Multi-Modal Classifiers for Open-Vocabulary Object Detection

Prannay Kaul[1] **Weidi Xie**[1 2 3] **Andrew Zisserman**[1]
https://www.robots.ox.ac.uk/vgg/research/mm-ovod/

Abstract

The goal of this paper is open-vocabulary object detection (OVOD) — building a model that can detect objects beyond the set of categories seen at training, thus enabling the user to specify categories of interest at inference without the need for model retraining. We adopt a standard two-stage object detector architecture, and explore three ways for specifying novel categories: via language descriptions, via image exemplars, or via a combination of the two. We make three contributions: *first*, we prompt a large language model (LLM) to generate informative language descriptions for object classes, and construct powerful text-based classifiers; *second*, we employ a visual aggregator on image exemplars that can ingest any number of images as input, forming vision-based classifiers; and *third*, we provide a simple method to fuse information from language descriptions and image exemplars, yielding a multi-modal classifier. When evaluating on the challenging LVIS open-vocabulary benchmark we demonstrate that: (i) our text-based classifiers outperform all previous OVOD works; (ii) our vision-based classifiers perform as well as text-based classifiers in prior work; (iii) using multi-modal classifiers perform better than either modality alone; and finally, (iv) our text-based and multi-modal classifiers yield better performance than a fully-supervised detector.

1. Introduction

In this paper, we consider the problem of open-vocabulary object detection (OVOD), which aims to localise and classify visual objects beyond the categories seen at training time. One may consider its usefulness from the perspective of online inference, when users want to freely specify categories of interest at inference time without the need or ability to re-train models. To specify categories of interest, three obvious ways exist, namely: (1) text-based, *e.g*. name the category or describe it in text form; (2) vision-based, *e.g*. give image examples; (3) multi-modal, *e.g*. indicate the category jointly with text and image.

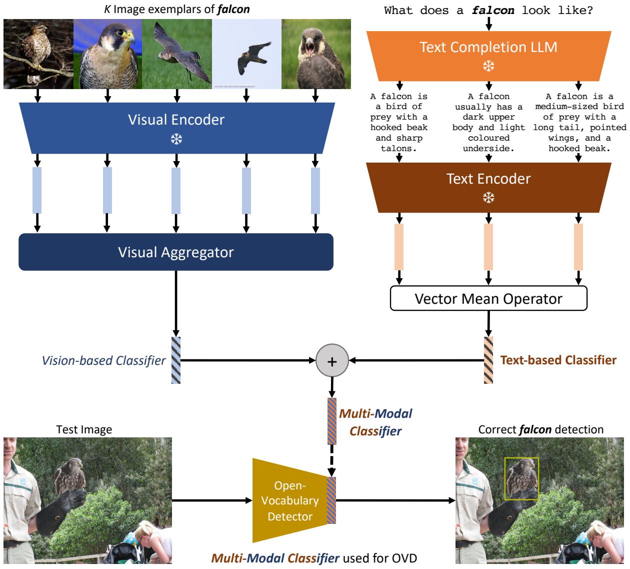
K Image exemplars of *falcon*
Visual Encoder
❄
Visual Aggregator
What does a ***falcon*** look like?
Text Completion LLM
❄
A falcon is a bird of prey with a hooked beak and sharp talons.
A falcon usually has a dark upper body and light coloured underside.
A falcon is a medium-sized bird of prey with a long tail, pointed wings, and a hooked beak.
Text Encoder
❄
Vector Mean Operator
Vision-based Classifier
+
Text-based Classifier
Multi-Modal Classifier
Test Image
Open-Vocabulary Detector
Correct *falcon* detection
Multi-Modal Classifier used for OVD

Figure 1. Overview of the architecture for generating text-based, vision-based, or multi-modal classifiers for OVOD. Vision (top left): A frozen visual encoder ingests image exemplars of **falcon** producing an embedding per exemplar. A trained aggregator takes these embeddings as input and produces a *vision-based classifier*. Text (top right): A text completion LLM is prompted to give descriptions of a **falcon** which are then encoded by a text encoder and averaged yielding a *text-based classifier*. Multi-Modal (middle): *Multi-Modal classifiers* are generated by adding the vision-based and text-based classifiers together. OVOD (bottom): The multi-modal classifier is used to detect the **falcon** in a standard model. Note, all three types of classifier: vision-based, text-based and multi-modal, can be used on the detector head for OVOD.

Existing works (Bansal et al., 2018; Gu et al., 2022; Zareian et al., 2021; Zhou et al., 2022; Feng et al., 2022) have explored replacing the learnt classifiers in a traditional detector with text embeddings, that are computed by passing the class name into a pre-trained text encoder with manual prompts, such as "a photo of a dalmatian", however, this design can

[1] Visual Geometry Group, University of Oxford [2] CMIC, Shanghai Jiao Tong University [3] Shanghai AI Lab. Correspondence to: Prannay Kaul <prannay@robots.ox.ac.uk>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

be sub-optimal from three aspects: *first*, the discriminative power of generated text embeddings relies entirely on the internal representation of the pre-trained text encoder, potentially leading to lexical ambiguities — *e.g.* “nail” can either refer to “the hard surface on the tips of the fingers” or “a small metal spike with a flat tip hammered into wood to form a joint” — simply encoding the class name will not be able to distinguish the two concepts; *second*, the class name for objects of interest may be unknown to the user, while exemplar images can be easily acquired — *e.g.* “dugong” refers to a herbivorous marine mammal with an adorable, plump appearance, a dolphin tail, round head and downward snout (an example is in Section I of the Appendix); *third*, in cases where multi-modal information is preferable to specify the category of interest, *e.g.* a species of butterfly with a distinctive wing pattern — language descriptions can be unsuitably long to capture all the intricacies of a given category, while an exemplar image can “tell a thousand words” and act as an effective complement to text.

To tackle these limitations, we propose a multi-modal open-vocabulary object detector, with the classifier of the detector for a particular category being constructed via natural language descriptions, image exemplars or a combination of the two. Specifically, we establish an automatic approach for sourcing visual descriptions of the object categories, by prompting a large language model with questions, *e.g.* “What does a dalmatian look like?”, yielding “A dalmatian is typically a large dog with a short coat of black spots on a white background”. Such a description provides additional visual cues to enhance the discriminative power of the classifier generated from a text encoder. For cases where collecting suitable informative language descriptions may be difficult or require unnecessarily long descriptions to establish the differences between classes, *e.g.* the dog breeds “pug” and “bulldog” have similar descriptions, we can generate classifiers from image exemplars — RGB images of the class of interest. Finally, we suggest a simple method to fuse both language descriptions and image exemplars, yielding multi-modal classifiers that perform better than either modality individually.

We explore the issue of how best to combine the set of language descriptions and the set of image exemplars into a classifier, by comparing the performance of aggregation methods on a standard detector architecture. By evaluating on the challenging LVIS (Gupta et al., 2019) open-vocabulary object detection benchmark we show that: (i) our automated method for sourcing rich natural language descriptions yields text-based classifiers superior to those of previous work that rely entirely on the class name; (ii) vision-based classifiers can be effectively constructed by a visual aggregator, enabling novel categories to be detected by specifying image exemplars; (iii) natural language descriptions and image exemplars can be simply combined to produce multi-modal classifiers, which perform better than either modality individually, and achieve superior results to existing approaches.

2. Related Work

Closed-Vocabulary Object Detection is one of the classical computer vision problems, making a full overview here impossible. Therefore, we outline some key milestones. In general, modern object detection methods can be cast into two sets: one-stage and two-(multi-)stage detectors. *One-stage* detectors directly classify and regress bounding boxes by either densely classifying a set of predefined anchor boxes (Redmon et al., 2016; Redmon & Farhadi, 2018; Liu et al., 2016; Lin et al., 2017; Tan et al., 2020), each which may contain an object, or densely searching for geometric entities of objects *e.g.* corners, centre points or boxes (Law & Deng, 2018; Zhou et al., 2019; Tian et al., 2019). Conversely, most *two-stage* detectors first propose class-agnostic bounding boxes that are pooled to fixed size region-of-interest (RoI) features and classified by a sub-network in the second stage (Girshick, 2015; Ren et al., 2016; Li et al., 2019). Two-stage detectors are extended to *multi-stage* detectors in which the additional stages refine predictions made by the previous stage (Cai & Vasconcelos, 2018; Chen et al., 2019; Zhou et al., 2021). A unique work in this area is that of (Carion et al., 2020) which uses the Transformer architecture (Vaswani et al., 2017) to treat object detection as a set prediction problem. **Note that**, the classifiers in these object detectors are jointly learnt on a training set, therefore only objects seen at training time can be detected during inference time, thus termed closed-vocabulary object detection.

Open-Vocabulary Object Detection goes beyond closed-vocabulary object detection and enables users to expand/change the detector vocabulary at inference time, without the need for model re-training. Recently, OVOD has seen increased attention and progress primarily driven by the emergence of large-scale vision-language models (VLMs) *e.g.* CLIP and ALIGN (Radford et al., 2021; Jia et al., 2021), which jointly learn image and natural language representations.

ViLD (Gu et al., 2022) distills representations from VLMs. First, groundtruth bounding boxes are used to crop an image and an embedding for the box is sourced from a frozen VLM image encoder. An object detection model is learnt by matching overlapping region-of-interest (RoI) features with the embedding for the relevant box from the VLM image encoder using a L1 reconstruction loss. RegionCLIP (Zhong et al., 2022) uses image-caption data to construct region-wise pseudo-labels, followed by region-text contrastive pre-training before transferring to detection. GLIP and MDETR (Li et al., 2022; Kamath et al., 2021) use captions to cast detection as a phrase grounding task and use early fusion between the image and text modalities, increasing complexity. OVR-CNN (Zareian et al., 2021) uses large image-caption data to pre-train a detector to learn a semantic space and finetunes on smaller detection data. OWL-ViT (Minderer et al., 2022) follows OVR-CNN but makes use of large transformer models and even larger image-caption data. OV-DETR (Zang et al., 2022) modifies the DETR framework for closed-vocabulary object detection (Carion et al., 2020) to make it suitable for the open-vocabulary setting. We note that OV-DETR can condition object detection on image exemplars, but only provides some qualitative examples, whereas we quantitatively benchmark our method using vision-based classifiers. Detic and PromptDet (Zhou et al., 2022; Feng et al., 2022) improve open-vocabulary detection by making use of image classification data to provide weak supervision on a large set of categories. Our work uses Detic as a starting point in experiments, and we investigate different methods for constructing the classifiers.

Low-shot Object Detection. Despite garnering less attention in recent literature, some low/few-shot object detection works make use of image-conditioned object detection (Kang et al., 2019; Hsieh et al., 2019; Osokin et al., 2020; Chen et al., 2021) in which image exemplars of novel categories are encoded at inference time and used to detect novel category instances. These works focus on architectural advances usually leveraging attention between the novel class image exemplars and the inference time image (Chen et al., 2021; Hsieh et al., 2019).

There are an increasing number of low/few-shot object detection works based on finetuning detector parameters on limited groundtruth data for novel categories (Wang et al., 2020; Sun et al., 2021; Qiao et al., 2021; Kaul et al., 2022). These works finetune the detector on limited groundtruth novel instances and so are not related to open-vocabulary object detection using vision-based classifiers, where no novel instances are available for re-training/finetuning.

Natural Language for Classification. Natural language is a rich source of semantic information for classification. CLEVER (Choudhury et al., 2021) matches descriptions of images in simple text form with descriptions from expert databases *e.g.* Wikipedia to perform fine-grained classification. ZSLPP (Elhoseiny et al., 2017) extracts visual information from large-scale text to identify parts of objects and perform zero-shot classification. The work by (Menon & Vondrick, 2023) uses class descriptions from GPT-3 (Brown et al., 2020) for classification, analysing which parts of the description contribute to classification decisions. CuPL (Pratt et al., 2022) uses a GPT-3 model to provide detailed descriptions enabling improved zero-shot image classification. The learnings from this work inform our use of natural language descriptions sourced from LLMs.

