Retrieval-Augmented Open-Vocabulary Object Detection

Jooyeon Kim[1,*] Eulrang Cho[2,*,†] Sehyung Kim[1] Hyunwoo J. Kim[1,‡]
[1]Department of Computer Science and Engineering, Korea University [2]Samsung Research
{parang, shkim129, hyunwoojkim}@korea.ac.kr eulrang.cho@samsung.com

Abstract

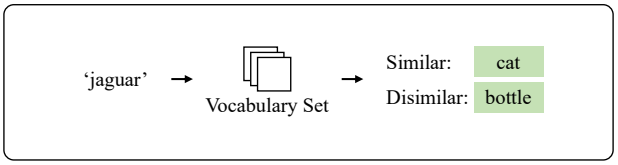
Open-vocabulary object detection (OVD) has been studied with Vision-Language Models (VLMs) to detect novel objects beyond the pre-trained categories. Previous approaches improve the generalization ability to expand the knowledge of the detector, using 'positive' pseudo-labels with additional 'class' names, e.g., *sock, iPod, and alligator. To extend the previous methods in two aspects, we propose Retrieval-Augmented Losses and visual Features (RALF). Our method retrieves related 'negative' classes and augments loss functions. Also, visual features are augmented with 'verbalized concepts' of classes,* e.g., *worn on the feet, handheld music player, and sharp teeth. Specifically, RALF consists of two modules: Retrieval Augmented Losses (RAL) and Retrieval-Augmented visual Features (RAF). RAL constitutes two losses reflecting the semantic similarity with negative vocabularies. In addition, RAF augments visual features with the verbalized concepts from a large language model (LLM). Our experiments demonstrate the effectiveness of RALF on COCO and LVIS benchmark datasets. We achieve improvement up to 3.4 box AP^N_{50} on novel categories of the COCO dataset and 3.6 mask AP_r gains on the LVIS dataset. Code is available at* https://github.com/mlvlab/RALF.

1. Introduction

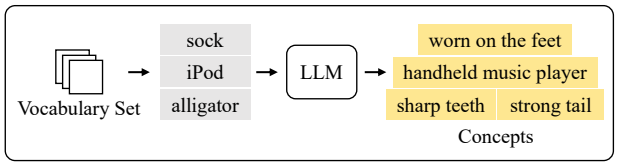
Open-vocabulary object detection (OVD) aims to detect objects belonging to open-set categories. It is a challenging task because novel categories do not appear during training. Pre-trained vision-language models (VLMs) enable the detector to recognize base categories as well as novel categories via their zero-shot visual recognition ability learned from large-scale image-text pairs from the Internet. For instance, CLIP [19] and ALIGN [12] are widely used in OVD to classify unseen objects [8, 16, 28].

Knowledge distillation is one approach to transfer the knowledge of VLMs to the detector. It has been extensively explored in the literature [5, 8, 23, 28, 36]. For the more effective knowledge distillation, some works have endeavored to match words at the region level rather than at the image level to ensure better alignment [23, 28, 36]. To improve generalization to novel categories, several studies employ pseudo-labeling to expand detector knowledge [4, 6, 35, 36]. Pseudo-labels (positive classes) are generated for region proposals by matching their visual features with words from additional vocabulary sets or captions. So, the additional vocabulary sets are constructed only focusing on 'positive' classes. We believe that more diverse vocabulary sets such as 'negative' classes and verbalized (visual) concepts with new techniques will open the door to further improve OVD frameworks.

To study the underexplored directions, in this paper we propose methods utilizing 'negative' classes and 'verbalized concepts' to more effectively use vocabulary sets to enhance the generalization ability to novel categories. One



(a) Retrieving hard (similar) and easy negative (dissimilar) classes.



(b) Verbalized concepts to augment visual features

Figure 1. **Negative vocabularies and verbalized concepts from a large vocabulary set.** (a) Example of negative vocabularies that can be derived from a large vocabulary set. From the vocabulary set, 'cat' and 'bottle' can be retrieved as hard negative (similar) and easy negative (dissimilar) vocabulary, given the category 'jaguar'. (b) Example of verbalized concepts that are generated from LLMs. The concepts of the objects provide more detailed information about the object, such as the attributes.

*Equal contribution.
†This work was done when she was working at Korea University.
‡Corresponding author.

approach involves retrieving vocabularies that have semantic relationships from large vocabulary sets when given a specific category. For example, given the category 'jaguar', we retrieve 'cat' and 'bottle' as hard (similar) and easy (dissimilar) negative vocabulary, respectively as shown in Figure 1a. Using these negative classes, we augment loss functions. Another approach is generating verbalized concepts that describe the attributes (*e.g.* sharp teeth) of classes by LLMs, as depicted in Figure 1b. This information is useful in learning effective representations.

We present a novel framework, **R**etrieval-**A**ugmented **L**osses and visual **F**eatures (RALF), which retrieves vocabularies and concepts from a large vocabulary set and augments losses and visual features. RALF is composed of two parts: Retrieval-Augmented Losses (RAL) and Retrieval-Augmented visual Features (RAF). Given the ground-truth label, we construct hard and easy negative vocabularies by retrieving from the vocabulary store based on similarity. Then, RAL optimizes the distance between the ground-truth label and pre-defined vocabularies with additional losses. Additionally, we utilize a large language model (LLM) to obtain abundant information rather than word units. After generating descriptions about large vocabularies from LLM, we extract verbalized concept details that represent the characteristics of the objects and pile them in a concept store. During inference time, RAF augments visual features with the verbalized concepts retrieved from the concept store. Then the enhanced features are used in classification. To validate the effectiveness of RALF, we conducted experiments on COCO and LVIS benchmarks. Overall, RALF improves the generalization ability of the detector.

Our contributions are threefold:

- We present **RALF** that retrieves vocabularies and augments losses and visual features to improve the generalizability of open-vocabulary object detectors.
- **RAL** optimizes an embedding space by reflecting the distance between ground-truth labels and negative vocabularies from a large vocabulary set. And **RAF** augments visual features with verbalized concepts relevant to visual attributes in images.
- With the method, we achieve 41.3 AP^{N}_{50} in novel categories on the COCO benchmark and also reach 21.9 mask AP_r in novel categories on LVIS.

