Bo Yang, Hao Wu, Sheng Zhong, and Fengyuan Xu. 2023. Dataset Preparation for Arbitrary Object Detection: An Automatic Approach based on Web Information in English. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '23), July 23–27, 2023, Taipei, Taiwan.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3539618.3591661

*Corresponding author.

SIGIR '23, July 23–27, 2023, Taipei, Taiwan

ACM ISBN 978-1-4503-9408-6/23/07…$15.00
https://doi.org/10.1145/3539618.3591661

1 INTRODUCTION

Preparing training datasets is usually labor-intensive and time-consuming, but crucial for deep learning models. For object detection, an important computer vision task, it is becoming more and more difficult to manually collect and annotate datasets to meet the demand of the community for a broader range of data categories and volumes. More concretely, provided an object like "dinosaur" which cannot be found in the benchmark datasets, is it possible to automatically prepare high-quality training datasets? In this paper, we explore the feasibility of automatically preparing an image dataset that can be used to train a (weakly-supervised) detection model. For dataset quality, we focus on three important metrics to model performance and generalization: the *relevance*, *naturality*, and *balance*. Given a target object, (i) the *relevance* means that an image should contain at least one target object; (ii) the *naturality* means that the distribution of the target object in the prepared dataset should be as close as possible to its real-world distribution, which is important to generalization; (iii) the *balance* means that the prepared dataset should avoid duplication of similar images.

Previous works mostly focus on the relevance metric, while ignoring the other two metrics. Specifically, works like [26] use data augmentation to reduce the impact of irrelevant noisy data, while other works employ re-ranking methods [8, 9] or image clustering [13, 39] to improve relevance of prepared datasets.

In this paper, we revisit a unique resource in the real world, the ***web***. With information retrieved from the web, previous works have tried to prepare data with prior knowledge [26] or augment data for existing datasets [31]. However, they do not unleash all the power of the web, which is the largest and most comprehensive knowledge and data resource in the real world. As shown in Fig. 1, we identify three key web information sources as follows,

- **Corpora**: the first source is the corpora edited and verified via crowdsourcing, such as English Wikipedia and BookCorpus [42], which can capture comprehensive correlations of almost any given term. This part of web information can be

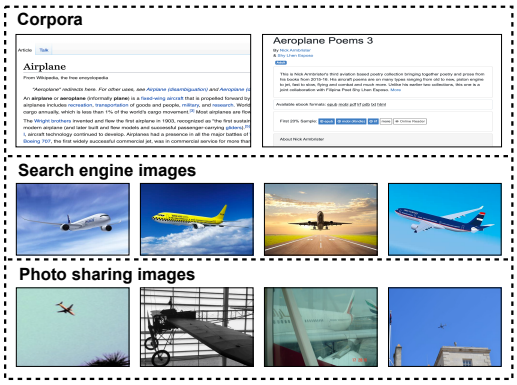


Figure 1: Examples of three information resources on the web for our automatic data preparation. *Top*: corpora including English Wikipedia. *Middle*: images from search engines. *Bottom*: images from photo sharing sites.

exploited via a large-scale pre-trained model like BERT [6], but it does not contain image data.

- **Search engine images**: the second source is the searchable images sophistically polished and posed by website owners for the purpose of easy understanding. These iconic or typical images [18] can be semantically matched via the image search engines like Bing or Google. However, this part of web data lack rich contextual information and various viewpoints [3, 18] with respect to a given object term.
- **Photo sharing images**: the last source is the massive diversified images taken by many people and directly posed to photo sharing sites like Flickr. These images are more natural and less biased than those from the search engines [7, 18, 32]. However, crawling this part of web images can easily include a lot of noisy or irrelevant images with respect to a given object term.

By bridging some technical gaps, it is feasible to integrally leverage their strengths to eliminate human involvement in data preparation.

Therefore, according to our insights above, we propose a *fully-automatic training dataset preparation mechanism for detection models of any objects*. It takes text terms describing any objects as inputs and auto-prepares a large training dataset for the detection task of these objects *from scratch* in the real world. Under the hood, our mechanism combines three web sources to form hybrid implicit supervision, which can be an alternative to explicit human supervision. We also address three challenges in the mechanism design so that we can bridge the technical gap in such web-based implicit supervision. **First**, we augment the human-oriented image search engines with a better ability to serve auto AI training. **Second**, we design an auto-pruning method of noisy samples, which are inevitable to photo sharing images, in preparing our datasets. **Last**, we propose an algorithm to auto-condense our prepared datasets, making the representative power of our picked images close to that of expert-picked ones.

To demonstrate the power of our mechanism, we build a tool for the scenario of arbitrary object detection. The input of this tool is text terms of any objects, and its output is a *ready-to-use detector of those objects*. This tool applies our mechanism to guide the data preparation and uses the Weakly Supervised Object Detection (WSOD) models [1, 15, 24, 29] as object detectors in the subsequent training. Typically, the WSOD models convert the detection task to a bag classification task and implicitly learn the selection of object proposals. Therefore, the image-level labels are enough (with no need for bounding box labels). The WSOD models could also handle cases where multiple instances from multiple categories exist in a single image. The tool above answers our feasibility question and provides the detection freedom of object categories.

Compared to manual approaches, our mechanism could save lots of time in image collection and annotation during the dataset construction. Moreover, compared to benchmark datasets released several years ago, it inherits the evolutionary nature of the web content, to easily adapt to essential changes of the same object, e.g., the changes in computer monitors from the 1990s to the 2010s, and timely cover newly emerging objects like the SpaceX Star Ship.

As to the training data quality, extensive experiments show that the WSOD detectors trained with datasets prepared by our auto mechanism are *comparable* to those trained with benchmark datasets like PASCAL VOC [7] or MS COCO [18] and *outperform* other baselines.

How does our mechanism behave in real use? Especially for those object categories not in the benchmarks. To answer this question, we choose several challenging object categories that could not be found in the existing benchmark datasets for evaluation. We prepare a training dataset with our mechanism for these categories, and the results of the trained WSOD detectors further demonstrate the generalization and the practicability of our mechanism.

In this work we make the following contributions:

- We design a fully-automatic training dataset preparation for arbitrary object detection. We propose and take into consideration three important metrics of prepared datasets with respect to the target object: relevance, naturality, and balance. These metrics ensure the overall quality of prepared datasets.
- We discover three unique opportunities of information resources on the web, which offer indirect hidden supervision for the purpose of data preparation. We propose a mechanism to build an alternative to explicit human supervision in preparing high-quality object-specific training datasets, which could be an inspiration for more works toward fully autonomous deep learning in the real world.
- Extensive experiments demonstrate that the quality of training datasets prepared by our mechanism is comparable to those benchmarks and outperforms baselines. Furthermore, we evaluate our mechanism on several challenging out-of-benchmark object categories and the results demonstrate its generalization.