3. Method

In this section, we start by providing background on open-vocabulary object detection (OVOD), then outline the proposed methods for constructing classifiers from language descriptions of a category (text-based classifiers, Section 3.2) and image exemplars (vision-based classifiers, Section 3.3). Our final method combines the classifiers found from language descriptions and image exemplars, yielding multi-modal classifiers (Section 3.4).

3.1. Preliminaries

Problem Scenario. Given an image ($\mathbf{I} \in \mathbb{R}^{3\times H\times W}$) fed input to an open-vocabulary object detector, two outputs are generally produced: (1) classification, in which a class label, $c_j \in \mathcal{C}^{\text{TEST}}$, is assigned to the j^{th} predicted object in the image, and $\mathcal{C}^{\text{TEST}}$ refers to the category vocabulary desired at inference time; (2) localisation, with bounding box coordinates, $\mathbf{b}_j \in \mathbb{R}^4$, denoting the location of the j^{th} predicted object. In accordance with the setting introduced by Detic (Zhou et al., 2022), two datasets are used at training time: a detection dataset, $\mathcal{D}^{\text{DET}}$, containing bounding box coordinates, class labels and associated images, addressing a category vocabulary, $\mathcal{C}^{\text{DET}}$; and an image classification dataset, $\mathcal{D}^{\text{IMG}}$, containing images with class labels only, addressing a category vocabulary, $\mathcal{C}^{\text{IMG}}$. In the most general case there are no restrictions on the overlap or lack thereof between the sets $\mathcal{C}^{\text{TEST}}, \mathcal{C}^{\text{DET}}$ and $\mathcal{C}^{\text{IMG}}$.

Architecture Overview. In this work, we make use of a popular multi-stage detector based on CenterNet2 (Zhou et al., 2021) as done in Detic. This detector, with outputs $\{c_j, \mathbf{b}_j\}_{j=1}^{M}$, can be formulated as (for simplicity we consider the two-stage variant below):

$$\{f_j\}_{j=1}^{M} = \Phi_{\text{ROI}} \circ \Phi_{\text{PG}} \circ \Phi_{\text{ENC}}\left(\mathbf{I}\right) \tag{1}$$

$$\{\mathbf{b}_j, c_j\}_{j=1}^{M} = \{\Phi_{\text{BBOX}}\left(f_j\right), \Phi_{\text{CLS}} \circ \Phi_{\text{PROJ}}\left(f_j\right)\}_{j=1}^{M} \tag{2}$$

where, each input image is first sequentially processed by a set of operations: an image encoder (Φ_{ENC}); a proposal generator (Φ_{PG}); a region-of-interest (RoI) feature pooling module (Φ_{ROI}), yielding a set of RoI features, $\{f_j\}_{j=1}^{M}$. The RoI features are processed by a bounding box module (Φ_{BBOX}) to infer position of objects, $\{\mathbf{b}_j\}_{j=1}^{M}$. Additionally, the RoI features are processed by a classification module, consisting of a linear projection (Φ_{PROJ}), and C classification vectors or classifiers (Φ_{CLS}), yielding a set of class labels, $\{c_j\}_{j=1}^{M}$ (C is the size of the category vocabulary).

In closed-vocabulary object detection all parameters listed above are learnt during training on $\mathcal{D}^{\text{DET}}$. While in the open-vocabulary scenario, the classifiers (Φ_{CLS}) *are not* learnt during training but are instead generated separately from an alternative source, *e.g.* a pre-trained text encoder. This allows $\mathcal{C}^{\text{TEST}} \neq \mathcal{C}^{\text{DET}}$, as the classifiers, Φ_{CLS}, for a specific

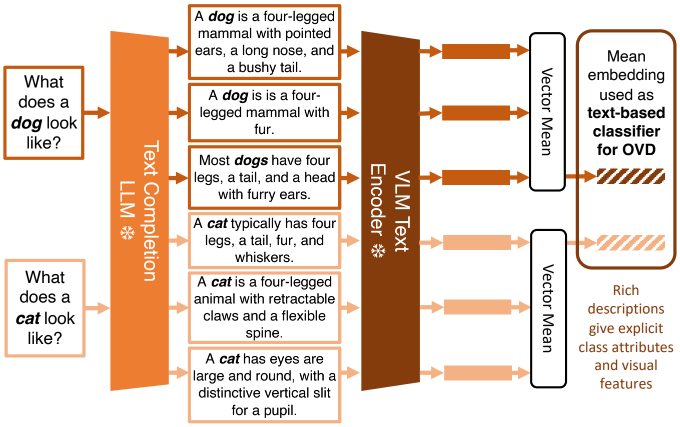
What does a **dog** look like?
What does a **cat** look like?
Text Completion LLM ❄
A **dog** is a four-legged mammal with pointed ears, a long nose, and a bushy tail.
A **dog** is a four-legged mammal with fur.
Most **dogs** have four legs, a tail, and a head with furry ears.
A **cat** typically has four legs, a tail, fur, and whiskers.
A **cat** is a four-legged animal with retractable claws and a flexible spine.
A **cat** has eyes are large and round, with a distinctive vertical slit for a pupil.
VLM Text Encoder ❄
Vector Mean
Vector Mean
Mean embedding used as text-based classifier for OVD
Rich descriptions give explicit class attributes and visual features

Figure 2. Generating powerful text-based classifiers. A LLM (GPT-3) is used to generate multiple rich descriptions of the class of interest. These descriptions are then encoded with the CLIP (Radford et al., 2021) VLM text encoder. The descriptions are more informative than the simple phrases, such as "(a photo of) a **dog**" or "(a photo of) a **cat**", used in previous work such as Detic and ViLD. Additional examples of class descriptions are given in the Appendix (Section F).

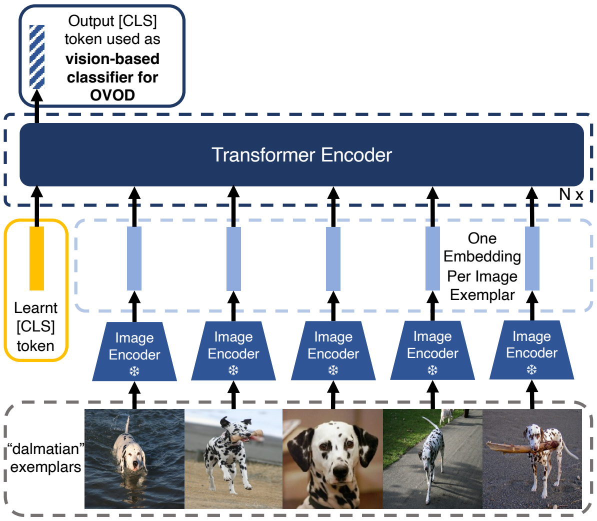

Figure 3. Generating an OVOD vision-based classifier from a set of image exemplars. A stack of transformer blocks is used to combine embeddings of multiple exemplars belonging to the same category.

set of user defined classes can be generated at inference time. In the following sections, we describe different options for constructing such classifiers: from natural language, from image exemplars, or from a combination of the two.

3.2. Text-based Classifiers from Language Descriptions

Existing OVOD approaches, *e.g*. Detic (Zhou et al., 2022) and ViLD (Gu et al., 2022), make use of simple text-based classifiers by encoding the category name with a manual prompt, *e.g*. `a photo of a(n) {class name}` or `a(n) {class name}`, using an appropriate encoder — *e.g*. a CLIP text encoder, thereby yielding a set of classifiers for $\mathcal{C}^{\text{TEST}}$. This method relies on the text encoder to produce a text-based classifier entirely from its internal understanding of `class name`.

Instead, we make use of natural language descriptions of categories sourced from a large language model (LLM). Such a design choice gives additional details like visual attributes, leading to increased discriminative information in the classifier. This alleviates lexical confusion — `class name` may have two different meanings, and effectively prevents the need for human efforts to manually write descriptions or spend time searching external sources for them.

Figure 2 outlines our method for generating informative text-based classifiers. Specifically, we start by prompting an autoregressive language model with a question: "`What does a(n) {class name} look like?`", and sample multiple descriptions per class. We use OpenAI's API for GPT-3 (Brown et al., 2020) and generate 10 descriptions per class with temperature sampling (Figure 2 shows 3 descriptions per class for clarity, more examples can be found in Section F of the Appendix), yielding multiple descriptions of the format `{class name} is a ...` or similar. Given a set of M plain text descriptions $\{s_i^c\}_{i=1}^M$ for class c, we encode each element of the set with a CLIP text encoder (Radford et al., 2021), $f_{\text{CLIP-T}}(\cdot)$, and the text-based classifier for class c is obtained from the mean of these text encodings:

$$\mathbf{w}_{\text{TEXT}}^{c} = \frac{1}{M}\sum_{i=1}^{M} f_{\text{CLIP-T}}\left(s_i^c\right) \tag{3}$$

At detector training time, text-based classifiers for categories of interest ($c \in \mathcal{C}^{\text{DET}}$ and $c \in \mathcal{C}^{\text{IMG}}$) are pre-computed and kept frozen, the rest of detector parameters are updated *i.e*. all parameters in Equations 1-2, except Φ_{CLS}. At inference time, classifiers for testing categories are computed similarly to enable open-vocabulary object detection.

Discussion. In this paper, we only consider a single straightforward question as a prompt to the LLM: "`What does a(n) {class name} look like?`". However, it is also feasible to use alternative question prompts, *e.g*. "`How can you identify a(n) {class name}?`" or "`Describe what a(n) {class name} looks like?`", and obtain visual descriptions with the same or similar concepts (Pratt et al., 2022).

We investigated using a transformer architecture to aggregate text embeddings from natural language descriptions, but we found this was not beneficial to OVOD performance over simply using the mean vector. In general, each text embedding of a natural language description summarises the category of interest very well and so the contrastive task, which is used to train the aggregator (explained in detail for the visual case below), is very easy with text embeddings yielding no improvement in text-based classifiers for OVOD.

3.3. Vision-based Classifiers from Image Exemplars

In addition to constructing classifiers with natural language descriptions, another natural option is to use image exemplars, especially in cases where a good description of the category is prohibitively long (such as the painted lady butterfly or *Vanessa cardui* which has an intricate wing pattern), or occasionally the class name is not known beforehand, *e.g*. "Deerstalker cap" refers to the hat often worn by Sherlock Holmes.

In such scenarios, we propose to construct classifiers by using image exemplars, as shown in Figure 3. Specifically, given a set of K RGB image exemplars for category c, $\{\mathbf{x}_i^c\}_{i=1}^K$, we encode each exemplar with a CLIP visual encoder, $f_{\text{CLIP-IM}}(\cdot)$, yielding K image embeddings, which are then passed to a multi-layer Transformer (Vaswani et al., 2017) with learnable [CLS] token, $\mathbf{t}_{\text{CLS}}$:

$$\mathbf{w}_{\text{IMG}}^c = \text{Transformer}\left(\{f_{\text{CLIP-IM}}\left(\mathbf{x}_i^c\right)\}_{i=1}^K ; \mathbf{t}_{\text{CLS}}\right) \qquad (4)$$

The Transformer architecture acts to best aggregate the K image exemplars, and the output from the [CLS] token is used as the vision-based classifier for OVOD. When training the transformer aggregator, all exemplars are sourced from ImageNet-21k-P (Ridnik et al., 2021). When generating vision-based classifiers for OVOD, where possible, we source our exemplars from ImageNet-21k (Deng et al., 2009) — where this is not possible, we use LVIS and/or VisualGenome training data to source image exemplars. Additional details on how we collate exemplars for training and testing are provided in the Appendix (Section D). This transformer architecture will be referred to as the *visual aggregator* and its training procedure is described next.