2. Related works

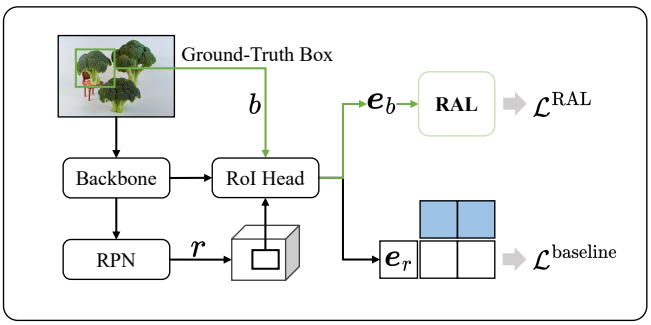
Pre-trained vision-language model. Pre-trained VLMs such as CLIP [19] and ALIGN [12] are trained on large image-text pair datasets via contrastive learning for joint representation in visual and language modalities. Pre-trained VLMs consist of two encoders - image encoder and text encoder - extracting image embedding and text embedding, respectively. By applying contrastive learning, they can be aligned with each other in the same latent space. For this reason, pre-trained VLMs have superior generalizability and transferability to various downstream tasks. For example, CLIP [19] which is pre-trained on extensive image-text pair data, shows impressive performance on zero-shot image classification tasks. Recently, thanks to the success of CLIP, there have been many studies for introducing VLMs to various downstream tasks such as image segmentation [15, 22, 37], image generation [20, 29], and object detection [8, 16, 23, 28, 30].

Open-vocabulary object detection. Object detection task refers to a task that detects an object in a scene and classifies the detected object. A representative study, Fast R-CNN [7], shows excellent object detection performance with CNN architecture. However, there is a limit to the object detection task that requires a lot of human cost for annotation. A zero-shot object detection approach was presented to determine whether a detector can detect a category that was not seen during learning. Recently, open-vocabulary object detection (OVD) [28, 30, 31] has attracted attention. OVD evaluates the ability to predict novel categories by learning using additional caption data such as CC3M [25]. As pre-trained VLMs trained with large datasets perform well with zero-shot for various downstream tasks, diverse approaches to solve through pre-trained VLMs have been studied in OVD. ViLD [8], the most representative study, uses knowledge from CLIP, one of the pre-trained VLMs, and learns the class-agnostic region proposal to perform well for unseen classes. Many follow-up studies have also been conducted on this, and Object-Centric-OVD [23] proposes an object-centric alignment method to solve the localization problem that occurs when CLIP is applied to OVD.

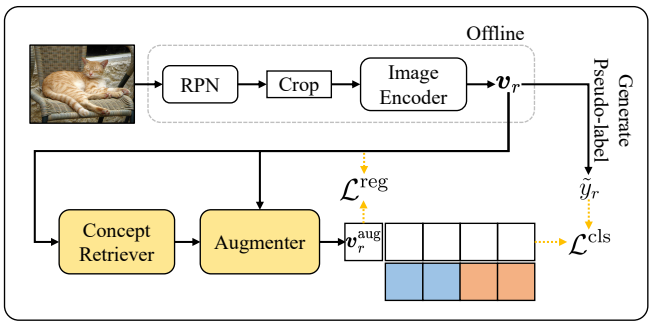
Retrieval-augmentation. Retrieval augmentation was initially introduced in language generation tasks for parameter efficiency. RAG [14] introduces generation models that combine parametric and non-parametric memory access. Recently, retrieval augmentation has been utilized in many vision tasks [18, 26, 32]. RDMs [1] suggest efficiently storing an image database and conditioning a relatively compact generative model. EXTRA [21] proposes a retrieval-augmented image captioning model that enhances performance by leveraging cross-modal representations. Unlike these methods employing retrieval augmentation in generation tasks, we applied retrieval augmentation to the OVD task for the first time in our knowledge.

3. Method

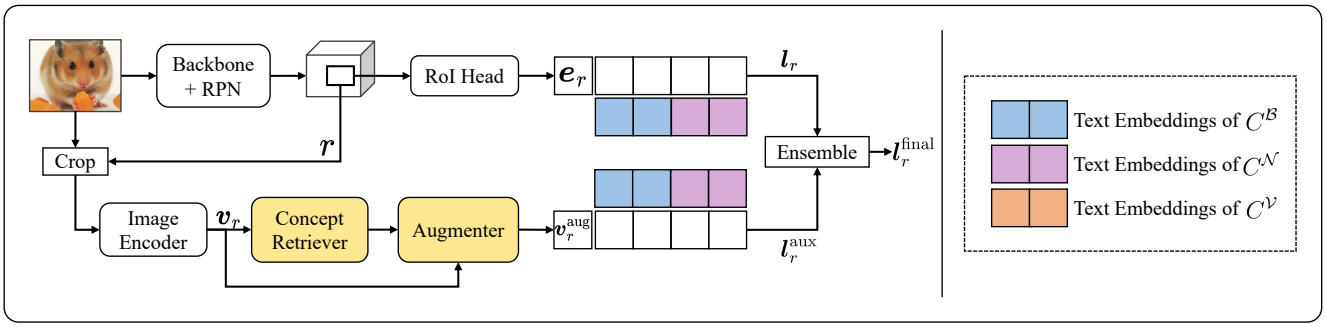
In this section, we proposed a new framework **RALF** that **R**etrieves information from a large vocabulary store and **A**ugments **L**osses and visual **F**eatures. Before we delve into RALF, we briefly introduce the open-vocabulary ob-



(a) Training pipeline w/RAL.



(b) RAF training.



(c) Inference pipeline w/RAF.

Figure 2. **Overall pipeline of RALF.** (a) The first module, RAL, is utilized during detector training. Given a ground-truth box b, the ground-truth box embedding $\boldsymbol{e}_b$ is extracted and used to define $\mathcal{L}^{\text{RAL}}$, which is augmented with hard and easy negative vocabulary. The augmented loss $\mathcal{L}^{\text{RAL}}$ and the baseline loss $\mathcal{L}^{\text{baseline}}$ are employed together to train the detector. The illustration of the other branches (*e.g.*, box regression, distillation, and mask prediction) is omitted in both training and inference pipelines. (b) The second module, RAF, augments visual features with verbalized concepts and is pre-trained before being used in the inference pipeline. Augmented visual features $\boldsymbol{v}_r^{\text{aug}}$ are created through a process involving concept retriever and augmenter, using visual features $\boldsymbol{v}_r$ generated from object proposals in offline. RAF is trained with two losses ($\mathcal{L}^{\text{cls}}$ and $\mathcal{L}^{\text{reg}}$), utilizing $\boldsymbol{v}_r$, $\boldsymbol{v}_r^{\text{aug}}$, and $\tilde{y}_r$, which is the pseudo-label of visual feature. (c) During detector inference time, the trained RAF is utilized. Classification logits $\boldsymbol{l}_r$ trained by RAL and auxiliary logits $\boldsymbol{l}_r^{\text{aux}}$ influenced by RAF are computed with text embeddings of test categories. Then, the final logits $\boldsymbol{l}_r^{\text{final}}$ are determined through an ensemble of $\boldsymbol{l}_r$ and $\boldsymbol{l}_r^{\text{aux}}$.

ject detection task in Section 3.1. The overall pipeline of the proposed method is described in Section 3.2. Our method consists of two modules: 1) Retrieval-Augmented Losses (RAL) to train the object detector, and 2) Retrieval-Augmented visual Features (RAF) using the generated concepts by a large language model. The modules are presented in Section 3.3 and Section 3.4, respectively.