2 APPROACH

In this section, we present the design intuition, pipeline procedure, and algorithmic modules of our auto training dataset preparation mechanism for weakly-supervised object detection models.

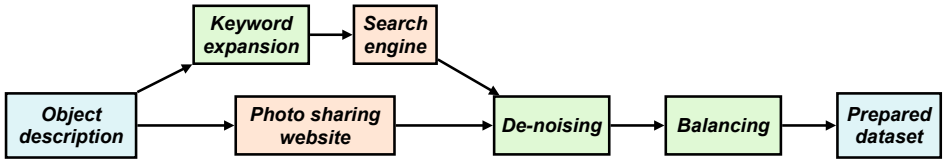


Figure 2: Overall workflow for our approach. Three green boxes are algorithmic modules. The corpora data resource is included in the keyword expansion module and not shown separately here.

Figure 3: Top-5 query results from Bing with keywords "dog" and "mountain dog". *Top:* search results with "dog", *Bottom:* search results with "mountain dog".

2.1 Design Intuition

Web photo sharing sites like Flickr are the data sources for popular benchmark datasets like PASCAL VOC or MS COCO because they can provide massive non-iconic and naturally-distributed image candidates [7, 18, 32]. However, those candidates have to be dedicatedly annotated by experts in order to ensure the dataset quality. When switching to the auto mode, we need some implicit supervision as an alternative for this job.

We discover that image search engines might be well suited for such a purpose if some technical gap can be bridged. For a given object description, as shown in Figure 3, image search engines like Bing or Google can provide accurate iconic images, especially in their top-ranked results [3]. Those images are manually picked and posted online to effectively convey information, containing hints of the hidden supervision in a canonical perspective. However, image search engines are human-oriented so that the users can learn a good appearance sketch of any object via various queries. Thus, we have to support two new functions so that it is feasible to build implicit supervision on top of the image search engines. One is to automatically query the engine like a human user to obtain iconic images of the target object in various semantic aspects, while the other is to extract the implicit hints in those samples to form a "reference" on how to filter out noisy samples from Flickr images like an expert.

Please note that we cannot directly prepare the training dataset from scratch only with the returned results of the image search engines. It is because most returned images are biased - often a single object instance centered in a clean background. They lack rich contextual information and various viewpoints (as shown in Figure 1 and Figure 3), which are critical to the model generalization.

2.2 Pipeline Procedure

Given the text name of the target object, the whole pipeline of our mechanism can automatically crawl, select, and refine the image data from the web, and finally outputs a dataset for training a weakly-supervised object detector for the target object.

(1) As shown in Figure 2, the pipeline first uses the keyword expansion module to collect a small set of ***"reference" images*** representing various aspects of the target object, under the hints provided by the large language model derived from the web corpora.
(2) Next, it crawls massive images from the photo sharing sites as the candidate data for the training dataset. And we design the de-noising module to prune out low-quality or even undesired candidate data with help from the "reference" data provided in the first step.
(3) As the last step, the pruned clean data is sent to the balancing module for further pruning with a condensation procedure concerning the image representative power. The output of this module is ready to be used in the weakly-supervised object detector training.

2.3 Algorithmic Modules

In the above pipeline procedure, there are three key modules that address three challenges of the hybrid implicit supervision we aim to provide through our auto preparation mechanism. They substantially improve the data quality of auto-prepared training data, which in turn delivers accurate training results. Therefore, supported by these three modules, the hybrid implicit supervision from the web could replace the human supervision efforts and give the freedom of picking an arbitrary object term as the target, as long as the knowledge and data of this object term are recorded and indexed online. We elaborate as follows on these three module designs and how they resolve challenges, respectively.

2.3.1 Module1: Keyword Expansion. In this module, in order to improve data relevance and naturality, we manage to make the implicit supervision from search engines more suitable for our data preparation mechanism.

Challenges. One would like to obtain detectors that should be generalizable. For example, when some person expresses a desire to obtain a detector about dogs, this person may expect that the obtained detector has the ability to detect different breeds of dogs and also dogs in different environments. While top images returned by the image search engines are semantically accurate, they are aforementioned often biased and even similar when the search term is as simple as an object name. The proposed methods should be able to expand the object term to various meaningful and natural terms at a more fine-grained level, in order to obtain a representative

image portfolio of the target object, which will be the important ***"reference"*** in the de-noising module.

Module design. The keyword expansion module seeks help from web corpora resources like the English Wikipedia or BookCorpus [42]. These corpora are organized in a collaborative learning way and contain natural language utterances of almost all object categories. This module exploits the knowledge in them through large language models pre-trained over these corpora. For the target object type, it first predicts the related context information (co-occurrence objects or background) and then predicts proper variants. In this way, our mechanism is able to get from the image search engines the relevant and diversified search results with respect to any target object, which could be used as the "reference" sample set in the next de-noising stage.

At the core of the keyword expansion module is the prompt learning technology. With the development of large pretrained language models in Natural Language Processing (NLP), such as ELMo [21], BERT [6], GPT [22], RoBERTa [20], etc., prompt learning [19] - as a new learning paradigm - has gained increasing attention recently. In prompt learning, we do not need to train a model between input $\boldsymbol{x}$ and output $\boldsymbol{y}$ like traditional supervised learning, or adapt the language model to downstream tasks by fine-tuning. Instead, with a large pretrained language model P, we can directly model the probability $P(\boldsymbol{x};\theta)$ of $\boldsymbol{x}$, then we can predict $\boldsymbol{y}$ with this probability. In this process, we do not have to prepare any extra training data manually.

Specifically, we design a cloze prompt for our keyword expansion task. We take the object category name like "dog" as the $\boldsymbol{x}$, and P as the pretrained language model. $\boldsymbol{x'}$ is the template prompt string with unfilled slots. For each unfilled slot in $\boldsymbol{x'}$, we employ the results with top-k probability predicted by P and get outputs $\hat{\boldsymbol{e}}$, then we fill $\hat{\boldsymbol{e}}$ into $\boldsymbol{x'}$ and get the expanded keywords $\boldsymbol{y}$. Formally,

$$\hat{\boldsymbol{e}} = \arg\max_{\boldsymbol{e}\in\mathcal{E}}(P(T(\boldsymbol{x'},\boldsymbol{e});\theta),k), \tag{1}$$

where $\boldsymbol{e}$ is values that can be filled in the slots and $\mathcal{E}$ is set of all possible values. T is the operation of filling $\boldsymbol{e}$ into the prompts $\boldsymbol{x'}$. θ stands for the parameters of the pretrained language model.