Offline Training. The visual aggregator is trained offline *i.e*. it is not updated during detector training. The training procedure needs to learn an aggregator which combines multiple image exemplars to produce effective vision-based classifiers for OVOD — a classifier for a given class needs to be discriminative w.r.t. other classes. A CLIP image encoder is used to provide initial embeddings for each exemplar. We keep the CLIP image encoder frozen during training to improve training efficiency and prevent catastrophic forgetting in the CLIP representation. To provide discriminative vision-based classifiers, contrastive learning is utilised. For a given class, the output embedding from the visual aggregator is trained to minimise similarity with the output embedding from other classes and maximise the similarity with an output embedding from the same class. To do this, the contrastive InfoNCE (van den Oord et al., 2018) loss is used. The visual aggregator should generalise well and not be trained for a specific downstream OVOD vocabulary, therefore it is trained with the ImageNet-21k-P dataset (Ridnik et al., 2021) for image classification, which contains $\sim$11M images across $\sim$11K classes. For category c during visual aggregator training, at each iteration, two distinct sets of K exemplars are sampled, augmented and encoded by the frozen CLIP image encoder. The two sets are input separately to visual aggregator, outputting 2 embeddings from the learnable [CLS] token for class c. Given a batch size B, the InfoNCE contrastive loss ensures sets formed from the same class have similar embeddings and those of different classes are separated. Once trained, the visual aggregator and visual encoder are frozen and provide vision-based classifiers for categories in $\mathcal{C}^{\text{DET}} \cup \mathcal{C}^{\text{IMG}}/\mathcal{C}^{\text{TEST}}$ during detector training/testing. Additional details on how the visual aggregator is trained are provided in the Appendix (Section C).

Discussion. Using image exemplars for open-vocabulary detection may share some similarity to few-shot object detection, however, there is a key distinction. In few-shot object detection, the given "novel/rare" annotations (albeit few) are available for training, *e.g*. recent works have found that fine-tuning a pre-trained object detector on few-shot detection data yields the best results (Wang et al., 2020; Qiao et al., 2021; Kaul et al., 2022), while in open-vocabulary detection, there are no bounding box annotations for "novel/rare" categories. Image exemplars (*i.e*. the wholes image without bounding boxes) are used to specify the categories of interest; we do not update any parameters based on "novel/rare" category bounding box data, unlike in few-shot object detection.

During ablation experiments in Section A, we compare the visual aggregator to a simple mean operator, when obtaining a vision-based classifier from multiple image exemplar embeddings, and show the benefit of the trained aggregator in this case.

3.4. Constructing Classifiers via Multi-Modal Fusion

To go one step further, a natural extension to the aforementioned methods is to construct classifiers from multi-modal cues; intuitively, natural language descriptions and image exemplars may contain complementary information. For a given class, c, with text-based classifier, $\mathbf{w}_{\text{TEXT}}^c$, and vision-based classifier, $\mathbf{w}_{\text{IMG}}^c$, the multi-modal classifier, $\mathbf{w}_{\text{MM}}^c$, is computed by a simple fusion method based on addition:

$$\mathbf{w}_{\text{MM}}^c = \frac{\mathbf{w}_{\text{TEXT}}^c}{\|\mathbf{w}_{\text{TEXT}}^c\|_2} + \frac{\mathbf{w}_{\text{IMG}}^c}{\|\mathbf{w}_{\text{IMG}}^c\|_2} \qquad (5)$$

Figure 1 demonstrates our entire pipeline for generating text-based, vision-based and multi-modal classifiers, showing how any of the three classifiers can be used with an open-vocabulary detector to detect a "falcon".

Discussion. Section 3.3 provides details of the visual aggregator, which yields our vision-based classifiers, but for multi-modal classifiers we simply compute the vector sum of our l^2-normalised text-based and vision-based classifiers. We investigated using a unified multi-modal aggregator which ingests both text and visual embeddings, sourced from class descriptions and image exemplars, respectively. Such a model did not generate good multi-modal classifiers for OVOD — distinguishing between sets of text and image embeddings for different classes becomes trivial as the text embeddings alone are sufficient to solve the contrastive learning task, thereby ignoring the visual embeddings altogether. Attempts to modify the training for a unified multi-modal aggregator by using Dropout (Srivastava et al., 2014) on the text embeddings were not fruitful.

4. Experiments

In this section, we first introduce the standard dataset and benchmark used in the literature (Gu et al., 2022; Zhou et al., 2022; Feng et al., 2022). Section 4.2 provides implementation and training details for our OVOD models, which use classifiers constructed from natural language descriptions, visual exemplars or the combination of both. We compare our models with existing works in Section 4.3, demonstrating state-of-the-art performance. Additionally, Section 4.3 provides results for cross-dataset transfer. Section 4.4 provides an ablation study regarding our design choices.

4.1. Datasets and Evaluation Protocol

Standard LVIS Benchmark. In this work, most experiments are based on the LVIS object detection dataset (Gupta et al., 2019), containing a large vocabulary and a long-tailed distribution of object instances. Specifically, the LVIS dataset contains class, bounding box and mask annotations for 1203 classes across 100k images in the MS-COCO dataset (Lin et al., 2014). Annotations are collected in a federated manner, *i.e.* manual annotations for a given image are not necessarily exhaustive. The classes are divided into three sets — rare, common and frequent — based on the number of training images containing a given class.

Training Datasets. To develop open-vocabulary object detectors, we follow the same setting as proposed in ViLD (Gu et al., 2022) and used in Detic (Zhou et al., 2022). Specifically, the original LVIS training set (*LVIS-all*) is slightly changed by removing all annotations belonging to the "rare" categories. This removes the annotations of 317 rare classes *but not* the associated images, in other words, objects belonging to rare categories appear in the training set but are unannotated. This subset of LVIS training data containing only "common" and "frequent" annotations is referred to as *LVIS-base*. LVIS-base serves as $\mathcal{D}^{\text{DET}}$ using notation from Section 3.1, unless stated otherwise. When using image classification data, $\mathcal{D}^{\text{IMG}}$, as extra weak supervision, we use the subset of categories in ImageNet-21K (Deng et al., 2009) that overlap with the LVIS vocabulary and denote this subset as *IN-L*, as in Detic. IN-L covers 997 of the 1203 classes in the LVIS vocabulary.

Evaluation Protocol. For evaluation, previous work evaluates OVOD models on the LVIS validation set (*LVIS-val*) for all categories — treating "rare" classes as novel categories as it is guaranteed that no groundtruth box annotations whatsoever are provided at the training stage. The main evaluation metric is the standard mask AP metric averaged over the "rare" classes and is denoted as APr. The mask AP averaged across *all* classes is also reported, indicating overall class performance and is denoted as mAP. The latter metric is an important consideration as a good model should improve both APr and mAP; a model should not improve APr at the cost of worse performance in terms of mAP.

4.2. Implementation Details

Object Detector Architecture. The architecture we use is almost identical to that in Detic, using the CenterNet2 (Zhou et al., 2021) model with a ResNet-50 backbone (He et al., 2016) pre-trained on ImageNet-21k-P (Ridnik et al., 2021). In addition to exploring different ways for constructing classifiers (Φ_{CLS}), as described in Section 3, we also add a learnable bias before generating final confidence scores, the effect of this bias term is investigated in the Appendix (Section A).

Detector Training. The training recipe is the same as Detic for fair comparison, using Federated Loss (Zhou et al., 2021) and repeat factor sampling (Gupta et al., 2019). While training our OVOD model on detection data only, $\mathcal{D}^{\text{DET}}$, we use a $4\times$ schedule ($\sim$58 *LVIS-base* epochs or 90k iterations with batch size of 64). When using additional image-labelled data (IN-L), we train jointly on $\mathcal{D}^{\text{DET}} \cup \mathcal{D}^{\text{IMG}}$ using a $4\times$ schedule (90k iterations) with a sampling ratio of $1:4$ and batch sizes of 64 and 256, respectively. This results in $\sim$15 *IN-L* epochs and an additional $\sim$11 *LVIS-base* epochs. For mini-batches containing images from $\mathcal{D}^{\text{DET}}$ and $\mathcal{D}^{\text{IMG}}$ we use input resolutions of 640^2 and 320^2, respectively. We conduct our experiments on 4 32GB V100 GPUs.

For image-labelled data, $\mathcal{D}^{\text{IMG}}$, an image with class label is given, but no groundtruth bounding box is available. Following Detic, the largest class-agnostic box proposal is used to produce an RoI feature for the given image, enabling detector training. See the Detic paper for more details.

Text-based Classifier Construction. To generate plain text class descriptions we use the GPT-3 DaVinci-002 model available from OpenAI. For each class in LVIS, we generate 10 descriptions and compute the classifier with the text encoder from a CLIP ViT-B/32 model (Radford et al., 2021), as detailed in Section 3.2. We follow the standard method from CLIP and use the output embedding corresponding to the final token in the input text.

Vision-based Classifier Construction. The visual aggregator, detailed in Section 3.3, should be general and not specific to any class vocabulary. To fulfil this goal, we use the curated ImageNet-21-P dataset (Ridnik et al., 2021) as training data *for the aggregator*. This dataset, designed for pre-training visual backbones, filters out classes with few examples from the original ImageNet-21k (Deng et al., 2009) dataset, leaving ∼11M images across ∼11K classes.

To generate a visual embedding from a given image exemplar, we use a CLIP ViT-B/32 visual encoder. For the aggregator, we use $N = 4$ transformer blocks, with dimension 512 (the same as the output dimension of the CLIP visual encoder) and a multilayer perceptron dimension of 2048. Comprehensive details of aggregator training is provided in the Appendix (Section C). To test the effectiveness of our visual aggregator, we generate baseline vision-based classifiers by taking the vector mean of the CLIP visual embeddings from the K image exemplars.

When constructing vision-based classifiers for OVOD, we find making use of test-time augmentation (TTA) improves performance. Note, TTA here refers to augmentation of the image exemplars used to build vision-based classifiers *not* test-time augmentation of the test image on which OVOD is performed. In our work, each image exemplar is augmented 5 times and input separately to the visual encoder. Therefore, given K image exemplars, we generate $5K$ visual embeddings to be ingested by our visual aggregator. More details on the use of TTA in constructing vision-based classifiers are found in the Appendix (Section A).

For total clarity regarding when datasets are used — the visual aggregator is *trained* using ImageNet-21k-P (Ridnik et al., 2021) and the vision-based classifiers for OVOD are generated from relevant image exemplars *using the trained aggregator* (see Section D of the Appendix for details on sourcing the image exemplars). Detection data (*e.g. LVIS-base*) is only used to train the open-vocabulary object detector and image-level data (*e.g. IN-L*) may be used as an extra source of weak supervision as in Detic (Zhou et al., 2022).

Multi-Modal Classifier Construction. When computing multi-modal classifiers, we simply compute the category-wise l^2-normalised classifier from each modality, regardless of method used to compute them, and take the vector sum. In all cases the text-based classifiers are sourced from class descriptions as described in Section 3.2. We combine our text-based classifiers with our vision-based classifiers, using the trained visual aggregator. However, once again, to test the effectiveness of our visual aggregator, we also combine our text-based classifiers with the baseline vision-based classifiers described in the previous paragraph.