3.1. Preliminaries

Open-vocabulary object detection (OVD) is an advanced visual recognition task that extends the capability of traditional object detectors beyond pre-trained categories. OVD aims to *localize* and *classify* a wide range of objects, including the categories not encountered during training. In OVD, the pre-trained categories and unseen categories are called base categories $C^{\mathcal{B}}$ and novel categories $C^{\mathcal{N}}$, respectively. In general, the approaches for OVD leverage a pre-trained region proposal method, *e.g.*, Region Proposal Network (RPN), for category-agnostic (initial) localization. After the region proposal step, OVD methods utilize a pre-trained vision-language model to classify a wide range of categories in a zero-shot learning manner. Specifically, given region proposal r, the methods extract a region embedding $\boldsymbol{e}_r \in \mathbb{R}^d$ and compute the similarity or perform zero-shot classification with the text embeddings of category $\mathcal{T}(c) \in \mathbb{R}^d$, where $\mathcal{T}$ is the text encoder and c is either a base or new category, *i.e.*, $c \in C^{\mathcal{B}} \cup C^{\mathcal{N}}$. Unless explicitly stated, vectors are row vectors in this paper.

3.2. Overview of RALF

We present the overall pipeline of **R**etrieval-**A**ugmented **L**osses and visual **F**eatures (**RALF**). As described in Figure 2, RALF consists of two modules: 1) **RAL** (Figure 2a), which retrieves negative vocabularies from the vocabulary store by semantic similarity, and enhances the loss function for training object detectors, and 2) **RAF** (Figure 2b), which augments visual features using verbalized concepts by a large language model given retrieved vocabulary.

In RAL, we define two negative vocabulary sets $V_{y_b}^{\text{hard}}$ and $V_{y_b}^{\text{easy}}$ to train the object detector. They are retrieved based on the semantic similarity score between the ground-truth class label y_b and vocabularies from the external vocabulary store $C^{\mathcal{V}}$. In RAF, we extract the concepts associated with objects by LLM and retrieve them to augment visual feature $\boldsymbol{v}_r \in \mathbb{R}^d$ by using the augmenter $\mathcal{A}$ for better classification.

Following the conventional OVD setup, our detector is trained with annotated bounding boxes of base categories $C^{\mathcal{B}}$. As shown in Figure 2a, retrieval-augmented losses are additionally used for training. The total loss for training the object detector is defined as follows:

$$\mathcal{L}^{\text{total}} = \mathcal{L}^{\text{baseline}} + \mathcal{L}^{\text{RAL}}, \tag{1}$$

where $\mathcal{L}^{\text{baseline}}$ denotes the loss of the baseline and $\mathcal{L}^{\text{RAL}}$ is explained in Section 3.3. We utilize the obtained conceptual information to train the RAF module independently. As illustrated in Figure 2c, RAF augments the visual feature in a plug-in manner during inference time. Detailed explanations of RAL and RAF will be introduced in Section 3.3 and Section 3.4, respectively.

3.3. Retrieval-Augmented Losses

In our method, we introduce Retrieval-Augmented Losses (RAL), a novel framework utilizing a large vocabulary store to enhance the detector's generalization power across both base and novel object categories. Our approach, as described in Figure 3, involves the creation of distinct negative vocabulary sets – categorized as 'hard' and 'easy' – based on their similarities to the ground-truth category. RALF trains the detector with RAL, which is established with triplet losses between hard and easy negative vocabulary and ground-truth box.

Negative retriever. To derive hard and easy negative vocabularies, we exploit a large vocabulary set containing a wider range of object classes. First, we remove redundant categories from a vocabulary set and refine the vocabulary store $C^{\mathcal{V}}$ to prevent the possibility that the detector may see novel categories $C^{\mathcal{N}}$, *i.e.*, $C^{\mathcal{V}} \cap C^{\mathcal{N}} = \varnothing$. Then, we define hard negative vocabulary $V_{y_b}^{\text{hard}}$ and easy negative vocabulary $V_{y_b}^{\text{easy}}$ using relative similarity. Specifically, given a ground-truth class label y_b, we obtain text embedding $\mathcal{T}(y_b) \in \mathbb{R}^d$. The negative retriever samples hard and easy negative vocabularies from $C^{\mathcal{V}}$ with respect to cosine similarity between $\mathcal{T}(y_b)$ and $\mathcal{T}(C^{\mathcal{V}})$. However, we observed that some vocabularies have constantly high (or low) similarity scores for any base category. In this case, the vocabularies are not useful to augment losses. To mitigate this issue, we adopt the rank variance sampling scheme, which filters out the vocabularies based on the variance of their rankings measured by similarity. Specifically, we first measure the similarity between a base category $c \in C^{\mathcal{B}}$ and all the vocabularies in $C^{\mathcal{V}}$. Then, the ranking of $C_i^{\mathcal{V}}$ given c is defined as:

$$\underset{c \in C^{\mathcal{B}}}{\text{rank}_c}(C_i^{\mathcal{V}}) = 1 + \sum_{j \neq i} \mathbb{1}[\mathcal{T}(c)\mathcal{T}(C_j^{\mathcal{V}})^\top > \mathcal{T}(c)\mathcal{T}(C_i^{\mathcal{V}})^\top]. \tag{2}$$

We finally compute the variance of $\text{rank}_c(C_i^{\mathcal{V}})$ using all $C^{\mathcal{B}}$. Vocabularies with a relatively low variance of rankings are removed. Then, top-m and bottom-m vocabularies based on the similarity score are selected for each base category as hard and easy negative vocabularies, respectively. For each iteration of training, n vocabularies are randomly selected among the m vocabularies to augment losses. Further details of sampling schemes are discussed in the supplementary materials.