Moreover, to improve the naturality of searched image distribution, the expansion strategy in our template prompt $\boldsymbol{x'}$ considers two dimensions. We describe them in detail as follows,

- **Proper co-occurrence objects or background information of the target object.** We want to discover some natural co-occurrence objects or background information for the object term. Take the object name "dog" for instance. We design a prompt shaped like "dog `[PREPS]` `[MASK]`", where "`[MASK]`" denotes an unfilled slot, and "`[PREPS]`" stands for some common prepositions of orientation like "in", "on", "under", and "over", etc. Some expanded results are like "*dog in the park*", "*dog on the ground*", etc. More details for "`[PREPS]`" can be found in the experiments section.
- **Proper variants of the target object.** Also with the object name "dog" as an example, we expand to find some natural variants for this object term by adding an unfilled slot token before it, like "`[MASK]` dog". Some results are "*barking dog*", "*mountain dog*", etc.

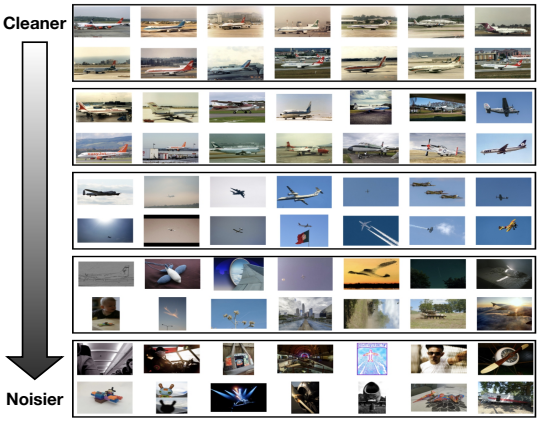


Figure 4: Our observation for photo sharing images. Clustering results of "aeroplane" images are employed as an example. *Top side:* clusters with higher intra-similarity. *Bottom side:* clusters with lower intra-similarity.

With the original search terms expanded following a natural distribution, we then exploit them to query the image search engines to obtain accurate, diversified, and natural returned images. These images could provide a "reference" to the next de-noising module for better dataset relevance and naturality.

2.3.2 Module2: De-noising. In this module, we utilize a clustering-based de-noising method with implicit supervision from the aforementioned "reference" data, aiming to improve the relevance of our candidate dataset of the photo sharing images.

Challenges. Photo sharing sites like Flickr provide massive non-iconic images uploaded by various amateur photographers. The image sampling of an object on them, although noisy by nature, approximately represents the natural distribution of their occurrences in the real world [7, 18]. The popular benchmarks [7, 18] use those data with experts manually improving the data quality. When removing the human efforts, we design an algorithm based on the observation below, in order to prune out noisy or low-quality images.

Observation: ***due to the randomness and dispersion of noisy data in the real world, for clustering results of photo sharing images, cleaner clusters tend to have higher intra-cluster similarity, while noisier clusters have the opposite.*** As illustrated in Figure 4, it could be found that clusters with higher intra-similarity tend to be cleaner, while clusters with lower intra-similarity tend to contain more noisy samples. This observation is important for our design of the de-noising module.

Module design. In a nutshell, we first crawl from the photo sharing site a large number of images tagged with the target object term. According to the observation above, we then group images into different clusters and calculate a score for each image. This score consists of two parts, (i) an intra-cluster similarity measure, and (ii) a similarity measure to the aforementioned "reference" set. In the end, we sort candidate images according to their scores and pick the top ones to obtain a relatively clean dataset without manual effort.

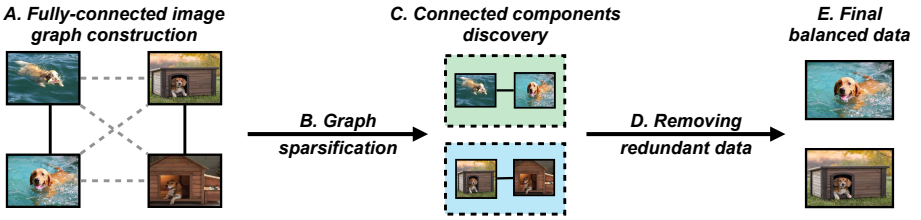


Figure 5: Illustration for our balancing module. The dash lines in the fully-connected graph represent masked edges by ϵ. And connected components covered with color are detected as redundant.

In the following, we explain how the de-noising module extract features and compute this image score.

First, given a photo sharing image $I_i \in \mathcal{I}$, we get its feature F_i, through a pretrained CNN backbone. For the feature of a "reference" image, i.e. $I_i \in \mathcal{I}_{ref}$, we denote its feature as $F_{\text{ref},i}$.

With the extracted features, we then can calculate the two parts of similarity measures for each image. For a cluster C_t, one of the clusters provided by our clustering algorithm in the de-noising module, we can compute the intra-cluster similarity of C_t as

$$S^t_{\text{intra}} = \frac{1}{|C_t|^2} \sum_{i=1}^{|C_t|} \sum_{j=1}^{|C_t|} f_{\text{sim}}(F_i, F_j), \tag{2}$$

where $|C_t|$ is the image number of C_t, and f_{sim} is the cosine similarity function. And for $I_i \in C_t$, $S_{\text{intra},i} = S^t_{\text{intra}}$.

We also compute the similarity measure of I_i to the "reference" image set, $S_{\text{ref},i}$, as

$$S_{\text{ref},i} = \frac{1}{|\mathcal{I}_{\text{ref}}|} \sum_{j=1}^{|\mathcal{I}_{\text{ref}}|} f_{\text{sim}}(F_i, F_{\text{ref},j}), \tag{3}$$

where $|\mathcal{I}_{\text{ref}}|$ denotes the number of the "reference" image set.

Then the final similarity score of a photo sharing image I_i, $S_{\text{final},i}$, is computed as

$$S_{\text{final},i} = \alpha \cdot S_{\text{intra},i} + (1-\alpha) \cdot S_{\text{ref},i}, \tag{4}$$

where α is a trade-off coefficient. The de-noising module delivers the image I_i to the balancing module if its similarity score is above an empirical threshold β, i.e. $S_{\text{final},i} \geq \beta$.

2.3.3 Module3: Balancing. In this module, we employ a balancing metric to improve the representative power of the prepared dataset and design a graph-based sparsification algorithm for such a purpose. We illustrate this module in Figure 5.

Challenges. Balance is one of the most important criteria to determine the quality of a dataset. Unbalance or redundancy in the dataset will affect both the performance and efficiency of corresponding model training. However, it is challenging to balance the dataset without expert assistance. We need to design an automatic method to identify those redundant data and selectively remove them from the prepared dataset.