Model	Backbone	Extra Data	APr	mAP
ViLD (Gu et al., 2022)	ResNet-50	✗	16.1	22.5
Detic (Zhou et al., 2022)	ResNet-50		16.3	30.0
ViLD-ens (Gu et al., 2022)	ResNet-50		16.6	25.5
OV-DETR (Zang et al., 2022)	ResNet-50 + DETR		17.4	26.6
F-VLM (Kuo et al., 2022)	ResNet-50		*18.6*	24.2
Ours (Text-Based)	ResNet-50	✗	**19.3**	*30.3*
Ours (Vision-Based)			18.3	29.2
Ours (Multi-Modal)			**19.3**	**30.6**
RegCLIP (Zhong et al., 2022)	ResNet-50	CC3M	17.1	28.2
OWL-ViT (Minderer et al., 2022)†	ViT-B/32	LiT	19.7	23.3
Detic (Zhou et al., 2022)	ResNet-50	IN-L	24.6	32.4
Ours (Text-Based)	ResNet-50	IN-L	*25.8*	*32.7*
Ours (Vision-Based)			23.8	31.3
Ours (Multi-Modal)			**27.3**	**33.1**
Fully-Supervised (Zhou et al., 2022)	ResNet-50	✗	25.5	31.1

Table 1. Detection performance on the LVIS Open Vocabulary Detection Benchmark using our three types of classifier compared with previous works. Best and second-best performing models are coloured **blue** and *red*, respectively. We split models into those which only use LVIS-base as training data (top) and those which use additional image-level data (bottom). Furthermore, we show results for a fully-supervised model from Detic trained on LVIS-all in grey. † OWL-ViT reports bbox AP metrics and was trained on Objects365 (Shao et al., 2019) and VisualGenome (Krishna et al., 2017) *not* LVIS-base, therefore it is possible LVIS-defined "rare" classes are contained in the detection training data of OWL-ViT. Due to limited compute resources we present and compare to models which use ResNet-50 (He et al., 2016) backbones or similar. We report mask AP metrics except for †.

4.3. Open-Vocabulary Detection Results

LVIS OVOD Benchmark. Table 1 shows results on *LVIS-val* for our work, which uses text-based, vision-based and multi-modal classifiers, compared to a range of prior work. We report overall mask AP performance and mask AP for "rare" classes only. The latter metric is the key measure of OVOD performance. We separate the comparisons into those models which do not use additional image-level data (top half of Table 1) and those which do (bottom half of Table 1). For a fair evaluation, we compare to models from prior works which use a ResNet-50 (He et al., 2016) backbone. There are two exceptions: (1) OWL-ViT (Minderer et al., 2022) which only investigates using Vision Transformers for OVOD (Dosovitskiy et al., 2021) – we compare ResNet-50 models to the ViT-B/32 OWL-ViT model as it requires similar compute during inference in terms of GLOPs (141.5 and 139.6, respectively); (2) OV-DETR (Zang et al., 2022) which uses a DETR-style architecture (Carion et al., 2020) consisting of a ResNet-50 CNN backbone and modified transformer encoder and decoder.

In the experiments without using extra data ($\mathcal{D}^{\text{IMG}} = \varnothing$), our models with text-based or multi-modal classifiers obtain the best performance on both APr and overall mAP, while F-VLM and Detic only performs strongly on APr and mAP,

Model	Extra Data	Objects365 mAP	Objects365 AP50	Objects365 APr
Detic (Zhou et al., 2022)	✗	13.9	19.7	9.5
Ours (Text-Based)		**14.8**	**21.0**	**10.1**
Detic (Zhou et al., 2022)	IN-L	15.6	22.2	12.4
Ours (Text-Based)		**16.6**	**23.1**	**13.1**

Table 2. Detection performance when training on *LVIS-all* (*i.e.* all LVIS training data) and evaluating on Objects365 (Shao et al., 2019), where the least frequent $\frac{1}{3}$ of classes are defined as "rare". Best performing models are coloured **blue**. Our text-based classifiers outperform Detic when transferring to Objects365 across all classes and "rare" classes only. Note the Detic (Zhou et al., 2022) models we compare to on Objects365 are not the same as the models listed in the Detic paper, which use a large Swin-B backbone and all of ImageNet-21k as extra data for weak supervision. We use a fair comparison Detic model which uses the same training data (see text for more details). We report box AP metrics for Objects365.

respectively. Our model with text-based classifiers is most directly comparable to Detic and our model outperforms Detic by 3.0 APr. When using extra data, we make use of a ImageNet-21k subset as in Detic (IN-L). Models with text-based and multi-modal classifiers outperform Detic (previous state-of-the-art) by 1.2 and 2.7 APr respectively. **Note that**, our models trained with IN-L even outperform the fully-supervised baseline using CenterNet2, *i.e.* trained on "rare" class box annotations. To the best of our knowledge, this is the first work on open-vocabulary detection that outperforms a comparable fully-supervised model on the challenging LVIS benchmark. Note, some other works use larger vision backbones, *e.g.* Swin-B (Liu et al., 2021), but due to limited computation resources we only present and compare to models with ResNet-50 backbones or similar. Our models with text-based and multi-modal classifiers surpass state-of-the-art performance when using a ResNet-50 backbone or similar.

Cross-dataset Transfer. Table 2 shows results for cross-dataset transfer from LVIS to Objects365 (Shao et al., 2019) when using our text-based classifiers. We compare our work to equivalent models from Detic, reporting box AP metrics as standard in Objects365. In all cases, these models are trained on *LVIS-all* and the models in the bottom two rows use *IN-L* as extra weak supervision. The trained open-vocabulary detectors are evaluated on the Objects365 validation set. Following Detic, we define "rare" classes in Objects365 as the $\frac{1}{3}$ of classes with the lowest frequency in the Objects365 training set. Note the Detic models we compare to in Table 2 are not the same as those listed in Table 4 of the Detic paper, which use a large Swin-B backbone and all of ImageNet-21k as extra data. Instead, we compare to other Detic models publicly available which are trained on *LVIS-all* (and *IN-L*). For ease, we provide links to the Detic configuration and checkpoint files used in this cross-dataset transfer evaluation [1] [2]. Evaluation on Objects365 after training on LVIS is easily done with Detic or with our text-based classifiers. In the case of Detic, the simple classifiers based on LVIS class names are replaced with equivalent simple classifiers based on Objects365 class names. For our text-based classifiers, plain text descriptions are generated for each of the Objects365 classes and are encoded as described in Section 3.2 and replace the text-based classifiers for LVIS. In both models, Detic and ours, all other parameters of the open-vocabulary detector remain the same. In all cases, using our text-based classifiers gives performance improvements over the equivalent Detic model. Considering all classes, our method with extra data outperforms Detic by 1.0 mAP and 0.9 AP50. For "rare" classes as defined above, our method with extra data outperforms the equivalent Detic model by 0.7 APr. These results demonstrate that our method, which uses text-based classifiers generated from rich class descriptions, provides additional information compared to using a simple classifier based on the class name only, even when the training and testing class vocabularies are disjoint.

4.4. Ablation Study

Results with Vision-Based Classifiers. Using vision-based classifiers for OVOD is an under-explored area and so we compare our method, detailed in Section 3.3, to baseline classifiers in which the same visual encoder is used, but the action of the aggregator is replaced by performing simple vector mean. The orange rows of Table 3 compares the use of our visual aggregator to performing simple vector mean across visual embeddings instead. When no additional image-level data is used (top two orange rows) our aggregator (Model B) boosts performance by 3.5 APr compared to the vector mean baseline (Model A). For models which train on additional image-level data (IN-L) our aggregator (Model G) boosts performance by 2.2 APr compared to the baseline (Model F). This comparison demonstrates the utility of our visual aggregator in constructing better vision-based classifiers rather than naïvely averaging the K visual embeddings. Note our vision-based classifiers and the baseline classifiers both utilise TTA as mentioned in Section 4.2 (see Section A of the Appendix for details on TTA). The results in Table 3 use $K = 5$. Further results for $K = 1, 2, 10$ are found in the Appendix (Section E).

Results with Multi-Modal Classifiers. To evaluate the effectiveness of multi-modal classifiers we perform similar experiments as those using vision-based classifiers, except for each model we combine the vision-based classifiers with the text-based classifiers, as described in Section 3.4. Results for multi-modal classifiers are shown in grey rows of Table 3. With no additional image-level data, the vector

[1] Detic Configuration Checkpoint Extra Data: ✗

[2] Detic Configuration Checkpoint Extra Data: *IN-L*

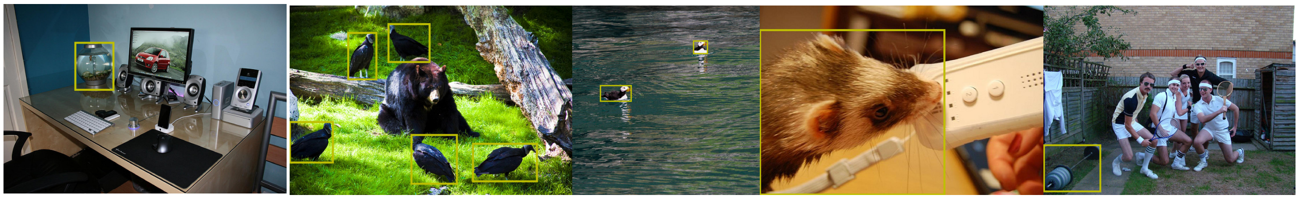

Figure 4. Some qualitative detection examples using our model with text-based classifiers, detecting "rare" category instances in *LVIS-val*. Our text-based classifiers are sourced from rich natural language descriptions of a given class by prompting an GPT-3 LLM.

Model	Visual Mean?	Visual Agg.?	Text Cls.?	Extra Data?	APr	mAP
A	✓			✗	14.8	28.8
B		✓			18.3	29.2
C			✓	✗	*19.3*	30.3
D	✓		✓	✗	**20.7**	*30.5*
E		✓	✓		*19.3*	**30.6**
F	✓			IN-L	21.6	31.3
G		✓			23.8	31.3
H			✓	IN-L	25.8	32.7
I	✓		✓	IN-L	*26.5*	*32.8*
J		✓	✓		**27.3**	**33.1**

Table 3. Detection performance on the LVIS OVOD benchmark comparing all three of our methods: (1) orange — vision-based classifiers; (2) blue — text-based classifiers; (3) grey — multi-modal classifiers. Results for models trained only on LVIS-base and LVIS-base+IN-L are shown in the top and bottom halves, respectively. Visual Mean?: *simple vector mean* is used to combine visual embeddings of image exemplars, Visual Agg.?: *our visual aggregator* is used to combine visual embeddings, Text Cls.?: text-based classifiers are used. Models which use text-based and vision-based classifiers represent our models with multi-modal classifiers. We report mask AP metrics.

mean baseline (Model D) outperforms the use of our aggregator (Model E) by 1.4 APr. However, for models with image-level data (IN-L) our aggregator (Model J) boosts performance by 0.8 APr compared to the baseline (Model I). Furthermore, comparing the multi-modal classifiers (grey rows in Table 3) with text-based classifiers (blue rows in Table 3) demonstrates that in all cases adding information from image exemplars yields improved OVOD performance — our best multi-modal model improves performance over our best text-based model by 1.5 APr confirming that combining the vision and text modalities utilises complementary information between the two.

Relationship between IN-L and LVIS "rare" classes. Section B of the Appendix splits the APr metric into two based on the "rare" LVIS categories contained in IN-L. One may expect improvements in APr performance when training on IN-L to only come from "rare" categories found in IN-L. Our evaluation finds this not to be the case. Detailed results can be found in the Appendix (Section B) which breaks the APr metric into "rare" categories found in IN-L and those not.

Additional Ablation Experiments. Section A of the Appendix presents ablation experiments which demonstrate: (1) applying a learnable bias before calculating the final detection score for a region improves OVOD performance; (2) improvements in OVOD performance using our text-based classifiers is orthogonal to applying this learnable bias; (3) applying TTA on image exemplars yield better vision-based classifiers for OVOD; (4) comparisons between our text-based classifiers and those generated from manual prompts. Please refer to Section A of the Appendix for details and evaluation results for these experiments.