Hard and easy negative losses. Hard and easy negative vocabularies are denoted as $V_{y_b}^{\text{hard}}$ and $V_{y_b}^{\text{easy}}$, respectively. $V_{y_b}^{\text{hard}}$ consists of the words similar to ground-truth y_b class whereas $V_{y_b}^{\text{easy}}$ is the least similar words to $V_{y_b}^{\text{hard}}$. Using the two sets of negative vocabularies with the triplet loss, we propose hard negative loss $\mathcal{L}^{\text{hard}}$ and easy negative loss $\mathcal{L}^{\text{easy}}$. Specifically, we first define the average cosine simi-

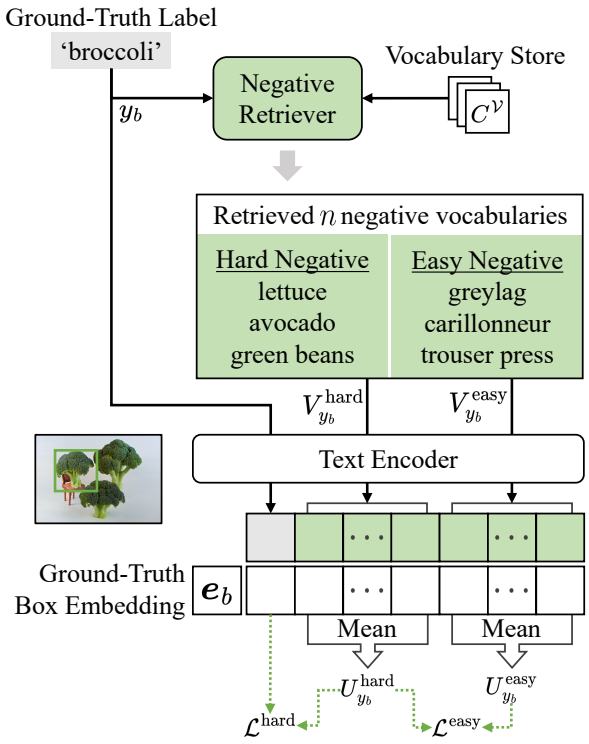


Figure 3. **RAL.** Given ground-truth class label y_b, negative retriever extracts hard negative vocabulary $V_{y_b}^{\text{hard}}$ and easy negative vocabulary $V_{y_b}^{\text{easy}}$ based on semantic similarity with $\mathcal{T}(y_b)$. To enhance the generalizability of the detector, two triplet losses (*i.e.* hard negative loss $\mathcal{L}^{\text{hard}}$ and easy negative loss $\mathcal{L}^{\text{easy}}$) are augmented with $V_{y_b}^{\text{hard}}$, $V_{y_b}^{\text{easy}}$, and ground-truth box embedding $\boldsymbol{e}_b$.

larity with ground-truth embedding $\boldsymbol{e}_b$ as below:

$$U_{y_b}^{\text{hard}} := \frac{1}{n}\mathbf{1}\mathcal{T}(V_{y_b}^{\text{hard}})\boldsymbol{e}_b^\top, \tag{3}$$

$$U_{y_b}^{\text{easy}} := \frac{1}{n}\mathbf{1}\mathcal{T}(V_{y_b}^{\text{easy}})\boldsymbol{e}_b^\top, \tag{4}$$

where $\mathbf{1} \in \mathbb{R}^n$ given n vocabularies. Then, the hard negative loss $\mathcal{L}^{\text{hard}}$ and easy negative loss $\mathcal{L}^{\text{easy}}$ are defined as follows:

$$\mathcal{L}^{\text{hard}} = \max\left(\lambda^{\text{hard}} U_{y_b}^{\text{hard}} - \mathcal{T}(y_b)\boldsymbol{e}_b^\top + \alpha^{\text{hard}}, 0\right), \tag{5}$$

$$\mathcal{L}^{\text{easy}} = \max\left(\lambda^{\text{easy}} U_{y_b}^{\text{easy}} - U_{y_b}^{\text{hard}} + \alpha^{\text{easy}}, 0\right), \tag{6}$$

where λ^{hard} and λ^{easy} are hyperparameters, and α^{hard} and α^{easy} denote margins. To sum up, hard negative loss $\mathcal{L}^{\text{hard}}$ encourages $\boldsymbol{e}_b$ to have higher similarity with y_b than $V_{y_b}^{\text{hard}}$. Easy negative loss $\mathcal{L}^{\text{easy}}$ prompts $V_{y_b}^{\text{hard}}$ to exhibit relatively higher similarity to y_b than $V_{y_b}^{\text{easy}}$. The total loss $\mathcal{L}^{\text{RAL}}$ is computed with a summation of hard negative loss and easy negative loss as follows:

$$\mathcal{L}^{\text{RAL}} = \beta^{\text{hard}}\mathcal{L}^{\text{hard}} + \beta^{\text{easy}}\mathcal{L}^{\text{easy}}, \tag{7}$$

where β^{hard} and β^{easy} are hyperparameters.

3.4. Retrieval-Augmented visual Features

We introduce Retrieval-Augmented visual Features (RAF) that augments visual features using the *verbalized concepts* of each object category, as depicted in Figure 4.

Concept store. The concept store consists of a set of characteristics describing the object (*e.g.*, color, scale, shape, etc.). The characteristics for $C^{\mathcal{B}} \cup C^{\mathcal{V}}$ are generated by a large language model (LLM) using '`Describe what a(n) {vocabulary} looks like.`' prompt template, following [13]. Note that $C^{\mathcal{N}}$ is not used to generate verbalized concepts. We remove meaningless words such as prepositions from descriptions generated by LLM and store only meaningful noun chunks in the concept store.

Concept retriever. Concepts for augmenting visual features are retrieved by the concept retriever. Concept embeddings H are obtained from the text encoder $\mathcal{T}$ with the concepts from the concept store. Given visual feature $\boldsymbol{v}_r \in \mathbb{R}^d$, the concept retriever calculates the cosine similarity between concept embeddings H and visual feature $\boldsymbol{v}_r$. Then, it returns the k most relevant concept embeddings $H_r \in \mathbb{R}^{k\times d}$ and corresponding scores $\boldsymbol{s}_r \in \mathbb{R}^k$.