Module design. We design a balancing module that can condense the size of the prepared dataset with the improved uniformity of the sample distribution, achieving the balancing objective. The strategy in our data balancing shares a similar principle with those contrastive learning methods [10, 14, 36] which are for extracting high-quality feature representations. Our strategy is to properly remove redundant images so that the images remained are pushed further apart in a balancing score perspective.

Specifically, we first project all images refined by the de-noising module onto the surface of a unit hypersphere which generally models the sampling distribution of the target object. Based upon this modeling, we can then design an overall balance score function of the prepared dataset,

$$S_{\text{balance}}(\mathcal{I}) \triangleq e^{-\left[\sum_{I_i \in \mathcal{I}} \sum_{I_j \in \mathcal{I}} \|N_i - N_j\|_2^2\right]}, \tag{5}$$

which is modified from the uniformity metric in [36]. In $S_{\text{balance}}(\mathcal{I})$, for each image $I_i \in \mathcal{I}$, N_i is the feature normalized from F_i. S_{balance} measures how uniform all image features are on the surface of a unit hypersphere. And the smaller S_{balance} is, the better balance our prepared dataset could maintain.

To calculate $S_{\text{balance}}(\mathcal{I})$, we construct a distance based fully-connected graph G_{full}. On this graph, a node i stands for image i in the dataset, and its node feature is N_i; each edge feature $\boldsymbol{e}_{i,j}$ is calculated by $\boldsymbol{e}_{i,j} = e^{-\|N_i - N_j\|_2^2}$. In this way, the balance score of $\mathcal{I}$ can be transformed into

$$S_{\text{balance}}(\mathcal{I}) = \frac{1}{|\mathcal{E}|} \sum_{\boldsymbol{e}_{i,j} \in \mathcal{E}} \boldsymbol{e}_{i,j}, \tag{6}$$

where $\mathcal{E}$ is the edge set of G_{full}, and $\mathcal{I}$ is the image set used to construct G_{full}.

In the next, we provide a graph based sparsification method to execute our data balancing strategy, while ensuring a reasonable dataset size. We leverage an epsilon-based sparsification method to indirectly control the budget of the removed node/image number. Its objective is to optimize

$$\arg\min_{\epsilon} (S_{\text{balance}}(\mathcal{I}_{\text{sp}}) + \frac{\lambda \cdot |\mathcal{I}|}{|\mathcal{I}_{\text{sp}}|}), \tag{7}$$

where $\mathcal{I}_{\text{sp}}$ is the image set after sparsification. ϵ is the controlling threshold. λ is a scaling coefficient.

Specifically, as shown in Figure 5, the balancing module masks off (remove) the edges in $\mathcal{E}$ smaller than ϵ, which is calculated according to Equation (7). It then discovers the connected components in the new graph and considers that redundancy exists inside a component (refer to examples marked in color in Figure 5). It then collapses each connected component into a single node, by only keeping nodes/images with the highest $S_{\text{final},i}$, to derive the final balanced image set $\mathcal{I}_{\text{sp}}$.

3 EXPERIMENTS

In this section, to evaluate our approach, we build an object detector generation tool, in which the core part is our auto training data preparation mechanism. This tool takes as input text terms describing objects and automatically returns a ready-to-use object detector as the output. No human efforts like data collection and annotation are required. Thus, we can evaluate our approach by evaluating the performance and generalization of the object detector generated by this tool.

3.1 Experiment Setup

Settings of our mechanism. For the pretrained language model in our keyword expansion module, we use RoBERTa [20], a popular pretrained language model. For the orientation preposition tokens "`[PREPS]`" in our cloze prompt, we use {*in, on, under, over, behind, before, inside, outside, near*}. We employ Bing as the image search engine to crawl images as the "reference" in the de-noising module. For each object category, the search engine image number is set to 300. The probability of each expanded keyword determines the proportion of the collected images from the image search engine. We denote datasets auto-prepared with our mechanism according to VOC and COCO categories as **Ours-VOC** and **Ours-COCO**, respectively. We crawl 2,000 images for each category from Flickr by using the category names as the search keywords. Overlapped images with benchmark datasets are removed from our datasets prepared. Considering privacy and copyright, we only collect public data with proper licenses. In the end, we get 78,988 images for Ours-VOC and 155,728 images for Ours-COCO.

Weakly supervised object detection model. For the weakly supervised object detection (WSOD) model used in our tool, we employ WSDDN [1], an end-to-end CNN-based WSOD model. WSDDN first employs object proposal generation algorithms like EdgeBox [43] to generate proposals for each image. Then WSDDN transforms the object detection task into a classification task, where object proposals in an image are regarded as a bag and WSDDN learns for this bag classification task. In this process, WSDDN could implicitly learn the selection of object proposals to accomplish the detection task. Moreover, WSDDN contains a spatial regularizer that makes spatially highly-overlapped object proposals share similar features. With prepared high-quality training datasets, WSDDN could handle cases where multiple instances from multiple object categories exist.

Hyperparameter. In our experiments, the learning rate, weight decay, and batch size are $1e^{-5}$, $5e^{-4}$, and 1, respectively. We use Adam [16] as the optimizer and EdgeBox [43] as the object proposal generation algorithm. For an image, the maximum number of object proposals used is 2,000. K-means is employed as our clustering algorithm and K is set to 10. We use VGG16 [27] with weights pre-trained on ImageNet [5] as the backbone of the WSDDN. We set the regularizer coefficient in WSDDN to $1e^{-3}$. In the de-noising module, α is set to 0.5, and β is set to 0.7. In the balancing module, λ is empirically set to 0.1. The threshold of the NMS module used in the test is set to 0.4.

Hardware information. We run our experiments on a single PC with four NVIDIA TITAN Xp GPUs. WSDDN[1] is implemented with PyTorch 1.10.0. The OS we used is Ubuntu 16.04.4 LTS, and the version of CUDA is 10.2.

Evaluation metric. When evaluating in the test sets of benchmarks, for VOC, we employ Average Precision (AP) and mean Average Precision (*m*AP) as evaluation metrics, with a standard 50% Intersection-over-Union (IoU). For COCO, we employ the standard COCO evaluation metrics AP and *m*AP of different IoU thresholds.

3.2 Comparison with Benchmarks

To show the data preparation quality, we compare our auto-prepared datasets with train sets of benchmark datasets: PASCAL VOC2007, VOC2012 [7], and MS COCO [18], which are manually prepared. For VOC2007, as shown in Table 1, the performance of the detector trained with Ours-VOC (32.2%) is comparable to the detector trained with VOC07 trainval split (32.5%). Furthermore, for VOC2012 evaluation results in Table 2, it could be found that object detectors trained on Ours-VOC (29.2%) and VOC12 trainval (29.4%) performed comparably. For COCO, as shown in Table 3, the performance difference of $m\text{AP}_{50}$ between Ours-COCO and COCO train2017 (13.5% vs. 13.8%) is also small. These results demonstrate the effectiveness of our mechanism and the overall high quality of the auto-prepared dataset.