5. Conclusion

In this paper, we tackle open-vocabulary object detection by investigating the importance of the method used to generate classifiers for open-vocabulary detection. This work goes beyond the very simple methods used in prior work to generate such classifiers — with the class name only. We present a novel method which combines a large language model (LLM) and a visual-language model (VLM) to produce improved classifiers. Moreover, we investigate using image exemplars to provide classifiers for OVOD and present a method for generating such classifiers using a large classification dataset and a simple transformer based architecture. Finally, we combine our classifiers from the two modalities to produce multi-modal classifiers for OVOD. Our experiments show that our method using natural language only outperforms current state-of-the-art OVOD works, especially in cases where no extra image-level data is used. Furthermore, our multi-modal classifiers set new state-of-the-art performance with a large improvement over prior work.

Acknowledgements

We thank Lauren Bates-Brownsword for help with proofreading. This research was supported by the EPSRC CDT in AIMS EP/L015897/1 and EP/S515541/1, the EPSRC Programme Grant VisualAI EP/T028572/1, and a Royal Society Research Professorship. Weidi Xie would like to acknowledge the National Key R&D Program of China (No. 2022ZD0161400).

References

Bansal, A., Sikka, K., Sharma, G., Chellappa, R., and Divakaran, A. Zero-shot object detection. In *Proceedings of the European Conference on Computer Vision*, September 2018.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, 2020.

Cai, Z. and Vasconcelos, N. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6154–6162. IEEE Computer Society, 2018.

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., and Zagoruyko, S. End-to-end object detection with transformers. In *Proceedings of the European Conference on Computer Vision*, pp. 213–229. Springer, 2020.

Chen, D.-J., Hsieh, H.-Y., and Liu, T.-L. Adaptive image transformer for one-shot object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 12242–12251, 2021.

Chen, K., Pang, J., Wang, J., Xiong, Y., Li, X., Sun, S., Feng, W., Liu, Z., Shi, J., Ouyang, W., et al. Hybrid task cascade for instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4974–4983, 2019.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. In *Proceedings of the International Conference on Machine Learning*, 2020.

Choudhury, S., Laina, I., Rupprecht, C., and Vedaldi, A. The curious layperson: Fine-grained image recognition without expert labels. In *Proceedings of the British Machine Vision Conference*, 2021.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale. *Proceedings of the International Conference on Learning Representations*, 2021.

Elhoseiny, M., Zhu, Y., Zhang, H., and Elgammal, A. Link the head to the "beak": Zero shot learning from noisy text description at part precision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

Feng, C., Zhong, Y., Jie, Z., Chu, X., Ren, H., Wei, X., Xie, W., and Ma, L. Promptdet: Towards open-vocabulary detection using uncurated images. 2022.

Girshick, R. B. Fast R-CNN. In *Proceedings of the International Conference on Computer Vision*, 2015.

Gu, X., Lin, T.-Y., Kuo, W., and Cui, Y. Open-vocabulary detection via vision and language knowledge distillation. In *Proceedings of the International Conference on Learning Representations*, 2022.

Gupta, A., Dollar, P., and Girshick, R. LVIS: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

Hsieh, T.-I., Lo, Y.-C., Chen, H.-T., and Liu, T.-L. One-shot object detection with co-attention and co-excitation. In *Advances in Neural Information Processing Systems*. 2019.

Jia, C., Yang, Y., Xia, Y., Chen, Y.-T., Parekh, Z., Pham, H., Le, Q., Sung, Y.-H., Li, Z., and Duerig, T. Scaling up visual and vision-language representation learning with noisy text supervision. In *Proceedings of the International Conference on Machine Learning*, pp. 4904–4916, 2021.

Kamath, A., Singh, M., LeCun, Y., Synnaeve, G., Misra, I., and Carion, N. Mdetr-modulated detection for end-to-end multimodal understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1780–1790, 2021.

Kang, B., Liu, Z., Wang, X., Yu, F., Feng, J., and Darrell, T. Few-shot object detection via feature reweighting. In *Proceedings of the International Conference on Computer Vision*, pp. 8420–8429, 2019.

Kaul, P., Xie, W., and Zisserman, A. Label, verify, correct: A simple few-shot object detection method. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2022.

Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.-J., Shamma, D. A., et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123(1):32–73, 2017.

Kuo, W., Cui, Y., Gu, X., Piergiovanni, A., and Angelova, A. F-vlm: Open-vocabulary object detection upon frozen vision and language models. *arXiv preprint arXiv:2209.15639*, 2022.

Law, H. and Deng, J. Cornernet: Detecting objects as paired keypoints. In *Proceedings of the European Conference on Computer Vision*, 2018.

Li, L. H., Zhang*, P., Zhang*, H., Yang, J., Li, C., Zhong, Y., Wang, L., Yuan, L., Zhang, L., Hwang, J.-N., Chang, K.-W., and Gao, J. Grounded language-image pre-training. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2022.

Li, Y., Chen, Y., Wang, N., and Zhang, Z. Scale-aware trident networks for object detection. In *Proceedings of the International Conference on Computer Vision*, 2019.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft coco: Common objects in context. In *Proceedings of the European Conference on Computer Vision*, 2014.

Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollár, P. Focal loss for dense object detection. In *Proceedings of the International Conference on Computer Vision*, 2017.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., and Berg, A. C. Ssd: Single shot multibox detector. In *Proceedings of the European Conference on Computer Vision*, pp. 21–37. Springer, 2016.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the International Conference on Computer Vision*, pp. 10012–10022, October 2021.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. In *Proceedings of the International Conference on Learning Representations*, 2018.

Menon, S. and Vondrick, C. Visual classification via description from large language models. *Proceedings of the International Conference on Learning Representations*, 2023.

Miller, G. A. Wordnet: A lexical database for english. *Communications of the Association for Computing Machinery*, 38(11): 39–41, nov 1995.

Minderer, M., Gritsenko, A., Stone, A., Neumann, M., Weissenborn, D., Dosovitskiy, A., Mahendran, A., Arnab, A., Dehghani, M., Shen, Z., Wang, X., Zhai, X., Kipf, T., and Houlsby, N. Simple open-vocabulary object detection. In *Proceedings of the European Conference on Computer Vision*, pp. 728–755, 2022.

Osokin, A., Sumin, D., and Lomakin, V. OS2D: One-stage one-shot object detection by matching anchor features. In *Proceedings of the European Conference on Computer Vision*, 2020.

Pratt, S., Liu, R., and Farhadi, A. What does a platypus look like? generating customized prompts for zero-shot image classification. *arXiv preprint arXiv:2209.03320*, 2022.

Qiao, L., Zhao, Y., Li, Z., Qiu, X., Wu, J., and Zhang, C. Defrcn: Decoupled faster r-cnn for few-shot object detection. In *Proceedings of the International Conference on Computer Vision*, pp. 8681–8690, October 2021.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *Proceedings of the International Conference on Machine Learning*, pp. 8748–8763, 2021.

Redmon, J. and Farhadi, A. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

Redmon, J., Divvala, S. K., Girshick, R. B., and Farhadi, A. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

Ren, S., He, K., Girshick, R., and Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, 2016.

Ridnik, T., Ben-Baruch, E., Noy, A., and Zelnik, L. Imagenet-21k pretraining for the masses. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, 2021.

Shao, S., Li, Z., Zhang, T., Peng, C., Yu, G., Zhang, X., Li, J., and Sun, J. Objects365: A large-scale, high-quality dataset for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8429–8438, 2019. doi: 10.1109/ICCV.2019.00852.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. In *Journal of Machine Learning Research*, 2014.

Sun, B., Li, B., Cai, S., Yuan, Y., and Zhang, C. Fsce: Few-shot object detection via contrastive proposal encoding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2021.

Tan, M., Pang, R., and Le, Q. V. Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

Tian, Z., Shen, C., Chen, H., and He, T. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the International Conference on Computer Vision*, 2019.

van den Oord, A., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need. In *NIPS*, 2017.

Wang, X., Huang, T. E., Darrell, T., Gonzalez, J. E., and Yu, F. Frustratingly simple few-shot object detection. In *Proceedings of the International Conference on Machine Learning*, 2020.

Zang, Y., Li, W., Zhou, K., Huang, C., and Loy, C. C. Open-vocabulary detr with conditional matching. 2022.

Zareian, A., Rosa, K. D., Hu, D. H., and Chang, S.-F. Open-vocabulary object detection using captions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 14393–14402, 2021.

Zhong, Y., Yang, J., Zhang, P., Li, C., Codella, N., Li, L. H., Zhou, L., Dai, X., Yuan, L., Li, Y., and Gao, J. Regionclip: Region-based language-image pretraining. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 16793–16803, 2022.

Zhou, X., Wang, D., and Krähenbühl, P. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019.

Zhou, X., Koltun, V., and Krähenbühl, P. Probabilistic two-stage detection. *arXiv preprint arXiv:2103.07461*, 2021.

Zhou, X., Girdhar, R., Joulin, A., Krähenbühl, P., and Misra, I. Detecting twenty-thousand classes using image-level supervision. In *Proceedings of the European Conference on Computer Vision*, pp. 350–368, 2022.

A. Ablation Studies

We now ablate some of the key components using the open-vocabulary LVIS benchmark without any extra image classification data *i.e.* $\mathcal{D}^{\text{IMG}} = \varnothing$, unless stated otherwise. All metrics have the standard LVIS definition and we report mask AP metrics in all cases. For reference APr, APc and APf represent mean average precision across "rare", "common" and "frequent" classes, respectively, as defined in LVIS. Moreover, mAP, AP50 and AP75 represent mean average precision across all classes but for all intersection-over-union (IoU) criteria, IoU= 0.5 and IoU= 0.75, respectively.

Bias?	Init. Value	mAP	AP50	AP75	APr	APc	APf
✗	N/A	29.7	43.7	31.6	15.6	30.5	35.0
✓	-2.0	29.9	43.5	32.0	17.7	30.1	35.0

Table 4. The effect of including a learnable bias on detections scores for OVOD. The top row does not use a learnable bias, as in Detic. The bottom row applies a learnable bias prior to computing final detection scores with logistic sigmoid. Applying a learnable bias improves performance on novel/rare categories (APr). We report mask AP metrics.

Effect of Detection Score Bias. Table 4 shows the effect of adding a learnable bias to the detection scores before applying a logistic sigmoid to get a final detection score in the range $[0, 1]$ To evaluate the effect of the learnable bias only, our proposed text-based classifiers sourced from rich class descriptions, as described in Section 3.2, *are not* used and instead the same simple text-based classifiers used in Detic, of form "`a(n) {class name}`", are used in this comparison. We observe adding a learnable bias improves open-vocabulary detection by 2.1 AP on rare categories compared to not using a bias, as done in Detic. Without the use of a bias, class-agnostic proposals are not biased towards being labelled as background. With respect to a given class, a proposal is most likely to be negative, therefore use of a bias makes intuitive sense to reflect this and stabilises early training of the detector. Similar findings were found in RetinaNet (Lin et al., 2017).

Model	mAP	AP50	AP75	APr	APc	APf
Detic	30.2	44.2	32.1	16.4	31.0	35.4
Ours (w/o bias)	30.4	44.4	32.3	18.6	30.8	35.2
Ours (w/ bias)	30.3	44.2	32.2	19.3	30.5	35.0

Table 5. The effect of using our text-based classifiers sourced from rich descriptions. In contrast, Detic uses simple classifiers based on class names only (top row). Results for a detector trained using our text-based classifiers but no learnable bias is shown in the middle row. Our proposed model makes use of text-based classifiers sourced from rich descriptions and a learnable bias (bottom row). Our method for text-based classifiers improves performance on novel/rare categories (APr) by a large amount. We report mask AP metrics.