Augmenter. We propose an augmenter $\mathcal{A}$ that augments visual feature $\boldsymbol{v}_r$ with the retrieved concepts. Let $\boldsymbol{v}_r^{\text{aug}} \in \mathbb{R}^d$ denote a combination of $\boldsymbol{v}_r^{\text{coarse}} \in \mathbb{R}^d$ and $\boldsymbol{v}_r^{\text{fine}} \in \mathbb{R}^d$. $\boldsymbol{v}_r^{\text{coarse}}$ is calculated as:

$$\boldsymbol{v}_r^{\text{coarse}} = \text{Proj}(\boldsymbol{v}_r), \tag{8}$$

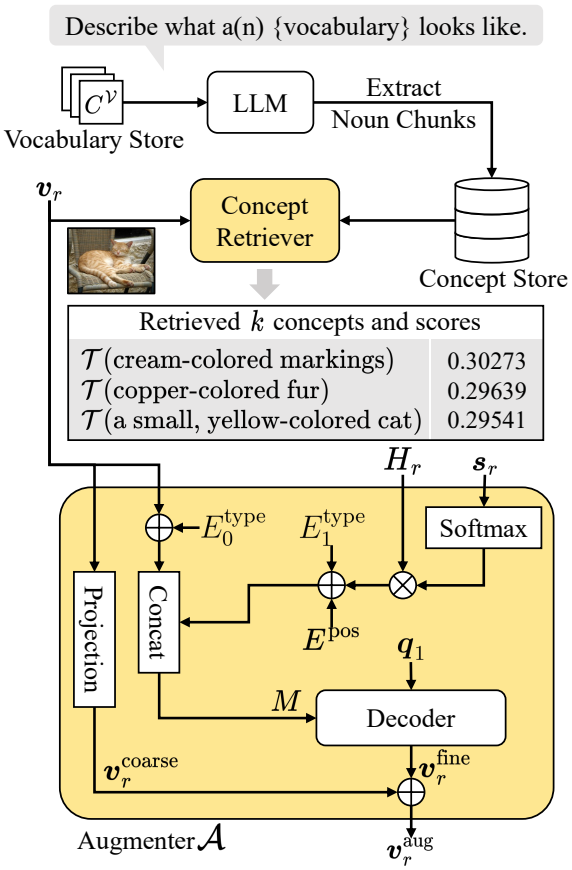


Figure 4. **RAF.** Verbalized concepts are generated by LLM with prompts and stored in the concept store. Given a visual feature $\boldsymbol{v}_r$, relevant concept embeddings H_r and scores $\boldsymbol{s}_r$ are retrieved by the concept retriever. Then the augmenter $\mathcal{A}$ creates augmented visual feature $\boldsymbol{v}_r^{\text{aug}}$ with related verbalized concepts.

where Proj$(\cdot)$ is a linear projection. On the other hand, $\boldsymbol{v}_r^{\text{fine}}$ is enhanced with the retrieved concepts. It is the final output of the decoder that has query embedding $\boldsymbol{q}_1 \in \mathbb{R}^d$ as query and $M \in \mathbb{R}^{(1+k)\times d}$ as key and value. M is computed as:

$$M = (\boldsymbol{v}_r + E_0^{\text{type}}) \| (\text{diag}(\text{softmax}(\boldsymbol{s}_r))H_r + E^{\text{pos}} + E_1^{\text{type}}), \tag{9}$$

where diag$(\cdot)$ denotes a diagonal matrix function and $\|$ is the concatenation function. $E^{\text{pos}} \in \mathbb{R}^{k\times d}$ denotes positional embeddings to determine how many concepts will be utilized. $E_0^{\text{type}} \in \mathbb{R}^d$ and $E_1^{\text{type}} \in \mathbb{R}^d$ are type embeddings to distinguish $\boldsymbol{v}_r$ between information from retrieved concepts. E_1^{type} is replicated to $\mathbb{R}^{k\times d}$ in Eq. (9). The augmenter $\mathcal{A}$ consists of L decoder layers. The operation in l-th decoder layer is as follows:

$$\begin{aligned} \boldsymbol{q}_l' &= \boldsymbol{q}_l + \text{CA}(\boldsymbol{q}_l, M, M), \\ \boldsymbol{q}_{l+1} &= \boldsymbol{q}_l' + \text{FFN}(\boldsymbol{q}_l'), \\ \boldsymbol{v}_r^{\text{fine}} &= \boldsymbol{q}_L, \end{aligned} \tag{10}$$

where $l \in \{1, \dots, L\}$. The term CA represents cross-attention operation, while FFN stands for feed-forward network. Finally, the augmented visual feature $\boldsymbol{v}_r^{\text{aug}}$ is obtained by adding the coarse and fine feature of $\boldsymbol{v}_r$ as below:

$$\boldsymbol{v}_r^{\text{aug}} = \boldsymbol{v}_r^{\text{coarse}} + \boldsymbol{v}_r^{\text{fine}}. \tag{11}$$

At test time, as depicted in Figure 2c the augmented visual features and text embeddings of test categories are used to compute auxiliary logits $\boldsymbol{l}_r^{\text{aux}}$:

$$\boldsymbol{l}_r^{\text{aux}} = \boldsymbol{v}_r^{\text{aug}} \mathcal{T}(C^{\mathcal{B}} \cup C^{\mathcal{N}})^{\top}. \tag{12}$$

Then the auxiliary logits are ensembled with $\boldsymbol{l}_r$ to compute the final logits $\boldsymbol{l}_r^{\text{final}}$ for the final classification of proposal r. The details of the logit ensemble are described in Section 4.

Loss for RAF training. We pre-trained RAF with visual features from region proposals. For pre-training, we use classification loss $\mathcal{L}^{\text{cls}}$ and regularization loss $\mathcal{L}^{\text{reg}}$. We first define the pseudo-label of region proposal r as $\tilde{y}_r$:

$$\tilde{y}_r = \underset{c \in (C^{\mathcal{B}} \cup C^{\mathcal{V}})}{\text{argmax}} (\mathcal{T}(c) \boldsymbol{v}_r^{\top}). \tag{13}$$

Then, using the pseudo-labels, classification loss $\mathcal{L}^{\text{cls}}$ is defined as:

$$\mathcal{L}^{\text{cls}} = \frac{1}{N} \sum_r \mathcal{L}^{\text{ce}}(\boldsymbol{v}_r^{\text{aug}} \mathcal{T}(C^{\mathcal{B}} \cup C^{\mathcal{V}})^{\top}, \tilde{y}_r), \tag{14}$$

where N is the number of proposals per image and $\mathcal{L}^{\text{ce}}$ is the cross entropy loss. We encourage augmented visual feature $\boldsymbol{v}_r^{\text{aug}}$ to similar to the original visual feature $\boldsymbol{v}_r$ using regularization loss $\mathcal{L}^{\text{reg}}$ defined as below:

$$\mathcal{L}^{\text{reg}} = \frac{1}{N} \sum_r (\boldsymbol{v}_r^{\text{aug}} - \boldsymbol{v}_r)^2. \tag{15}$$

Finally, the total loss $\mathcal{L}^{\text{RAF}}$ for RAF training is the combination of Eq. (14) and Eq. (15) as follows:

$$\mathcal{L}^{\text{RAF}} = \beta^{\text{cls}} \mathcal{L}^{\text{cls}} + \beta^{\text{reg}} \mathcal{L}^{\text{reg}}, \tag{16}$$

where β^{cls} and β^{reg} are hyperparameters.