Among these results, we notice that in Table 1 the AP_{50} of "*TV/Monitor*" of Ours-VOC is significantly lower than the VOC trainval split (29.5% vs. 46.1%). After an inspection at the data level, we find that the main reason is that in the VOC07 test split, "*TV/Monitor*" images are old TVs and monitors produced before 2007, which have changed a lot by the time we collect the data. This illustrates that our approach has inherited the evolutionary nature of the web and allows for a wider range of applications. In later experiments, we also prove that the detector trained from our mechanism can work well to detect "*old monitor*" and "*monitor*".

3.3 Comparison with Baselines

We compare Ours-VOC and Ours-COCO with three baselines: (i) Flickr-VOC [26]: containing raw images directly retrieved from Flickr, with category names of PASCAL VOC as queries, the first 4,000 search results are kept for each category. It contains 83,905 images in total; (ii) Flickr-COCO [26]: similar to Flickr-VOC, with category names of MS COCO as queries and contains 335,327 images in total; (iii) Flickr-clean [37]: containing 41,625 images in total, it is constructed from Flickr with PASCAL VOC categories, then a salient object detection method [35] and a saliency-cut segmentation method in [4] are employed to remove noisy images and keep relatively simple images.

As shown in Table 1, when evaluated in the VOC07 test split, Ours-VOC (32.2%) outperforms Flickr-VOC (27.6%) and Flickr-clean (28.8%). For evaluation results in the VOC12 test split in Table 2, Ours-VOC (29.2%) also outperforms Flickr-VOC (24.1%) and Flickr-clean (24.8%). And for COCO, as shown in Table 3, Ours-COCO (13.5%) outperforms Flickr-COCO (7.0%) in terms of $m\text{AP}_{50}$. These results further demonstrate the effectiveness of our mechanism.

[1]We run our own implementation of WSDDN in PyTorch in our experiments.

Table 1: Comparison with PASCAL VOC2007 and other baselines. For model training on all datasets, only image-level annotations are used. AP_{50} (%) is used as evaluation metric. VOC 2007 *test* split is used for evaluation.

Datasets	Aero	Bike	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow	Table	Dog	Horse	Motor	Person	Plant	Sheep	Sofa	Train	TV	Average
Comparing with PASCAL VOC 2007																					
VOC07 trainval	40.1	39.2	26.2	20.7	**12.0**	**54.7**	42.5	37.7	2.3	**35.0**	**32.7**	38.5	**41.0**	**50.8**	13.6	**12.8**	29.9	34.9	39.3	**46.1**	**32.5**
Comparing with other baselines																					
Flickr-VOC	35.8	**39.5**	**35.8**	9.6	10.0	51.5	39.5	**41.3**	7.1	22.4	7.4	31.0	33.4	47.3	13.0	9.2	**32.7**	27.5	**44.6**	14.2	27.6
Flickr-clean	42.7	29.5	21.4	23.8	9.4	45.9	**44.5**	39.9	**9.7**	23.3	24.0	**41.1**	39.3	40.2	15.9	10.8	23.3	33.3	37.1	21.0	28.8
Ours-VOC	**43.5**	36.4	27.2	**27.8**	10.8	53.4	43.7	40.1	8.8	33.4	25.5	35.8	38.0	44.3	**15.9**	12.3	28.5	**45.6**	43.5	29.5	32.2

Table 2: Comparison with PASCAL VOC2012 and other baselines. For model training on all datasets, only image-level annotations are used. AP_{50} (%) is used as evaluation metric. VOC 2012 *test* split is used for evaluation.

Datasets	Average
VOC12 trainval	**29.4**
Flickr-VOC	24.1
Flickr-clean	24.8
Ours-VOC	29.2

Table 3: Comparison with MS COCO. For model training on all datasets, only image-level annotations are used. *m*AP (%) with different IoUs is used as the evaluation metric. COCO *val2017* split is used for evaluation.

Datasets	IoU=0.50	IoU=[0.50:0.95]	IoU=0.75
COCO train2017	**13.8**	**5.7**	**3.8**
Flickr-COCO	7.0	3.1	2.3
Ours-COCO	13.5	5.5	**3.8**

3.4 Qualitative Results

Here, we show the intermediate results of each module in our mechanism as follows, to help understand how the whole mechanism works.

- For the keyword expansion module, we show the keyword expansion results of category "*dog*" in Figure 6a. (i) For co-occurrence objects or background information: we could find that this module could make some natural and valuable discoveries, such as the "*park*", "*ground*", "*floor*", etc. And it could also find some less obvious but equally reasonable co-occurrence objects or backgrounds like "*leash*", "*sidewalk*", "*corner*", etc. These expansions are valuable for our mechanism from the perspective of comprehensiveness and robustness. (ii) For variants of "*dog*": this module also gives some common and natural variants like "*barking dog*" and "*mountain dog*". It also outputs some other inspiring and reasonable variants including "*mad dog*", "*guide dog*", etc.
- For the de-noising module, we show the clean and noisy outputs of "*aeroplane*" in Figure 6b. For noisy samples in the bottom, we could find that they actually could be divided into several different types: (i) the image contains only part of target objects; (ii) the image describes another object type; (iii) the object type is correct in the image, but not real-world objects. These typical noisy samples filtered out by our de-noising module further demonstrate the effectiveness of our mechanism.
- In Figure 6c, we show the redundant connected components in the image graph for "*aeroplane*" in the balancing module. We could find that images in a same color box have a high degree of similarity, and this module could condense them to improve the representative power of the prepared dataset.

Table 4: Ablation experiment results of our tool. We use VOC 2007 *test* split for evaluation, with mAP$_{50}$ (%) as the evaluation metric.

Datasets	mAP$_{50}$
Variant-A	30.4
Variant-B	28.3
Variant-C	31.0
Variant-D	29.7
Ours-VOC	**32.2**

Table 5: Results of our tool on out-of-benchmark object categories. We use AP_{50} (%) as the evaluation metric.

