TUM Data Innovation Lab

Munich Data Science Institute (MDSI)

Technical University of Munich

&

PreciBake GmbH

Final report of project:

Open-vocabulary Object Detections of Inventory Items

Authors	Burak Bekci, Umut Onat, Florian Schraitle and Yushan Zheng
Mentor(s)	M.Sc. Mathias Sundholm, M.Sc. Maximilian Schreil, M.Sc. Sebastian Freytag
Project Lead	Dr. Ricardo Acevedo Cabra (MDSI)
Supervisor	Prof. Dr. Massimo Fornasier (MDSI)

Jul 2023

Abstract

The centrality of effective inventory management to food retail businesses' operational viability and profitability underscores the need for accurate and efficient inventory management systems. This research represents a collaborative undertaking between the TUM Data Innovation Lab and PreciBake, exploring the potential of an open-vocabulary object detection model in automating inventory tracking. The objective is to leverage recent advances in computer vision techniques and open-vocabulary object detection methods to overcome the complexities associated with large-scale inventory management.

Traditional inventory tracking methods frequently encounter difficulties concerning accuracy and time efficiency. Modern supervised learning solutions can address these issues given significant amounts of annotated data. However, considering the conditions of the inventory management systems, annotating data to train such models is quite costly. Even though such data is collected, the long tail distribution limits learning capabilities. Maintenance of such models will be challenging due to the dynamic nature of inventory items and their packages.

On the other hand, open-vocabulary models address the issues in inventory tracking without requiring a specifically annotated dataset. This pioneering technological solution extends the scope of object recognition beyond the constraints of annotated label space, thus accommodating an extensive range of object categories without excessive labeling and training. This automation significantly enhances the efficiency and accuracy of inventory management and effectively manages the introduction of new items or modifications in packaging. This research investigates the possibility of applying an open-vocabulary object detector to offer a flexible solution compared to supervised learning methods.

Our approach adopts a two-stage model, where the first stage proposes potential regions for potential object instances using a state-of-the-art object detection model. The second stage then classifies the proposed regions in an open-vocabulary setting. To assess the model's accuracy and efficiency, we conduct extensive experiments utilizing different methods as modular components of the model. We test our model not only on a retail product dataset that is highly aligned with the background of PreciBake, but also on a standard object benchmark, reporting the results qualitatively and quantitatively. The core contributions of this study encompass the proposal of a two-stage model for object detection in an open-vocabulary setting, the conduct of extensive experimental testing to evaluate the impact of various model components on detection accuracy and scalability, and the assessment of the model's performance and generalization ability across a diverse set of classes.

In conclusion, our research proposes an adaptable solution to the longstanding challenges faced in inventory management in the food retail industry. While we are encouraged by our research findings, we also recognize the need for further exploration in this domain. Future work will refine our model and expand its applicability to ensure excellent reliability and versatility across real-world scenarios.

Contents

1 Introduction

In the food industry, efficient inventory management ensures smooth operations and profitability. Inventory tracking is a specific component within inventory management that focuses on keeping accurate records of the quantity and location of various items within a culinary or retail business to prevent stockouts, minimize excess inventory, and dynamically schedule and optimize the supply chain.[12]

Traditional manual inventory tracking methods often suffer from inaccuracies, time-consuming processes, and limited visibility, necessitating a shift towards automated and technology-driven inventory management systems. The rapid development of deep-learning-based computer vision techniques has inspired leveraging computer vision into this field to automate the monitoring process, enhance accuracy, and improve efficiency. This facilitates food retail businesses to overcome these challenges and achieve enhanced inventory control in real-time.

Despite their success in various experimental settings, applying computer vision methods to track inventory items in real-world scenarios poses substantial challenges due to the inherent complexities of managing large-scale inventories comprising numerous unique items. As the size of the inventory expands, the task of accurate object detection becomes increasingly intricate, and the acquisition and maintenance of high-quality datasets grow progressively more demanding. The computational and logistical demands escalate due to the sheer volume of items and the inherent variations in appearance and characteristics across the diverse array of items within the inventory. Compounding these difficulties is the intermittent introduction of new items or modifications in packaging, which, from a conventional point of view, would entail meticulous relabeling and retraining efforts to ensure the sustained accuracy and efficacy of the inventory tracking system.