4. Experiments

In this section, we briefly discuss the experimental setup, including datasets and implementation details. Next, we evaluate the performance of RALF compared to various baselines and investigate RALF with further analysis.

Datasets. We evaluate RALF on two public benchmarks, COCO [17] and LVIS [9]. In the open-vocabulary object detection setting, COCO dataset is split into 48 base categories and 17 novel categories following OVR-CNN [34]. It includes 118k images, which separated 107,761 images for training and 4,836 images for validation. Referring to ViLD [8], we divide the LVIS dataset into 866 base categories and 337 novel categories. For both benchmarks, we use base categories during training, yet novel categories are also validated for inference. We adopt the mean average precision (mAP) as the evaluation metric. We report AP_{50}, $\text{AP}_{50}^{\text{B}}$, and $\text{AP}_{50}^{\text{N}}$ for COCO and AP_{r}, AP_{c}, AP_{f}, and AP for LVIS. Note that instance segmentation (mAP) results are reported on LVIS. To generate the vocabulary store, we adopt V3Det [27] as a vocabulary set.

Implementation details. We implement RALF with pre-trained CLIP [19] with ViT-B/32 image encoder backbone and text encoder from the official repository. Note that we freeze all parameters in image and text encoders during training. Additionally, we use GPT-3 [2] DaVinci-002 to generate descriptions from RAF as a large language model. We use Faster R-CNN [24] with ResNet-50 [11] backbone and Mask R-CNN [10] with ResNet-50 FPN backbone for COCO and LVIS, respectively. For training RAF, we use region proposals provided from OADP [28]. We set our baselines as OADP [28], Object-Centric-OVD [23], and DetPro [5]. As RALF enhances the generalizability power via a plug-in manner, we construct RALF into baselines. In all experiments, we train and evaluate on NVIDIA RTX-3090 4 GPUs. More implementation details, including hyperparameters, are discussed in the supplementary materials.

Logit ensemble in RAF. Since each baseline has a different range of logits, the final logits $\boldsymbol{l}_r^{\text{final}}$ are calculated as below:

$$\boldsymbol{l}_r^{\text{final}} = \begin{cases} \boldsymbol{l}_r + \sigma(\boldsymbol{l}_r^{\text{aux}}) & \text{if Object-Centric-OVD,} \\ \boldsymbol{l}_r * (\sigma(\boldsymbol{l}_r^{\text{aux}}) - 0.25) & \text{if DetPro,} \\ \boldsymbol{l}_r + \boldsymbol{l}_r^{\text{aux}} & \text{if OADP,} \end{cases} \tag{17}$$

where $\sigma(\cdot)$ denotes the sigmoid function. Without using all values of auxiliary logits when adding to $\boldsymbol{l}_r$, only top-1 figures on COCO and top-10 or 20 figures on LVIS are utilized, considering the number of test categories.

4.1. Main results

We evaluate RALF on COCO and LVIS benchmarks in the open-vocabulary object detection (OVD) setting and compare with various baselines. Overall results are reported in Table 1 and Table 2.

COCO benchmark. As reported in Table 1, RALF shows great performance enhancement when plugged into baselines for all evaluation metrics. RALF significantly improved 4.7 $\text{AP}_{50}^{\text{N}}$ when plugged into Object-Centric-OVD [23] and achieved state-of-the-art results. Moreover, RALF surpasses Object-Centric-OVD for not only novel categories but also base and all categories, with 0.3 $\text{AP}_{50}^{\text{B}}$ and 1.5 AP_{50} improvements, respectively. Like the inclination observed in the Object-Centric-OVD, implementing

Method	AP^N_{50}	AP^B_{50}	AP_{50}
ViLD [8]	27.6	59.9	51.2
PB-OVD [6]	29.1	44.4	40.4
OV-DETR [33]	29.4	61.0	52.7
VL-Det [16]	32.0	50.6	45.8
MEDet [3]	32.6	53.5	48.0
BARON [30]	34.0	60.4	53.5
OADP [28]	30.0	53.3	47.2
OADP + RALF	**33.4**	**54.5**	**49.0**
Object-Centric-OVD [23]	36.6	54.0	49.4
Object-Centric-OVD + RALF	**41.3**	**54.3**	**50.9**

Table 1. **Results of OVD on COCO.**

Method	AP_r	AP_c	AP_f	AP
ViLD [8]	16.1	20.0	28.3	22.5
OV-DETR [33]	17.4	25.0	32.5	26.6
BARON [30]	18.0	24.4	28.9	25.1
Object-Centric-OVD [23]	17.1	21.4	26.7	22.8
Object-Centric-OVD + RALF	**18.5**	21.0	26.3	22.6
DetPro [5]	19.8	25.6	28.9	25.9
DetPro + RALF	**21.1**	**25.7**	**29.2**	**26.3**
OADP [28]	19.9	26.0	28.7	26.0
OADP + RALF	**21.9**	**26.2**	**29.1**	**26.6**

Table 2. **Results of OVD on LVIS.**

RALF on OADP [28] demonstrates remarkable effectiveness. The results show performance gains for the novel, base, and all categories by 3.4 AP^N_{50}, 1.2 AP^B_{50}, and 1.8 AP_{50}, correspondingly.

LVIS benchmark. To verify that our method improves performance in various cases, we experimented on the LVIS benchmark and added another baseline – DetPro [5]. The results are noted in Table 2. Overall, RALF improves detection ability on novel categories for all baselines by up to 2.0 AP_r. Although Object-Centric-OVD [23] slightly decreases except AP_r, RALF improves all metrics on DetPro and OADP [28]. To summarize, the results imply that RALF improves generalizability.

4.2. Ablation study

As discussed in Section 3.2, RALF consists of two modules – RAL and RAF. We conducted ablation studies on RAL and RAF to verify the effectiveness of each module on COCO and LVIS benchmarks.

Effectiveness of RAL. We demonstrate the results of RAL in Table 3 and Table 4. From the results, RAL improves performance gain in all baselines. RAL shows an improvement of up to 1.3 AP^N_{50} and at least 1.0 AP^N_{50} on COCO. Interestingly, RAL demonstrates exceptional generalizability, specifically within the COCO base categories, exhibiting no performance declines. While there are marginal decreases in AP_c, AP_f, and AP on LVIS, the performance of novel categories is notably increased up to 3.2 AP_r.