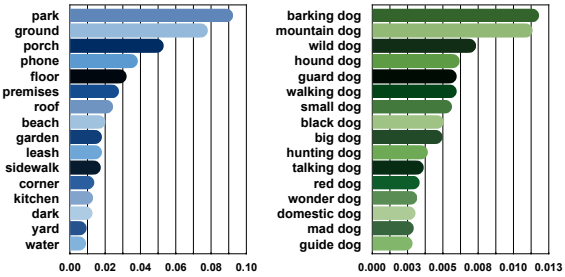
Categories	AP_{50}
Old monitor	52.8
Monitor	53.7
Dinosaur	24.0
Starship	32.5
Wall-E	46.3
Average	41.9

3.5 Ablation Study

We mainly conduct ablation experiments for variants of Ours-VOC as follows: (i) Variant-A: we remove our keyword expansion module; (ii) Variant-B: we remove our de-noising module; (iii) Variant-C: we remove our balancing module; (iv) Variant-D: we replace our de-noising module with the mixup de-noising method in [26]. As illustrated in Table 4, removing the keyword expansion, de-noising, or balancing module in our approach will cause a performance drop, especially the de-noising module, which causes a 4% drop of mAP$_{50}$. And if we compare the Variant-D (29.7%) and Ours-VOC (32.2%), it could be found that our de-noising module performs better than the mixup method. These results demonstrate the superiority and necessity of the modules in our approach.

3.6 Detection of Out-of-benchmark Objects

To prove the effectiveness of our mechanism in real scenarios, we choose some challenging object categories out-of-benchmark, i.e., *Old monitor*, *Monitor*, *Dinosaur*, *SpaceX starship*, and *Robot Wall-E*.



(a) Example keyword expansion results for "dog" with prediction probability. *Left*: related context information (co-occurrence objects or background) for "dog". *Right*: variants of "dog".

(b) De-noising module outputs for "aeroplane". *Top*: clean results. *Bottom*: noisy results.

(c) Balancing module outputs for "aeroplane". Images in a same color box are seen as redundant.

Figure 6: Qualitative results for three modules in our mechanism.

The selected categories include ancient objects (*Dinosaur*), objects that appear in the recent one or two years (*SpaceX starship*), virtual objects from films (*Robot wall-E*), and variations of the same kind of objects in different eras (from *Old monitor* to *Monitor*). Since no existing datasets are available for evaluation, for each category here, we crawled 300 images from Bing and 1,500 images from Flickr. And we manually annotated 200 Flickr images as the test split (the rest for trainval). The settings or hyperparameters have been introduced in the experiment setup. Detection results visualization and performance are shown in Figure 7 and Table 5, respectively. These results demonstrate the effectiveness and adaptability of our approach in practical usage.

3.7 Search Engine Impact

We have explained the purpose of introducing search engine images and their bias in the design intuition (Section 2.1). For experiments here, we aim to explore the search engine impact by comparing the performance of two common image search engines, Bing and Google, through experiments on our out-of-benchmark data. As shown in Figure 8, the $m\text{AP}_{50}$ values of Bing and Google are close to each other. This proves that our mechanism is not restricted to a particular image search engine.

In addition, though the search engine images are more accurate compared to those photo sharing images, they may also contain some fake images or human bias. We explain how this problem is solved in our mechanism. First, as mentioned in Section 2.1, we do not use images from search engines directly but as a reference. Second, we choose top-ranked images with better data quality in the searched results as the reference images. At last in Equation (4), in addition to the reference similarity score $S_{\text{ref},i}$, we also consider the intra-cluster similarity score $S_{\text{intra,i}}$, which is not affected by biased or fake images from search engines. And we use the whole reference image set for the calculation of $S_{\text{ref},i}$ to control the impact of a few outliers. In this way, we could solve the problem of biased or fake images from image search engines.

3.8 Sensitivity Analysis

We conduct an analysis of two important parameters in our mechanism: α and λ. α is mainly used in the de-noising module for the trade-off between $S_{\text{intra},i}$ and $S_{\text{ref},i}$ in Equation (4), and λ is a scaling coefficient to adjust the effect of the number of images in the balancing module.

As shown in Figure 9, for α, combining $S_{\text{intra},i}$ and $S_{\text{ref},i}$ is beneficial to improve the overall performance. Using either of them alone (when α is set to 0 or 1) will cause a performance drop. For λ, when it is extremely low ($1e^{-5}$), the redundant connected components will be very large, making the vast majority of images discarded, therefore the mAP is close to 0. And vice versa, when λ is too large (0.5), most images have remained and the redundancy of the dataset will increase, which also causes a performance drop. These results illustrate that it is helpful to choose proper values for α and λ during practical use.

4 RELATED WORK

This paper is mainly related to web-based automated dataset preparation and weakly supervised object detection.

Webly automated dataset preparation. In recent years, larger datasets and benchmarks [7, 17, 18, 25] have been released for better evaluation, more powerful models, and broader applications. These datasets are constructed with a web-based image collection in conjunction with labor-intensive manual annotations. In comparison, webly automated dataset preparation could be more efficient, large-scale, and cost-friendly. Taking the object detection task as an example, images of arbitrary object categories can be collected from the web and used for dataset preparation automatically. Existing works mainly focus on how to reduce the impact of noisy data and can be mainly divided into two parts: (i) general web-based datasets for classification, object recognition, etc. (ii) web-based datasets for object detection. For (i), works like [8, 9] use re-ranking methods to handle noisy images. In contrast to directly determining whether an image is noisy or not, works like [13, 39] employ clustering to group similar images first and then handle noisy images. For (ii), [26] directly crawl images from the web with object category names as keywords. And then a mixup data augmentation method is exploited to reduce the impact of images not containing target object instances.

Figure 7: Detection results visualization for object categories out-of-benchmark.

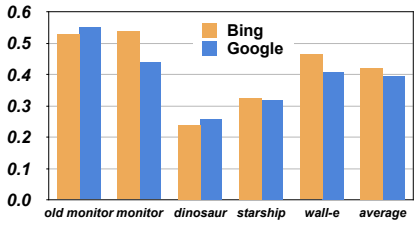


Figure 8: Image search engine comparison of Bing and Google with our out-of-benchmark data. AP_{50} is used as the evaluation metric.

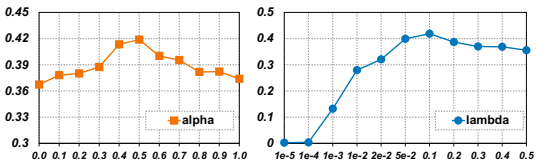


Figure 9: Parameter sensitivity analysis of α and λ on our out-of-benchmark data. mAP_{50} is used as the evaluation metric.

Weakly supervised object detection. Traditionally, object detection models [2, 11, 12, 23] require training images with bounding boxes for the supervised signal of localization. However, it is extremely costly or labor-intensive to get bounding box annotations for large-scale datasets. Therefore, Weakly Supervised Object Detection (WSOD) methods [1, 15, 24, 29, 30, 34, 38, 41] are proposed where training data with image-level category labels are sufficient. These works mostly abstract the object detection task as a multi-instance learning task. Specifically, WSOD methods first generate object proposals for each image with some traditional object proposal generation algorithms like SelectiveSearch [33] or EdgeBox [43]. Then, they take the object proposal set in each image as a bag and transform the object detection task into a bag classification task. In such a classification task, WSOD methods could implicitly learn the selection for the correct proposals.