Natural Language Descriptions. Table 5 shows the effect of using rich class descriptions, sourced from a LLM, rather than forming text-based classifiers from simple text prompts of the format "`a(n) {class name}`" as in Detic. To compare fairly to Detic with detection data only (top row), we report a set of results which do not make use of the learnable bias on the detection scores as detailed above (middle row). Using our text-based classifiers without a learnable bias improves performance on rare categories by 2.2 APr compared to the public Detic model. Using rich class descriptions, a learnable bias and our method (bottom row) further improves open-vocabulary detection on novel/rare categories by 2.9 and 0.7 APr compared to the public Detic model and our method without a learnable bias, respectively.

Visual Encoder	TTA?	mAP	APr	APc	APf
CLIP ViT-B/32	✗	29.0	16.3	29.1	34.4
	✓, harsh	29.0	17.2	28.7	34.6
	✓, gentle	29.2	18.3	28.7	34.4

Table 6. The effect of using test-time augmentation (TTA) when generating classifier embeddings from image exemplars using a CLIP image encoder. Both TTA recipes use two common augmentations — `ColorJitter` and `RandomHorizontalFlip`. For harsh/gentle TTA — min scale of `RandomResizedCrop` $= 0.5/0.8$. Both harsh and gentle TTA perform better than no TTA in terms of performance on novel/rare categories (APr). We report mask AP metrics.

Test-Time Augmentation on Image Exemplars for Vision-Based Classifiers. Table 6 shows the effect of using test-time augmentation (TTA) on image exemplars to produce vision-based classifiers with our trained aggregator. For each image exemplar we generate 5 augmentations. As in the main paper, we use the case of $K = 5$ — for each class in the LVIS vocabulary we are have 5 RGB image exemplars. As mentioned in Section 4.2, we augment each exemplar 5 times when using TTA. We consider two augmentation variations, with each containing `ColorJitter` and `RandomHorizontalFlip`. The 'harsh' variation uses `RandomResizedCrop(scale=(0.5,1.0))` and the 'gentle' variation uses `RandomResizedCrop(scale=(0.8,1.0))`. We find adding 'gentle' TTA performs best, improving open-vocabulary detection by 2.0 AP on rare categories compared to no use of TTA. In the main paper, when using vision-based classifiers we utilise 'gentle' TTA on the image exemplars.

Prompt	mAP	APr	APc	APf
a/an `class name`	29.9	17.7	30.1	35.0
a photo of a/an `class name`	29.5	15.9	30.0	34.9
a photo of a/an `class name` in the scene	29.4	16.2	29.7	34.8
Our LLM descriptions	30.3	19.3	30.5	35.0

Table 7. The effect of using manually crafted prompts against our rich class descriptions sourced from LLMs. All models use a learnable bias on the detection scores and text-based classifiers. Using our class descriptions improves performance on "rare" classes compared to manually crafted prompts.

Comparing our LLM Descriptions to Manually Designed Text Prompts. Table 7 compares the detector performance between using simple manually crafted prompts (first three rows) and our rich class descriptions sourced from an LLM (final row), for constructing text-based classifiers. We report results for the case where the detector is trained on *LVIS-base* only (*i.e.* no additional image-level data is used). In all cases we apply the learnable bias on the detection score. We note that of the manual prompts, the simplest one, of the form "`a(n) {class name}`", performs best across all metrics. Using our text-based classifiers generated from LLM descriptions improves performance on rare classes by 1.6 APr and by 0.4 mAP across all classes. Performance on "common" and "frequent" classes is largely similar as the availability of labelled detection data for these classes renders the quality of the classifier less important.

B. A Closer Look at "rare" Class Performance

Model	$\mathcal{D}^{\text{IMG}}$	mAP	APr	APr-w	APr-z
Detic (Zhou et al., 2022)	✗	30.2	16.3	15.7	19.7
Ours (Text-Based)		30.3	**19.3**	**19.2**	19.4
Ours (Multi-Modal)		**30.6**	19.2	18.5	**22.2**
Detic (Zhou et al., 2022)	✓	32.4	24.9	25.4	23.0
Ours (Text-Based)		32.6	25.8	26.7	21.7
Ours (Multi-Modal)		**33.1**	**27.3**	**27.8**	**24.9**

Table 8. Comparison between Detic and our models using text-based classifiers and multi-modal classifiers on LVIS. As IN-L does not cover all classes in the LVIS vocabulary, we split the rare class metric APr into APr-w and APr-z which represent rare classes with and without weak annotations from IN-L, respectively. Best performing models are shown in **bold**. We report mask AP metrics.

Section 4.1 detailed the data used to train our detector. IN-L contains images from the 997 classes from LVIS present in ImageNet-21k (Deng et al., 2009).

IN-L gives weak supervision during detector training. Out of the 337 "rare" classes in LVIS, 277 are covered by IN-L and are therefore weakly supervised, leaving 60 classes for which no weak supervision is available.

To investigate the improvement in performance when using IN-L, we split the rare class metric (APr), which reports average precision across rare classes, into APr-w which averages across rare classes *present* in IN-L and APr-z which averages across rare classes *not present* in IN-L which are therefore truly zero-shot classes. Note that when no extra image-level data is used, *i.e.* $\mathcal{D}^{\text{IMG}} = \varnothing$, all rare classes are truly zero-shot classes.

Table 8 shows the result of using this breakdown. These results show training on IN-L improves performance on rare classes not contained in IN-L, which may not be expected. The weak supervision from IN-L leads to a reduction in false positives for all rare classes leading to improved performance across all metrics. Moreover, our multi-modal classifiers perform best across all metrics.

C. Vision-based Classifier Pipeline Implementation Details

For the transformer architecture of the visual aggregator detailed in Section 3.3, we use 4 transformer encoder blocks, each with a hidden dimension of 512 and MLP dimension 2048. As input to the first transformer block, we encode each image exemplar with a CLIP image encoder which remains frozen throughout training, yielding one embedding per exemplar. The set of embeddings are input with a learnable [CLS] token. The output [CLS] token is used as the final vision-based classifier.

To train the model we use the ImageNet-21k-P dataset (Ridnik et al., 2021) for 10 epochs. To speed up and improve training we store a dynamic queue of size $4096 \times K$ CLIP encoded embeddings, with 512 positions in the queue updated each iteration, using a last-in first-out policy. Each set of K represents encodings from K randomly sampled images for a single class. We use $K = 5$. For contrastive training, we use a temperature of 0.02 in the InfoNCE (van den Oord et al., 2018) loss function, the AdamW (Loshchilov & Hutter, 2018) optimiser with standard hyperparameters and a learning rate of 0.0002. Furthermore, during training we uniformly sample $k \in [1 : K]$ to simulate varying numbers of image exemplars being available for downstream OVOD when $\mathcal{C}^{\text{TEST}}$ is defined. Therefore, for a given iteration, there may $1 - 5$, visual embeddings input per class. Prior to input to the CLIP image ViT encoder, we apply random augmentations to each sampled image from ImageNet-21k-P. We use an augmentation policy similar to SimCLR (Chen et al., 2020), which includes `RandomResizedCrop`, `ColorJitter` and `RandomGrayscale`. We discover that test-time augmentation of the image exemplars available for the vocabulary in $\mathcal{C}^{\text{TEST}}$ improves downstream OVOD performance. For each image exemplar, we generate 5 test-time augmentations. Therefore if we have L image exemplars for a given class in $\mathcal{C}^{\text{TEST}}$, $5L$ augmented images are encoded using the CLIP image encoder and fused using the learnt transformer architecture — the visual aggregator — as described in Section 3.3. Use of test-time augmentation is ablated in Section A.

D. Sourcing Image Exemplars

In this section, we detail how image exemplars are sourced when performing experiments using vision-based classifiers and multi-modal classifiers, as described in Section 3.3 and 3.4, respectively. We start with an empty image exemplar dictionary (IED) for the 1203 LVIS classes.

From constructing the image-level dataset IN-L, we know that ImageNet-21k (Deng et al., 2009) contains 997 out of the 1203 classes in the LVIS vocabulary, using exact WordNet (Miller, 1995) synset matching. We add IN-L to our IED. The result is 988 classes have more than 40 images in ImageNet-21k. This leaves 215 classes for which there are too few or no image exemplars (< 40).

Next, to try and fill this gap, we turn to LVIS itself. We add the LVIS training annotations with area greater than 32^2 to our IED. There are now 1095 LVIS classes with more than 40 examples, leaving 48 classes with at least 10 exemplars and 60 classes with less than 10 exemplars.

The final dataset we turn to is VisualGenome (Krishna et al., 2017), which provides bounding box annotations for 7605 WordNet synsets. We include the annotations from VisualGenome with an exact WordNet synset match with the LVIS vocabulary to our IED. We now have 1110 LVIS classes with at least 40 exemplars and 1160 with at least 10 exemplars. Reducing our minimum required number of exemplars per class from 40 to 10 leaves 43 classes with too few exemplars.

At this point, we inspect each of the remaining 43 classes by hand and find that all have other synsets present in ImageNet-21k which are visually identical or very similar. For example, "anklet" is a "common" class in LVIS, for which LVIS gives a definition of "an ornament worn around the ankle" and a WordNet synset of `anklet.n.03`. This synset is not found in ImageNet-21k but `anklet.n.02`, defined as "a sock that reaches just above the ankle" by WordNet, is present and visual inspection shows these images to actually exactly match `anklet.n.03`. Therefore, we add ImageNet-21k images relating to `anklet.n.02` to our IED. As another example, `penny.n.02` (as in the penny coin) is a "rare" class in LVIS for which exemplars could not be found automatically. However, ImageNet-21k contains images of `coin.n.01` which is a hypernym `penny.n.02`. The images for `coin.n.01` are visually extremely similar and often identical to those one would expect for `penny.n.02` and so we add ImageNet-21k images relating to `coin.n.02` to our IED.

After applying some human effort as described above, our IED contains at least 40 image exemplars for 1110 (92% of LVIS classes) and at least 10 image exemplars for all LVIS classes.

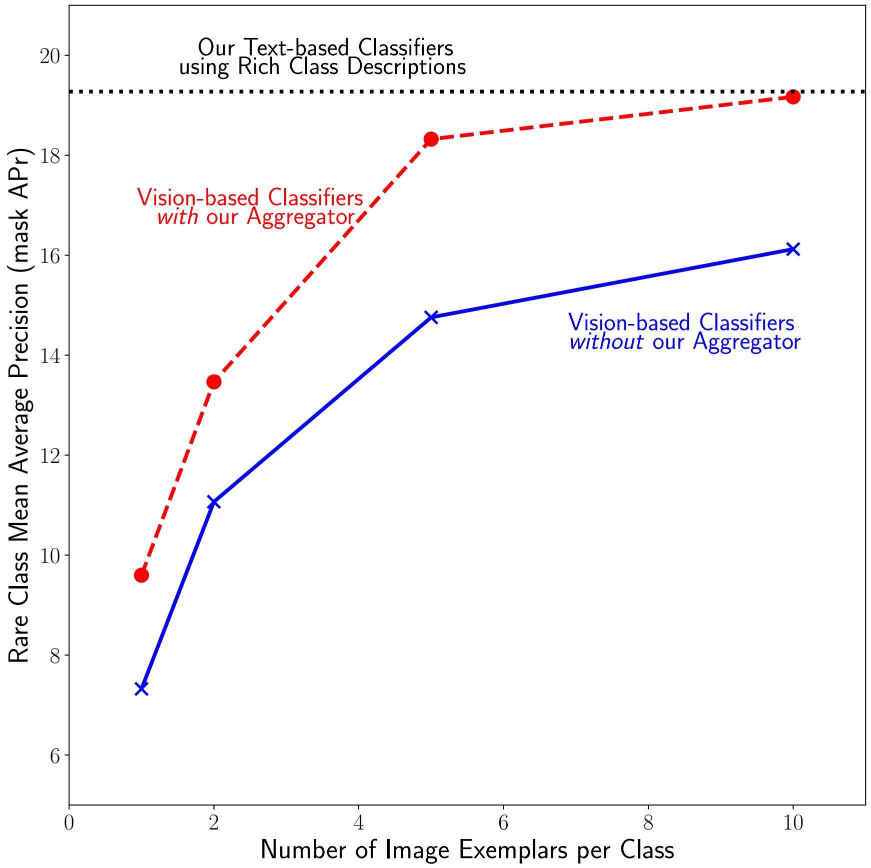

Figure 5. Detection performance of our vision-based classifiers on the LVIS OVOD benchmark. We vary the number of image exemplars available per class, K, to investigate the effect of the number of image exemplars on OVOD performance.