Open-vocabulary object detection methods have recently been proposed due to the expeditious progress of vision language pre-training models. These novel approaches aim to extend the scope of object categories beyond the confines of annotated label space and offer a more general, practical, and effective solution capable of locating and recognizing a broader range of object categories without excessive labeling and training.

In this joint project between TUM Data Innovation Lab and PreciBake, an AI and sensor technology company providing solutions to the gastronomy and baking industry, we investigate the possibility of constructing an open-vocabulary object detector for inventory tracking that can detect and classify an open-set of inventory items merely based on object descriptions or example images and can be easily extended to new categories that it was not explicitly trained for. We approach this issue by implementing a two-stage model that first proposes potential regions for potential object instances using a state-of-the-art (SOTA) object detection models[15, 21] and then classifies the proposed regions in an open-vocabulary setting. We focus on a comprehensive retail product dataset[39], aligning with PreciBake's expertise and background in this domain. We evaluate various methods as modular model components on the dataset with different experimental settings and report quantitative and qualitative results.

Our main contributions are summarized as follows:

- We propose a training-free two-stage model for object detection in an open-vocabulary setting.
- We conduct extensive experiments with vast combinations of variant components of the model to understand their impact on object detection accuracy and efficiency.
- We test our model on a retail product dataset as well as a common object benchmark to assess its performance and generalization ability.

In section 2, we shortly review object detection methods in closed-vocabulary settings and open-vocabulary settings. Then we introduce our proposed pipeline and describe our two-stage method in section 3. The qualitative and quantitative results and their analysis will be discussed in section 4. Then we will also include other explorations we have implemented in section 5 and conclude with future work worth looking into.

2 Related Work

Closed-Vocabulary Object Detection. Traditional object detection is generally a closed-vocabulary task that locates and classifies objects within images from a finite set of categories. Numerous approaches [41] have been developed due to significant advancements in deep learning technologies.

Closed-vocabulary object detection methods fall into two main categories: proposal-based and proposal-free. Proposal-based methods, like the Faster R-CNN [9] family, use a two-step process that first generates region proposals from image features, which are then refined and classified. Though accurate, they are computationally complex and may lack global context. In contrast, proposal-free methods such as YOLO [30], SSD [24], and EfficientDet [36] bypass the region proposal stage, using predefined anchor boxes to predict object classes and bounding box coordinates directly. These models are more efficient and simpler but may lack some accuracy.

Recent advances in Transformers [38] have revolutionized the natural language processing (NLP) field and begun to reshape the computer vision community. These technologies have improved object detection by modeling complex spatial relationships within images. DeTR[4] introduced an end-to-end trainable detector by combining CNN and Transformers and removing hand-crafted modules. The Swin Transformer[25] provided a transformer-based backbone by splitting input images into patches and processing the patches hierarchically. Though still in an early stage, Transformer-based approaches have achieved state-of-the-art results on the COCO benchmark[22] and have exhibited great potential for further breakthroughs.

However, closed-vocabulary methods have limitations, including restrictions on generalizing to new or unseen object classes. Therefore, research now focuses on open-vocabulary methods capable of adapting to diverse and evolving object categories.

Open-Vocabulary Object Detection. The pursuit of a more general objective beyond the constrained set of object categories has been propelled by the exponential growth in computational power and the advancement of deep learning methodologies. Despite the inherent challenges,

different approaches have been proposed to identify and classify emerging classes characterized by an unbounded open vocabulary at inference time.

VilD [10] uses a two-stage pixel-based detection pipeline, distilling knowledge from vision and language models. HierKD [26] combines instance-level and global-level knowledge distillation, bridging gaps between two-stage and one-stage methods. The Pseudo Caption Labeling (PCL)[5] pre-processing technique employs a caption model to create descriptive pseudo caption labels for object instances, providing dense samples for distillation.