Among these works, WSDDN [1] is a CNN-based end-to-end learning architecture for WSOD, which employs a spatial regularizer to make spatially highly-overlapped object proposals share similar features. Based on this work, works such as [28, 29] add instance classifiers to make highly-overlapped object proposals share similar label information. And then [24] uses a more sophisticated self-training method and a modified drop block method to solve the part domination problem. Besides, other methods also obtain improvements by leveraging the attention mechanism [15], object instance mining [40], etc.

5 CONCLUSION

In this work, we design a fully-automatic training dataset preparation for object detection with web resources. The preparation process takes into account the relevance, naturality, and balance of prepared datasets. The object detectors trained with our prepared datasets outperform baselines and have comparable performance to those trained with public benchmarks. With our auto preparation mechanism, the object detection models can be set free from limited object categories in the public benchmarks, accelerating their applications in practice.

ACKNOWLEDGEMENTS

This work was supported in part by the National Key R&D Program of China under Grants 2022YFF0604503, 2021YFB3100300, and 2020YFB1005900, in part by NSFC under Grants 62272224 and 62272215, in part by the Leading edge Technology Program of Jiangsu Natural Science Foundation under Grant BK20202001, and in part by the Science Foundation for Youths of Jiangsu Province under Grant BK20220772.

REFERENCES

[1] Hakan Bilen and Andrea Vedaldi. 2016. Weakly Supervised Deep Detection Networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. IEEE Computer Society, 2846–2854. https://doi.org/10.1109/CVPR.2016.311

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. 2020. End-to-End Object Detection with Transformers. In *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part I (Lecture Notes in Computer Science, Vol. 12346)*, Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (Eds.). Springer, 213–229. https://doi.org/10.1007/978-3-030-58452-8_13

[3] Xinlei Chen and Abhinav Gupta. 2015. Webly Supervised Learning of Convolutional Networks. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*. IEEE Computer Society, 1431–1439. https://doi.org/10.1109/ICCV.2015.168

[4] Ming-Ming Cheng, Niloy J. Mitra, Xiaolei Huang, Philip H. S. Torr, and Shi-Min Hu. 2015. Global Contrast Based Salient Region Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* 37, 3 (2015), 569–582. https://doi.org/10.1109/TPAMI.2014.2345401

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*. IEEE Computer Society, 248–255. https://doi.org/10.1109/CVPR.2009.5206848

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, Jill Burstein, Christy Doran, and Thamar Solorio (Eds.). Association for Computational Linguistics, 4171–4186. https://doi.org/10.18653/v1/n19-1423

[7] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John M. Winn, and Andrew Zisserman. 2010. The Pascal Visual Object Classes (VOC) Challenge. *Int. J. Comput. Vis.* 88, 2 (2010), 303–338. https://doi.org/10.1007/s11263-009-0275-4

[8] Robert Fergus, Li Fei-Fei, Pietro Perona, and Andrew Zisserman. 2005. Learning Object Categories from Google's Image Search. In *10th IEEE International Conference on Computer Vision (ICCV 2005), 17-20 October 2005, Beijing, China*. IEEE Computer Society, 1816–1823. https://doi.org/10.1109/ICCV.2005.142

[9] Robert Fergus, Pietro Perona, and Andrew Zisserman. 2004. A Visual Category Filter for Google Images. In *Computer Vision - ECCV 2004, 8th European Conference on Computer Vision, Prague, Czech Republic, May 11-14, 2004. Proceedings, Part I (Lecture Notes in Computer Science, Vol. 3021)*, Tomás Pajdla and Jiri Matas (Eds.). Springer, 242–256. https://doi.org/10.1007/978-3-540-24670-1_19

[10] Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple Contrastive Learning of Sentence Embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (Eds.). Association for Computational Linguistics, 6894–6910. https://doi.org/10.18653/v1/2021.emnlp-main.552

[11] Ross B. Girshick. 2015. Fast R-CNN. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*. IEEE Computer Society, 1440–1448. https://doi.org/10.1109/ICCV.2015.169

[12] Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. 2014. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*. IEEE Computer Society, 580–587. https://doi.org/10.1109/CVPR.2014.81

[13] Sheng Guo, Weilin Huang, Haozhi Zhang, Chenfan Zhuang, Dengke Dong, Matthew R. Scott, and Dinglong Huang. 2018. CurriculumNet: Weakly Supervised Learning from Large-Scale Web Images. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part X (Lecture Notes in Computer Science, Vol. 11214)*, Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss (Eds.). Springer, 139–154. https://doi.org/10.1007/978-3-030-01249-6_9

[14] Raia Hadsell, Sumit Chopra, and Yann LeCun. 2006. Dimensionality Reduction by Learning an Invariant Mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2006), 17-22 June 2006, New York, NY, USA*. IEEE Computer Society, 1735–1742. https://doi.org/10.1109/CVPR.2006.100

[15] Zeyi Huang, Yang Zou, B. V. K. Vijaya Kumar, and Dong Huang. 2020. Comprehensive Attention Self-Distillation for Weakly-Supervised Object Detection. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (Eds.). https://proceedings.neurips.cc/paper/2020/hash/c3535febaff29fcb7c0d20cbe94391c7-Abstract.html

[16] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.). http://arxiv.org/abs/1412.6980

[17] Ivan Krasin, Tom Duerig, Neil Alldrin, Vittorio Ferrari, Sami Abu-El-Haija, Alina Kuznetsova, Hassan Rom, Jasper Uijlings, Stefan Popov, Andreas Veit, et al. 2017. Openimages: A public dataset for large-scale multi-label and multi-class image classification. *Dataset available from https://github. com/openimages* 2, 3 (2017), 18.

[18] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: Common Objects in Context. In *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V (Lecture Notes in Computer Science, Vol. 8693)*, David J. Fleet, Tomás Pajdla, Bernt Schiele, and Tinne Tuytelaars (Eds.). Springer, 740–755. https://doi.org/10.1007/978-3-319-10602-1_48

[19] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing. *CoRR* abs/2107.13586 (2021). arXiv:2107.13586 https://arxiv.org/abs/2107.13586

[20] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *CoRR* abs/1907.11692 (2019). arXiv:1907.11692 http://arxiv.org/abs/1907.11692

[21] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, Marilyn A. Walker, Heng Ji, and Amanda Stent (Eds.). Association for Computational Linguistics, 2227–2237. https://doi.org/10.18653/v1/n18-1202

[22] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. (2018).