E. Varying Number of Image Exemplars for Vision-based Classifiers

In this section we show results using vision-based classifiers varying the number of K image exemplars used per class. Figure 5 shows performance on the LVIS OVOD benchmark for rare classes using $K = 1, 2, 5, 10$, where K is the number of image exemplars per class used. We compare our method which makes use of our aggregator (red dashed), which has a transformer architecture, with the simple vector mean of the embeddings (blue solid) for the K image exemplars. In both cases we apply the 'gentle' TTA detailed and ablated in Section A.

These results use LVIS-base as detection training data, no additional image-level labelled data *i.e.* $\mathcal{D}^{\text{IMG}} = \varnothing$ and CLIP ViT-B/32 as the pre-trained visual encoder to produce initial embeddings from each exemplar.

Figure 5 shows that for each value of K, the use of our aggregator boosts performance on rare classes demonstrating the utility of our aggregator at combining the most useful information from the K given exemplars. Our method for $K = 5$ surpasses the performance of $K = 10$ with simple vector averaging. For $K = 1$, our method improves performance by 2.3 APr which further demonstrates the improved feature representation — $K = 1$ involves no aggregation as only 1 exemplar is available per class.

Furthermore, we compare to the performance of our text-based classifiers which make use of rich class descriptions sourced from a GPT-3 model. Our vision-based classifiers cannot surpass the performance of our text-based classifiers demonstrating the need for more research into using image exemplars for OVOD.

F. Additional Example Class Descriptions

In this section we show a selection of rich class descriptions sourced from the `text-davinci-002` text completion model from OpenAI. For each class in the LVIS vocabulary we generate 10 rich descriptions. We also give the LVIS frequency category — rare, common, frequent.

F.1. Generated descriptions for "bagpipe" (rare)

1. A bagpipe is a wind instrument with a bag that is filled with air.
2. A bagpipe typically consists of a blowstick, a chanter, and one or more drones.
3. A bagpipe is a musical instrument that has a bag, a blowpipe, and usually two drones.
4. A bagpipe is a wind instrument with a bag that collects air, a reed pipe for each note, and a blowpipe.
5. A bagpipe is a musical instrument that is played by blowing into a bag of air.
6. Bagpipes vary in appearance, but most have a bag made from a animal skin, a blowpipe, a chanter, and one or more drones.
7. A bagpipe is a musical instrument that has a bag, a blowpipe, and usually drone pipes.
8. A bagpipe is a musical instrument that is usually made out of wood.
9. A typical Highland bagpipe has a chanter with a double reed, a blowstick, three drones with single reeds, and a bag.
10. A bagpipe consists of a blowing bag, a chanter, a drone, and usually one or more drones.

F.2. Generated descriptions for "trench coat" (rare)

1. A trench coat is a coat made of heavy cloth, sometimes waterproof, that hangs to about knee length.
2. A trench coat looks like a long, military-style coat.
3. A trench coat typically looks like a long, belted raincoat.
4. A trench coat is a long, water-resistant coat that is typically worn over top of a suit.
5. A trench coat typically has a removable liner, a double-breasted front, and belted cuffs.
6. A trench coat generally refers to a type of coat that is longer than waist length.
7. A trench coat is a coat that is usually a little bit longer than waist length, has a tie or a belt around the waist, and has a collar.
8. A trench coat is a coat made of waterproof material, typically hip-length or longer, with a belt and a collar.
9. A trench coat is a long, light coat with a belt.
10. A trench coat is a raincoat made of heavy-duty fabric, typically poplin, gabardine, or drill.

F.3. Generated descriptions for "walrus" (rare)

1. A walrus is a large, flippered marine mammal with a bulky body, short limbs, and a large head with two long tusks protruding from the mouth.
2. A walrus is a blubbery mammal with long tusks, whiskers, and a seal-like face.
3. A walrus is a large, flippered marine mammal with a long, tusked head.
4. A walrus is a stocky, rounded pinniped with small flippers, short fur, and long tusks.
5. A walrus is a large, flippered marine mammal with a bulky body, short tail, and wide, flat head.
6. A walrus is a large ocean mammal with two long tusks, a thick fur coat, and large flippers.
7. A walrus is a large flippered marine mammal with a discontinuous distribution about the North Pole in the Arctic Ocean and sub-Arctic seas of the Northern Hemisphere.
8. A walrus is a large flippered marine mammal with a thick fur coat.
9. A walrus is a large marine mammal with a body shaped somewhat like a seal.
10. A walrus is a seal with a long face and large tusks.

F.4. Generated descriptions for “briefcase” (common)

1. A briefcase is a rectangular, portable case used to hold papers, documents, or other materials.
2. A briefcase is a small case used to carry documents and other small items.
3. A briefcase is a small, rectangular-shaped case that is used to carry important documents or other items.
4. A briefcase is typically a rectangle shaped bag made of leather or synthetic materials.
5. A briefcase generally looks like a small, rectangular case made out of a variety of materials, such as leather, canvas, or nylon.
6. A briefcase is a narrow rectangular case used to carry documents and other valuables.
7. A briefcase is a box-shaped bag typically used by businesspeople to transport important documents.
8. A briefcase is a rectangular leather case with a handle.
9. A briefcase is a small case used to carry documents and other small items.
10. A typical briefcase is rectangular and has a handle on the top.

F.5. Generated descriptions for “coin” (common)

1. A coin is a small, flat, round piece of metal or plastic that is used as money.
2. A coin has a head side and a tail side.
3. A coin is usually a small, flat, round piece of metal or plastic that is used as money.
4. A coin has a round shape and is flat.
5. A coin generally has a circular shape with a raised edge, and two faces --- one on each side.
6. A coin is a small, flat, round piece of metal or plastic that is used as money.
7. A coin is a round piece of metal with an image on one side and the words ``United States of America'' on the other.
8. A coin is a small, round, flat piece of metal or plastic that is used as money.
9. Sure, a coin is a small, round, flat piece of metal or plastic that is used as money.
10. A coin is a small, round, metal disk with an image on one side and raised lettering on the other.

F.6. Generated descriptions for “waffle” (common)

1. A waffle is a pancake-like pastry that is cooked in a waffle iron and has a distinctively grid-like pattern on the top and bottom.
2. A waffle is a dish made from leavened batter or dough that is cooked between two plates that are patterned to give a characteristic size, shape, and surface impression.
3. A waffle is a thin, round, batter-based cake that is cooked in a waffle iron and is usually served with syrup.
4. A waffle looks like a round, honeycomb-patterned cake that is cooked in a waffle iron.
5. A waffle is typically a leavened batter or dough that is cooked between two plates that are patterned to give it a characteristic size, shape, and surface impression.
6. A waffle looks like an egg-shaped pancake with deep indentations.
7. A waffle looks like a grid of squares.
8. A waffle is a pancake-like pastry that is made from a leavened batter or dough and is cooked between two heated plates.
9. A waffle is a pancake-like food that is cooked in a waffle iron.
10. A waffle is a hotcake with different patterns on it.

F.7. Generated descriptions for “avocado” (frequent)

1. An avocado looks like a pear-shaped fruit with green or blackish skin.
2. It is a green fruit that has a dark brown or black seed in the center.
3. An avocado is a pear-shaped green fruit with smooth, green skin and a large seed in the center.
4. An avocado is a fruit that is brown and bumpy on the outside and green and creamy on the inside.
5. An avocado is a fruit with a dark green or blackish skin and a soft, fleshy inside.
6. An avocado is a green, pear-shaped fruit with a smooth, fleshy texture.
7. An avocado is a pear-shaped fruit with smooth, green skin.
8. An avocado is shaped like an egg and has a greenish-brownish skin.
9. An avocado is typically a dark green or black color on the outside with a soft, light green or yellow color on the inside.
10. An avocado is a pear-shaped fruit with smooth, green skin and a large, pit in the center.

F.8. Generated descriptions for “sausage” (frequent)

1. A sausage is a cylindrical shape made of ground meat, typically pork, and a variety of spices and other ingredients.
2. A sausage is typically a ground-up mix of pork, beef, or other meats with spices and salt, encased in a thin skin.
3. A sausage is typically a cylindrical shaped food made from ground meat, spices, and other ingredients.
4. A sausage is a long, thin, cylindrical piece of meat.
5. Sausages are typically long, cylindrical shaped foods made from ground meat and spices.
6. A sausage is a tubular meat product typically made from ground pork, beef, or poultry.
7. A sausage is a cylindrical casing filled with meat, typically pork, and various herbs and spices.
8. A sausage looks like a tubular shape made of ground up meat that is usually encased in a thin layer of intestine.
9. A sausage is a cylindrical casing of meat that is typically filled with ground pork, although many other variations exist.
10. When cooked, a sausage is typically cylindrical and can vary in length.

F.9. Generated descriptions for “spectacles” (frequent)

1. A spectacle is a pair of eyeglasses.
2. A spectacles is a type of eyewear that consists of a frame that holds two lenses in front of the eyes.
3. Spectacles are a type of eyewear that helps people see more clearly.
4. Spectacles are glasses that are worn in order to improve vision.
5. A spectacles usually refers to a glass or plastic lens worn in front of the eye to correct vision, or protect the eye from debris, dust, wind, etc.
6. A spectacles is a type of corrective lens used to improve vision.
7. A spectacle is a lens worn in front of the eye to correct vision, for cosmetic reasons, or to protect the eye.
8. A spectacles has a frame that goes around your head and two lenses in front of your eyes.
9. A spectacles has two glass or plastic lenses in metal or plastic frames that rest on the ears.
10. A pair of spectacles is a frame that holds two eyeglasses lenses in front of a person’s eyes.

G. Example Image Exemplars

In this section we show a selection of image exemplars, for LVIS classes, found using the process described in Section D. We also give the LVIS frequency category — rare, common, frequent. For cases where the image exemplar comes from a dataset with bounding boxes (LVIS or VisualGenome) we show the bounding box in yellow.

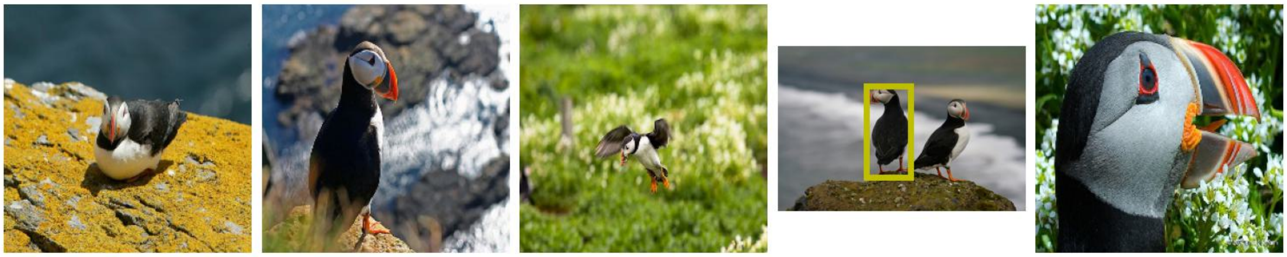

Figure 6. Image Exemplars for “puffin” (rare).