Besides methods based on knowledge distillation models, another popular category involves visual language models (VLMs)[13, 29] and region text pre-training approaches that leverage large-scale image-text pairs to align visual and text features. OVR-CNN [43] leverages caption data for novel class detection and replaces classifiers with fixed text embeddings. In contrast, attribute-Sensitive OVR-CNN [3] aligns vision regions with word embeddings, introducing an adjective-noun negative caption sampling strategy. OWL-ViT [27] modifies pre-trained VLMs' image encoder and adds a lightweight classification and box regression head. RegionCLIP[20] matches image regions to region-level descriptions using pseudo labels generated by CLIP[29] and finetunes a visual encoder. MaMMUT[17] combines contrastive and generative learning in a multi-modal pre-training framework. GroundingDINO[23] combines Transformer-based detector DINO with grounded pre-training by performing vision-language modality fusion.

To facilitate knowledge transfer, some methods adopted prompt modeling techniques to incorporate learned prompts into the foundation model. DetPro[6] forces negative proposals to be equally dissimilar to any object class. PromptDet[7] introduces category descriptions and explores their position in the prompt, using cached web data for enhancement. Based on DETR[4], CORA[40] proposes region prompting and anchor pre-matching mechanisms to mitigate the gap between global image features and region features. Following OV-DETR[42], Prompt-OVD[34] improves novel object classification with Region of Interest (RoI) pruning techniques and RoI-based masked attention.

Additionally, attempts have been made to address the long tail problem and deal with rare and unseen data in open vocabulary settings. Detic[45] improves performance via image-level supervision, while MM-OVOD[14] utilizes multi-modal text embeddings as the classifier. PB-OVD[8] generates pseudo bounding box annotations using Grad-CAM[32] and aligns region and word embeddings. VL-PLM[44] trains Faster R-CNN[9] as a two-stage class-agnostic proposal generator. Whereas LocOV[2] trains Faster R-CNN with class-agnostic proposals by matching region features with images and text embeddings with captions.

Visual Language Pre-training Models Contrastive Language-Image Pre-training (CLIP)[29] is a novel neural network model that amalgamates visual perception and language comprehension. CLIP utilizes a dataset populated with image-caption pairs sourced from the internet, unlike traditional approaches that rely on manually labeled image datasets for training. This method bears two key advantages: (i) The reliance on publicly available internet-sourced data eliminates the need for further annotation, thereby enabling the model to capitalize on large-scale data availability, and (ii) The inherent diversity in this dataset empowers the model to learn a broad spectrum of concepts.

CLIP uses contrastive learning to identify similarities and differences across data by learning to

align the feature spaces of analogous image-text pairs while concurrently distancing dissimilar examples. Through this, CLIP can discern the semantic relationship linking images to their corresponding textual descriptions. Consequently, CLIP's understanding of natural language allows it to refer to familiar visual concepts or describe novel ones, facilitating its zero-shot transfer to various downstream tasks.

3 Methods

3.1 Preliminaries

The advent of open vocabulary object detection models [20, 23, 40] has provided a significant shift in object detection dynamics. Such models typically input arbitrary class labels as a query, offering an ostensibly generalizable approach to object classification. However, applying these models to the unique domain of supermarket products introduces a notable challenge.

Supermarket products, characterized by diverse packaging styles for a single product type, problematize straightforward categorization based on class labels. Such an approach may inadvertently group visually dissimilar items under different classifications. Figure 1 illustrates this issue, where even though the left and middle image depicts the same class, their visual appearance varies significantly. This challenge is further compounded when instances from two distinct categories exhibit more visual similarity than those within the same category. As evidenced in Figure 1, the packaging of different types of products can present striking similarities, rendering object classification increasingly tricky. Objects in the left and right images exhibit greater resemblance than those in the left and middle image, adding another layer of complexity to the detection process.

Figure 1: **Example images from the RPC dataset [39].** The left and middle images belong to the instant drink class, while the right image is an instance of the category dessert.

One solution to ameliorate the impact of such issues involves training models using detailed textual descriptions of product packaging rather than solely relying on class labels. Such a model would harness the visual elements of packaging, including color schemes, shapes, and textual descriptions, to accurately detect target objects within an image. Nonetheless, this solution calls

for a considerable volume of annotated data and corresponding efforts in training, which extends beyond the scope of this current project.

An alternative strategy, centering on visual-based features, could present a more fitting solution in this unique context. Distinctly, these features arise from visual stimuli rather than ambiguous semantic classes. Visual-based models offer a promising avenue for mitigating the limitations associated with linguistic ambiguity and subjectivity inherent in text-based descriptions. They provide an opportunity to capture the wealth of data in images, bypassing the reliance on textual interpretations. The capacity to harness this visual information can be pivotal for effective object detection, especially in environments as diverse and visually rich as supermarkets.