Method	RAL	RAF	AP^N_{50}	AP^B_{50}	AP_{50}
OADP† [28]			30.0	54.5	48.1
		✓	31.5	54.3	48.3
	✓		31.3	54.5	48.4
	✓	✓	33.4	54.5	49.0
Object-Centric-OVD† [23]			40.0	53.8	50.2
		✓	40.8	54.0	50.5
	✓		41.0	54.1	50.7
	✓	✓	41.3	54.3	50.9

Table 3. **Ablation study of RALF on COCO dataset.** † denotes reproduce result.

Method	RAL	RAF	AP_r	AP_c	AP_f	AP
Object-Centric-OVD† [23]			18.1	21.4	26.6	22.8
		✓	19.5	21.4	26.6	23.1
	✓		18.5	20.8	26.3	22.6
	✓	✓	18.5	21.0	26.3	22.6
DetPro† [5]			19.9	25.8	29.2	26.1
		✓	20.5	25.8	29.2	26.2
	✓		21.3	25.7	29.2	26.3
	✓	✓	21.1	25.7	29.2	26.3
OADP† [28]			18.3	26.2	28.9	25.9
		✓	19.3	26.2	28.9	26.1
	✓		21.5	26.1	29.1	26.5
	✓	✓	21.9	26.2	29.1	26.6

Table 4. **Ablation study of RALF on LVIS dataset.** † denotes reproduce result.

Effectiveness of RAF. We plug RAF into several baselines and evaluate performance to verify whether RAF is effective. Table 3 and Table 4 show the performance of RAF on COCO and LVIS, respectively. There are notable improvements in overall baselines. In particular, RAF improves the prediction ability for novel categories of OADP and Object-Centric-OVD by 1.5 AP^N_{50} and 0.8 AP^N_{50} on COCO, respectively. Without performance decreasing in base categories, the performance of novel categories improved by up to 1.4 AP_r on LVIS.

From the results in Table 3 and Table 4, irrespective of the baselines, RAL and RAF enhance the prediction of novel categories while maintaining somewhat the prediction of base categories. The combination of RAL and RAF, *i.e.*, RALF, shows significant performance gain of the baselines compared with the sole use of each module. To sum up, RAL and RAF demonstrate enhanced performance individually, and their combined shows superior generalization capabilities.

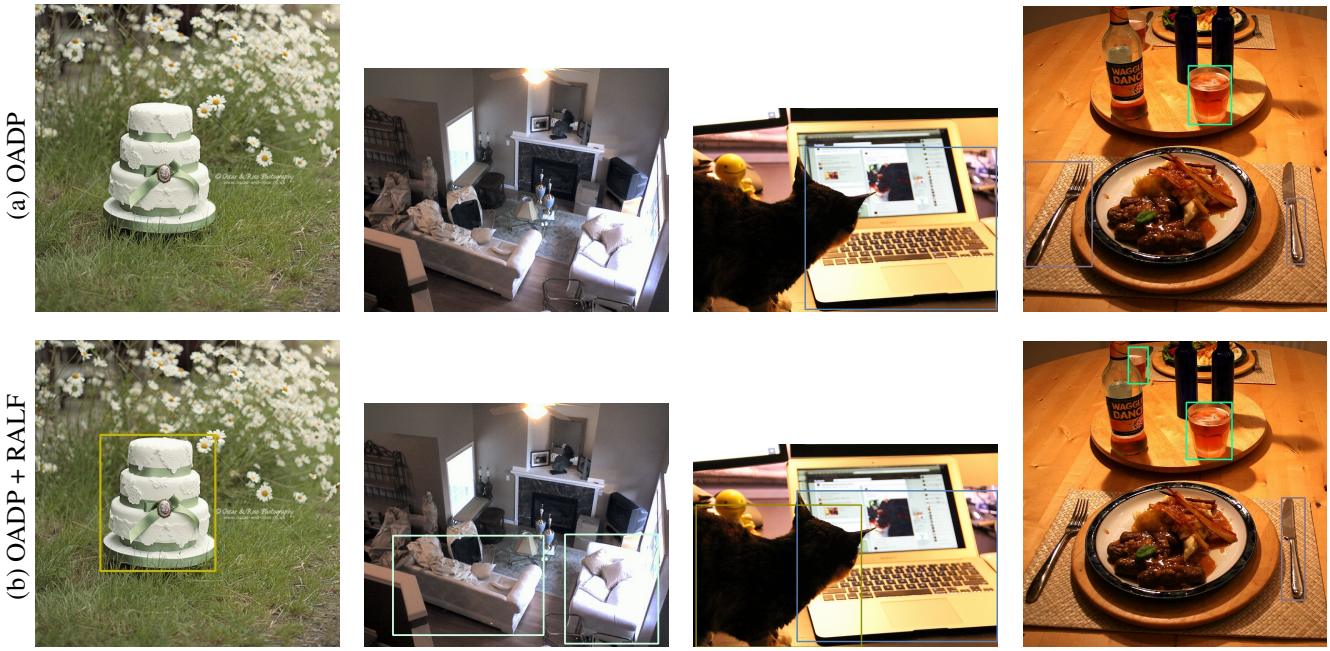


Figure 5. **Qualitative results on COCO.** Results of bounding box predictions on novel categories for (a) OADP and (b) OADP+RALF.

4.3. Analysis on RALF

Easy and hard negatives. We evaluate the effects of various approaches to handle easy and hard negatives. First, we use only one type of negative; 'Easy negative only' and 'Hard negative only'. Second, we merge the easy and hard negatives into one negative group denoted as 'Merged'. Unlike RAL, $\mathcal{L}^{\text{easy}}$ of Eq. (6) that uses two types of negatives is not applied in the 'Merged' setting. Table 5 shows that the baselines above show inferior performance to our method **RAL**. Overall, our method that differentially treats easy and hard negatives boosts performance the most.

	AP^{N}_{50}	AP^{B}_{50}	AP_{50}
Easy negative only	31.0	54.3	48.2
Hard negative only	30.9	54.6	48.4
Merged	30.9	55.5	49.0
RAL	31.3	54.5	48.4

Table 5. **Analysis of easy and hard negatives on COCO.**

4.4. Qualitative results

We visualize the detection results to verify that RALF captures novel categories well. We compare one of our baselines (*i.e.*, OADP) and OADP + RALF at the top and bottom of Figure 5, respectively. Each box in the image represents the box prediction results. From the qualitative results, RALF captures novel categories better than baseline, which means RALF improves generalizability.