[23] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. 2017. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* 39, 6 (2017), 1137–1149. https://doi.org/10.1109/TPAMI.2016.2577031

[24] Zhongzheng Ren, Zhiding Yu, Xiaodong Yang, Ming-Yu Liu, Yong Jae Lee, Alexander G. Schwing, and Jan Kautz. 2020. Instance-Aware, Context-Focused, and Memory-Efficient Weakly Supervised Object Detection. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*. Computer Vision Foundation / IEEE, 10595–10604. https://doi.org/10.1109/CVPR42600.2020.01061

[25] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Li Fei-Fei. 2015. ImageNet Large Scale Visual Recognition Challenge. *Int. J. Comput. Vis.* 115, 3 (2015), 211–252. https://doi.org/10.1007/s11263-015-0816-y

[26] Yunhang Shen, Rongrong Ji, Zhiwei Chen, Xiaopeng Hong, Feng Zheng, Jianzhuang Liu, Mingliang Xu, and Qi Tian. 2020. Noise-Aware Fully Webly Supervised Object Detection. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*. Computer Vision Foundation / IEEE, 11323–11332. https://doi.org/10.1109/CVPR42600.2020.01134

[27] Karen Simonyan and Andrew Zisserman. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.). http://arxiv.org/abs/1409.1556

[28] Peng Tang, Xinggang Wang, Song Bai, Wei Shen, Xiang Bai, Wenyu Liu, and Alan L. Yuille. 2020. PCL: Proposal Cluster Learning for Weakly Supervised Object Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* 42, 1 (2020), 176–191. https://doi.org/10.1109/TPAMI.2018.2876304

[29] Peng Tang, Xinggang Wang, Xiang Bai, and Wenyu Liu. 2017. Multiple Instance Detection Network with Online Instance Classifier Refinement. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*. IEEE Computer Society, 3059–3067. https://doi.org/10.1109/CVPR.2017.326

[30] Peng Tang, Xinggang Wang, Angtian Wang, Yongluan Yan, Wenyu Liu, Junzhou Huang, and Alan L. Yuille. 2018. Weakly Supervised Region Proposal Network and Object Detection. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XI (Lecture Notes in Computer Science, Vol. 11215)*, Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss (Eds.). Springer, 370–386. https://doi.org/10.1007/978-3-030-01252-6_22

[31] Qingyi Tao, Hao Yang, and Jianfei Cai. 2019. Exploiting Web Images for Weakly Supervised Object Detection. *IEEE Trans. Multim.* 21, 5 (2019), 1135–1146. https://doi.org/10.1109/TMM.2018.2875597

[32] Antonio Torralba and Alexei A. Efros. 2011. Unbiased look at dataset bias. In *The 24th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2011, Colorado Springs, CO, USA, 20-25 June 2011*. IEEE Computer Society, 1521–1528. https://doi.org/10.1109/CVPR.2011.5995347

[33] Jasper R. R. Uijlings, Koen E. A. van de Sande, Theo Gevers, and Arnold W. M. Smeulders. 2013. Selective Search for Object Recognition. *Int. J. Comput. Vis.* 104, 2 (2013), 154–171. https://doi.org/10.1007/s11263-013-0620-5

[34] Fang Wan, Chang Liu, Wei Ke, Xiangyang Ji, Jianbin Jiao, and Qixiang Ye. 2019. C-MIL: Continuation Multiple Instance Learning for Weakly Supervised Object Detection. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*. Computer Vision Foundation / IEEE, 2199–2208. https://doi.org/10.1109/CVPR.2019.00230

[35] Jingdong Wang, Huaizu Jiang, Zejian Yuan, Ming-Ming Cheng, Xiaowei Hu, and Nanning Zheng. 2017. Salient Object Detection: A Discriminative Regional Feature Integration Approach. *Int. J. Comput. Vis.* 123, 2 (2017), 251–268. https://doi.org/10.1007/s11263-016-0977-3

[36] Tongzhou Wang and Phillip Isola. 2020. Understanding Contrastive Representation Learning through Alignment and Uniformity on the Hypersphere. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event (Proceedings of Machine Learning Research, Vol. 119)*. PMLR, 9929–9939. http://proceedings.mlr.press/v119/wang20k.html

[37] Yunchao Wei, Xiaodan Liang, Yunpeng Chen, Xiaohui Shen, Ming-Ming Cheng, Jiashi Feng, Yao Zhao, and Shuicheng Yan. 2017. STC: A Simple to Complex Framework for Weakly-Supervised Semantic Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* 39, 11 (2017), 2314–2320. https://doi.org/10.1109/TPAMI.2016.2636150

[38] Gao Yan, Boxiao Liu, Nan Guo, Xiaochun Ye, Fang Wan, Haihang You, and Dongrui Fan. 2019. C-MIDN: Coupled Multiple Instance Detection Network With Segmentation Guidance for Weakly Supervised Object Detection. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*. IEEE, 9833–9842. https://doi.org/10.1109/ICCV.2019.00993

[39] Yazhou Yao, Jian Zhang, Fumin Shen, Xian-Sheng Hua, Jingsong Xu, and Zhenmin Tang. 2016. Automatic image dataset construction with multiple textual metadata. In *IEEE International Conference on Multimedia and Expo, ICME 2016, Seattle, WA, USA, July 11-15, 2016*. IEEE Computer Society, 1–6. https://doi.org/10.1109/ICME.2016.7552988

[40] Yufei Yin, Jiajun Deng, Wengang Zhou, and Houqiang Li. 2021. Instance Mining with Class Feature Banks for Weakly Supervised Object Detection. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*. AAAI Press, 3190–3198. https://ojs.aaai.org/index.php/AAAI/article/view/16429

[41] Yongqiang Zhang, Yancheng Bai, Mingli Ding, Yongqiang Li, and Bernard Ghanem. 2018. W2F: A Weakly-Supervised to Fully-Supervised Framework for Object Detection. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*. Computer Vision Foundation / IEEE Computer Society, 928–936. https://doi.org/10.1109/CVPR.2018.00103

[42] Yukun Zhu, Ryan Kiros, Richard S. Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning Books and Movies: Towards Story-Like Visual Explanations by Watching Movies and Reading Books. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*. IEEE Computer Society, 19–27. https://doi.org/10.1109/ICCV.2015.11

[43] C. Lawrence Zitnick and Piotr Dollár. 2014. Edge Boxes: Locating Object Proposals from Edges. In *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V (Lecture Notes in Computer Science, Vol. 8693)*, David J. Fleet, Tomás Pajdla, Bernt Schiele, and Tinne Tuytelaars (Eds.). Springer, 391–405. https://doi.org/10.1007/978-3-319-10602-1_26