We introduce a two-stage pipeline to address the challenge of visually diverse same-type products. It inputs an RGB image and outputs object bounding boxes with their respective class labels. The first stage employs a class-agnostic object proposal method to detect all objects of interest within the input image, followed by stage two, object classification in an open-vocabulary setting. We implemented three variants for the second stage: image-based object retrieval, CLIP [29]-based classifier with text queries, and CLIP-based classifier with image queries. These will be illustrated in the ensuing sections of this Chapter.

3.2 Stage 1: Region Proposal

In this stage, we adopt a state-of-the-art transformer-based class-agnostic object proposal method [21], which instead of operating predominantly on local features, like previously prevalent convolutional neural networks (CNNs), is capable of capturing both local and global patterns within an image. This property has been proven to boost performance significantly for tasks such as object detection. With this object proposal backbone, we obtain a set of n bounding box proposals $\mathbf{B} \in \mathbb{R}^{n\times 4}$, which represent all objects of interest in the input image.

Our object proposal method over-detects an image, predicting large bounding boxes containing multiple objects, small unrelated objects, and unannotated regions. To address this issue of over-detection, we use a post-processing step $\Phi(\cdot)$ (see Figure 2) to decrease the amount of false positive predictions. We empirically determined $\Phi(\cdot)$ according to the inherent characteristics of the RPC dataset and define it as follows: (i) remove a prediction if it covers 60% of the image area, (ii) remove small bounding boxes that lie entirely within larger bounding boxes. We thus obtain $\mathbf{B}' = \Phi(\mathbf{i}, \mathbf{B})$ as resulting bounding boxes where $\mathbf{i} \in \mathbb{R}^{3\times k\times k}$ is the input image of size $k \times k$ and $\mathbf{B}' \in \mathbb{R}^{n'\times 4}$. Using these bounding boxes, we then create a set of image regions $\mathbf{R} = \{\mathbf{r}_0, \mathbf{r}_1, ..., \mathbf{r}_{n'}\} \in \mathbb{R}^{n'\times 3\times m\times m}$. Note that we resize each region to a fixed dimension m for the subsequent feature extraction step.

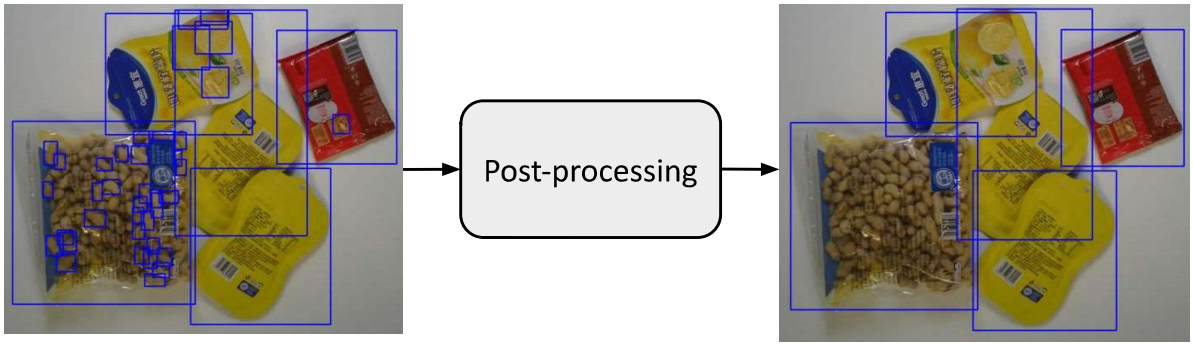

Figure 2: **An illustration of the impact of a post-processing step applied to the output of the object proposal model.** The image on the left exemplifies the initial result, containing multiple tiny bounding boxes that capture trivial and insignificant objects, leading to false positives. However, after applying our post-processing step, these erroneous proposals are effectively filtered out, resulting in a refined output. Consequently, recall and precision for the first stage are significantly elevated, enhancing the overall performance of the object detection pipeline.