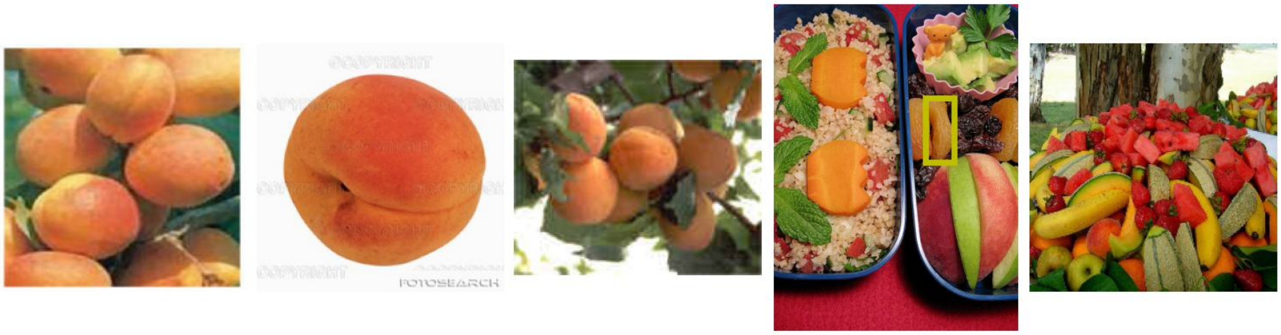

Figure 7. Image Exemplars for “apricot” (rare).

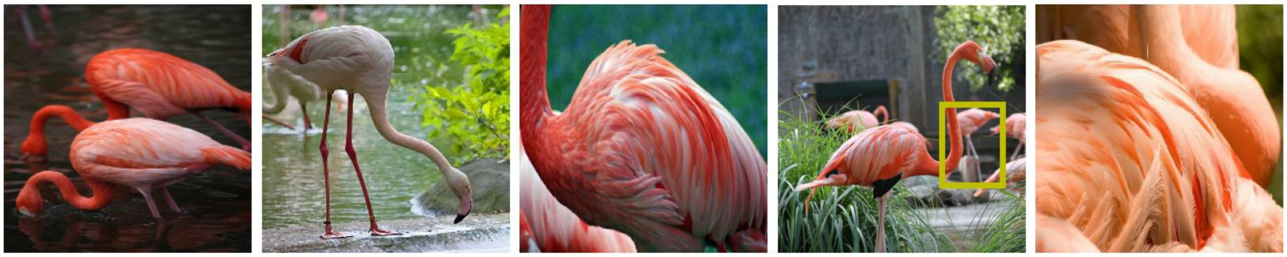

Figure 8. Image Exemplars for “flamingo” (common).

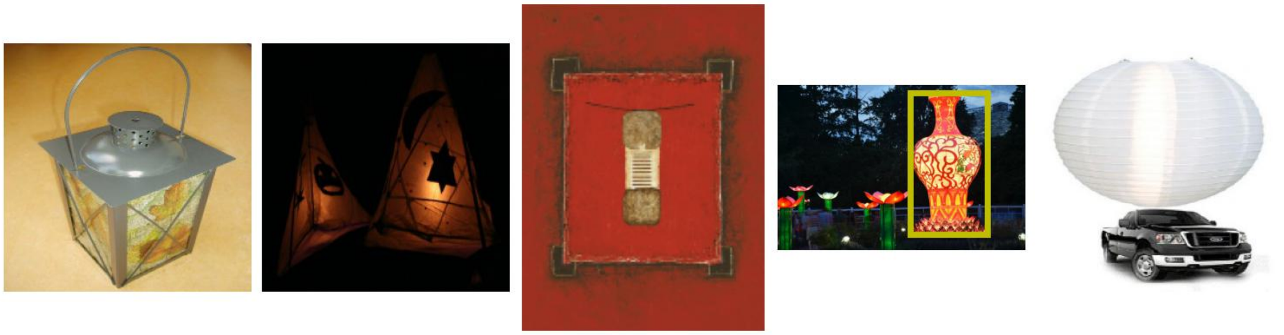

Figure 9. Image Exemplars for “lantern” (common).

Figure 10. Image Exemplars for "aerosol can" (common).

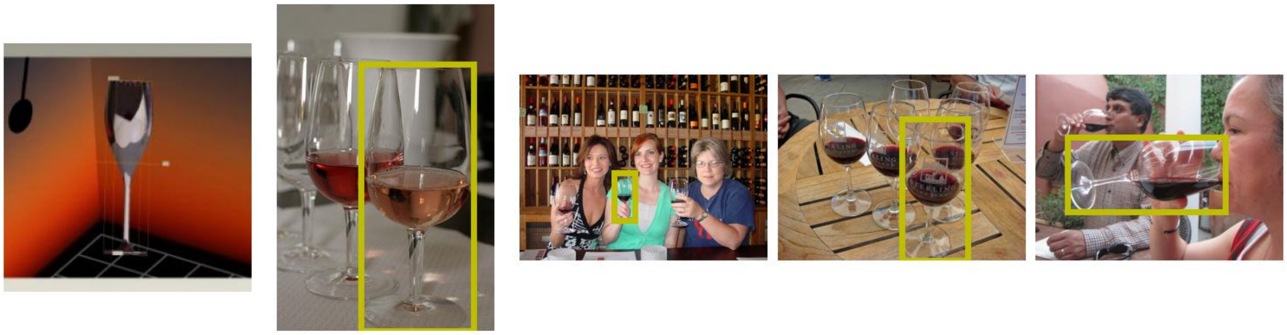

Figure 11. Image Exemplars for "wineglass" (frequent).

Figure 12. Image Exemplars for "beanie" (frequent).

Figure 13. Image Exemplars for "fire engine" (frequent).

H. More Qualitative Results

In this section we show more rare category detections on the LVIS OVOD benchmark using our multi-modal classifier trained with IN-L.

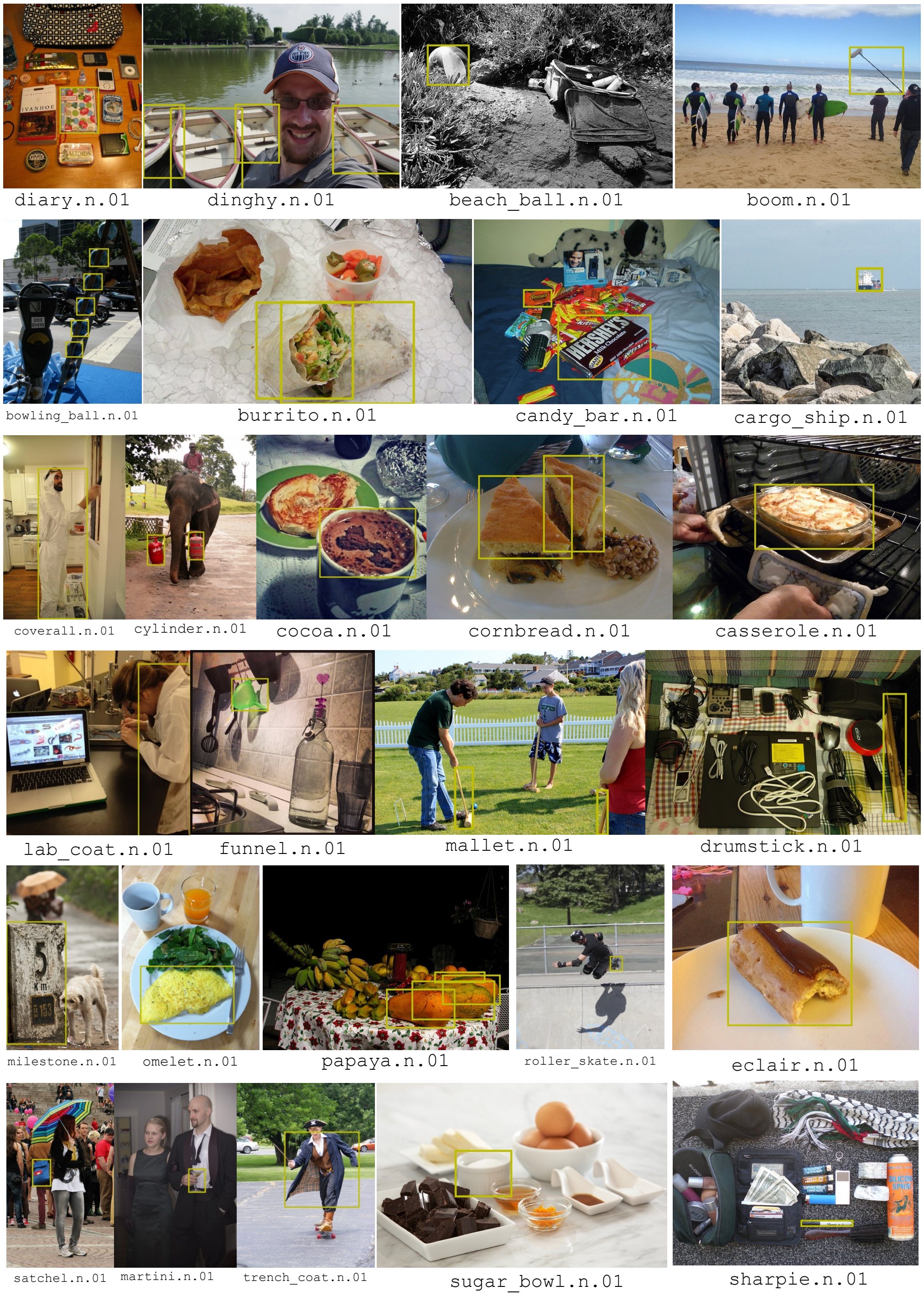

Figure 14. Additional qualitative results on LVIS OVOD benchmark.

I. Dugong

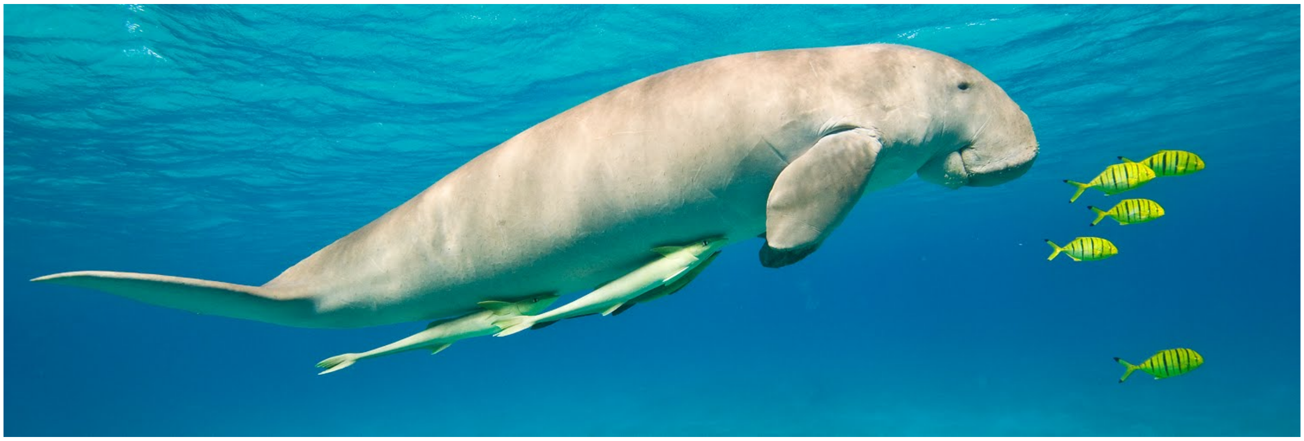

Figure 15. An example of a dugong — a visually distinctive marine species with a less well known name.