5. Conclusion

In this paper, we present Retrieval-Augmented Losses and visual Features (RALF) that retrieves information from a large vocabulary set and augments losses and visual features. To optimize the detector, we add Retrieval-Augmented Losses (RAL), which brings hard and easy negative vocabulary from the pre-defined vocabulary store and reflects the semantic similarity with the ground-truth label. Additionally, Retrieval-Augmented visual Features (RAF) augments visual features with generated concepts from a large language model and enables improved generalizability. To sum up, RALF combines both modules and easily plugs into various detectors and significantly improves detection ability not only base categories but also novel categories.

Acknowledgements. This work was partly supported by ICT Creative Consilience Program through the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant (IITP-2024-2020-0-01819), Convergence security core talent training business (Korea University) grant (No.2022-0-01198), and the National Research Foundation of Korea (NRF) grant (NRF-2023R1A2C2005373) funded by the Korea government (MSIT).

References

[1] Andreas Blattmann, Robin Rombach, Kaan Oktay, Jonas Müller, and Björn Ommer. Retrieval-augmented diffusion models. In *NeurIPS*, 2022. 2

[2] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *NeurIPS*, 2020. 6

[3] Peixian Chen, Kekai Sheng, Mengdan Zhang, Mingbao Lin, Yunhang Shen, Shaohui Lin, Bo Ren, and Ke Li. Open vocabulary object detection with proposal mining and prediction equalization. *arXiv preprint arXiv:2206.11134*, 2022. 7

[4] Han-Cheol Cho, Won Young Jhoo, Wooyoung Kang, and Byungseok Roh. Open-vocabulary object detection using pseudo caption labels. *arXiv preprint arXiv:2303.13040*, 2023. 1

[5] Yu Du, Fangyun Wei, Zihe Zhang, Miaojing Shi, Yue Gao, and Guoqi Li. Learning to prompt for open-vocabulary object detection with vision-language model. In *CVPR*, 2022. 1, 6, 7

[6] Mingfei Gao, Chen Xing, Juan Carlos Niebles, Junnan Li, Ran Xu, Wenhao Liu, and Caiming Xiong. Open vocabulary object detection with pseudo bounding-box labels. In *ECCV*, 2022. 1, 7

[7] Ross Girshick. Fast r-cnn. In *ICCV*, 2015. 2

[8] Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. Open-vocabulary object detection via vision and language knowledge distillation. In *ICLR*, 2022. 1, 2, 6, 7

[9] Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *CVPR*, 2019. 6

[10] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017. 6

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

[12] 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. 1, 2

[13] Prannay Kaul, Weidi Xie, and Andrew Zisserman. Multi-modal classifiers for open-vocabulary object detection. In *ICML*, 2023. 5

[14] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *NeurIPS*, 2020. 2

[15] Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen Koltun, and Rene Ranftl. Language-driven semantic segmentation. In *ICLR*, 2022. 2

[16] Chuang Lin, Peize Sun, Yi Jiang, Ping Luo, Lizhen Qu, Gholamreza Haffari, Zehuan Yuan, and Jianfei Cai. Learning object-language alignments for open-vocabulary object detection. In *ICLR*, 2023. 1, 2, 7

[17] 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. 6

[18] Alexander Long, Wei Yin, Thalaiyasingam Ajanthan, Vu Nguyen, Pulak Purkait, Ravi Garg, Alan Blair, Chunhua Shen, and Anton van den Hengel. Retrieval augmented classification for long-tail visual recognition. In *CVPR*, 2022. 2

[19] 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. 1, 2, 6

[20] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. 2

[21] Rita Ramos, Desmond Elliott, and Bruno Martins. Retrieval-augmented image captioning. In *EACL*, 2023. 2

[22] Yongming Rao, Wenliang Zhao, Guangyi Chen, Yansong Tang, Zheng Zhu, Guan Huang, Jie Zhou, and Jiwen Lu. Denseclip: Language-guided dense prediction with context-aware prompting. In *CVPR*, 2022. 2

[23] Hanoona Rasheed, Muhammad Maaz, Muhammad Uzair Khattak, Salman Khan, and Fahad Shahbaz Khan. Bridging the gap between object and image-level representations for open-vocabulary detection. In *NeurIPS*, 2022. 1, 2, 6, 7

[24] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015. 6

[25] 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. 2

[26] Hung-Yu Tseng, Hsin-Ying Lee, Lu Jiang, Ming-Hsuan Yang, and Weilong Yang. Retrievegan: Image synthesis via differentiable patch retrieval. In *ECCV*, 2020. 2

[27] Jiaqi Wang, Pan Zhang, Tao Chu, Yuhang Cao, Yujie Zhou, Tong Wu, Bin Wang, Conghui He, and Dahua Lin. V3det: Vast vocabulary visual detection dataset. In *ICCV*, 2023. 6

[28] Luting Wang, Yi Liu, Penghui Du, Zihan Ding, Yue Liao, Qiaosong Qi, Biaolong Chen, and Si Liu. Object-aware distillation pyramid for open-vocabulary object detection. In *CVPR*, 2023. 1, 2, 6, 7

[29] Zihao Wang, Wei Liu, Qian He, Xinglong Wu, and Zili Yi. Clip-gen: Language-free training of a text-to-image generator with clip. *arXiv preprint arXiv:2203.00386*, 2022. 2

[30] Size Wu, Wenwei Zhang, Sheng Jin, Wentao Liu, and Chen Change Loy. Aligning bag of regions for open-vocabulary object detection. In *CVPR*, 2023. 2, 7

[31] Xiaoshi Wu, Feng Zhu, Rui Zhao, and Hongsheng Li. Cora: Adapting clip for open-vocabulary detection with region prompting and anchor pre-matching. In *CVPR*, 2023. 2

[32] Rui Xu, Minghao Guo, Jiaqi Wang, Xiaoxiao Li, Bolei Zhou, and Chen Change Loy. Texture memory-augmented deep patch-based image inpainting. *TIP*, 2021. 2

[33] Yuhang Zang, Wei Li, Kaiyang Zhou, Chen Huang, and Chen Change Loy. Open-vocabulary detr with conditional matching. In *ECCV*, 2022. 7

[34] Alireza Zareian, Kevin Dela Rosa, Derek Hao Hu, and Shih-Fu Chang. Open-vocabulary object detection using captions. In *CVPR*, 2021. 6

[35] Shiyu Zhao, Zhixing Zhang, Samuel Schulter, Long Zhao, BG Vijay Kumar, Anastasis Stathopoulos, Manmohan Chandraker, and Dimitris N Metaxas. Exploiting unlabeled data with vision and language models for object detection. In *ECCV*, 2022. 1

[36] 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. 1

[37] Chong Zhou, Chen Change Loy, and Bo Dai. Extract free dense labels from clip. In *ECCV*, 2022. 2