3.3 Stage 2: Object Classification

3.3.1 Variant 1: Image Based Object Retrieval

Open-Vocabulary Setting Our initial idea of Variant 1 is illustrated in Figure 3. First, a feature extractor [35] computes features for each cropped region produced by stage 1 and for the input image query, which are then used to retrieve the most similar regions to the query. This conception fulfills the requirements of our goal and requires only a single image as a query to detect an open set of items without training. However, this method cannot fully utilize the capabilities of image encoders and dataset properties. To explore the limits of our models and achieve better results, we opt for a few-shot method and reformulate it by implementing a K-NN classifier, which requires a set of images and their labels for feature matching and label assignment. Therefore, we compiled a set of feature vectors with labels based on the training set of RPC dataset [39], referred to as object gallery.

Figure 4 illustrates an overview of our revised method of variant 1. In this variant, we utilize a pre-computed gallery to label each object in the input image, reformulating our task as two-stage object detection.

Image Feature Extraction. Our feature extractor $\Psi(\cdot)$ extracts salient features from each of the regions $\Psi(\mathbf{r}_i) = \mathbf{F}_i^{\text{regions}}$. This feature extractor captures the nuances of each object by analyzing colors, shapes, and patterns within a region. The resulting feature vectors $\mathbf{F}^{\text{regions}} \in \mathbb{R}^{n' \times d}$ are d dimensional embeddings for each region that encode visual details to differentiate one object from another.

Object Gallery. Our object gallery consists of 53739 regions of objects from the exemplar images in the RPC (train) dataset [39], which were cropped using the corresponding ground truth boxes. Detailed clarification of the dataset can be found in Section 4.1. For each region

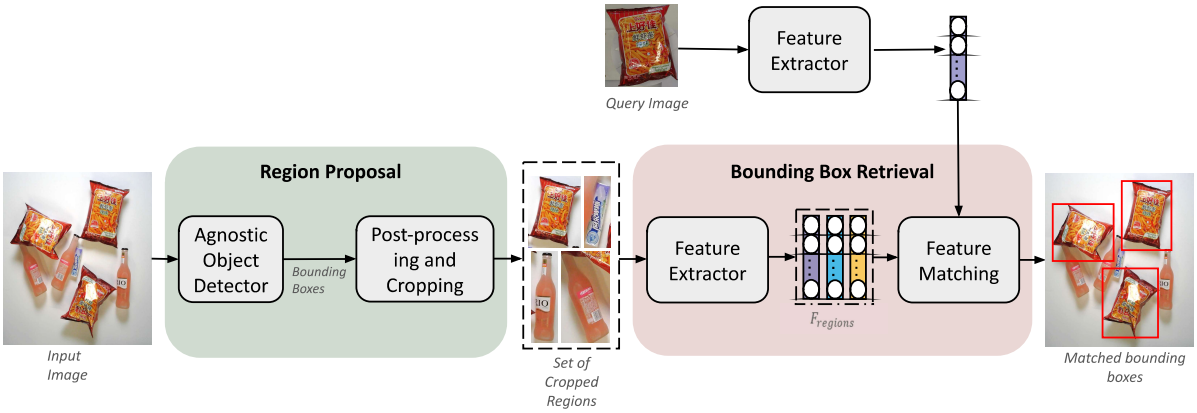

Figure 3: **The architecture of our visual-based open vocabulary two-stage framework with bounding box retrieval.** A Region Proposal Network finds all regions likely to contain objects. Then a pre-trained image encoder is used to extract image features of both the query image and proposed regions. Finally, regions whose image features yield a similarity score higher than a threshold are returned.

$\{\mathbf{r}_0^{\text{gal.}}, \mathbf{r}_1^{\text{gal.}}, ..., \mathbf{r}_{53739}^{\text{gal.}}\} = \mathbf{R}^{\text{gallery}} \in \mathbb{R}^{53739\times3\times m\times m}$ we follow the same principle as for the input image to create feature embeddings $\Psi(\mathbf{R}^{\text{gallery}}) = \mathbf{F}^{\text{gallery}}$.

Feature Matching. The final stage of our two-stage object detection model is feature matching using the K-Nearest Neighbors (K-NN) [33] algorithm. This algorithm is fundamentally based on the principle of proximity, wherein 'closeness' in the feature space translates to similarity. We calculate the similarity scores $\mathbf{S}$ between the region features $\mathbf{F}^{\text{regions}}$ and a set of q gallery features $\mathbf{F}^{\text{gallery}} \in \mathbb{R}^{q\times d}$ using the cosine similarity distance $\cos(\mathbf{F}^{\text{regions}}, \mathbf{F}^{\text{gallery}}) = \mathbf{S} \in \mathbb{R}^{n'\times q}$. This subset of gallery features is randomly sampled at inference time. To keep the solution in few-shot settings, we set $q << 53739$, including at least 1 sample for each existing class. At last, we assign labels to each bounding box based on the label of the gallery object that shares the highest similarity score with the respective bounding box.

3.3.2 Variant 2: CLIP-based Classifier with Text Queries

This variant of our approach heavily relies on the CLIP model [29]. The overall structure is shown in Figure 5. Using the identical region proposal method with variant 1, cropped regions are generated and fed to the pre-trained image encoder of CLIP to extract visual embeddings. Simultaneously, text embeddings are calculated from query texts. After normalizing, cosine similarity is calculated for each region-query pair, providing an alignment score accordingly. We refer to the outputs as similarity scores, ranging from -1 to 1, where 1 indicates a perfect match. We apply a second threshold on the similarity scores to select the objects that are aligned sufficiently with at least one of the queries. If there are multiple matching objects, i.e., satisfying the similarity score threshold, the object is assigned to the class label corresponding to the query with the highest similarity score. Furthermore, more than one query can correspond to a single label. In this sense, CLIP provides a very flexible framework. We compare these approaches in the evaluation section. For fairness, we maintained an equal number of queries for all our

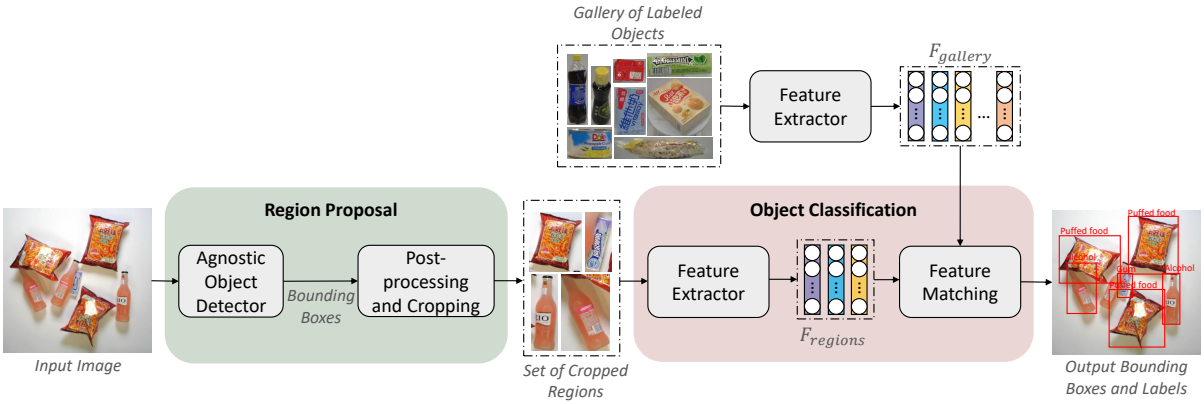

Figure 4: **The overall architecture of variant 1 of our two-stage model: Image-based Object Retrieval.** All regions detected by a Region Proposal Network [21] are processed by a pre-trained feature extractor [28, 35] to obtain feature vectors. Likewise, each training object is cropped using ground truth bounding box annotations, and their feature vectors are extracted with the same image encoder. Each feature vector in the test image is then matched with a training feature vector and assigned the label of the matched sample.

experiments for each class.

Specifically for this variant, we utilize the original Transformer-based text encoder of CLIP [29] to process input text query. A critical aspect of our approach lies in selecting suitable query texts. The choice of language to describe an object image influences the resulting embeddings. We perform prompt engineering to discover optimal query structures yield high model performance. We experiment with different templates as follows: "{}", "A photo of a {}.", "This is a photo of {}." and "A photo of the {}." where the curly brackets are filled with class labels.

We also attempted to generate image captions using BLIP [19] and BLIP 2 [18] to incorporate object descriptions, assuming that well-crafted descriptions could produce better embeddings and thus improved performance. Another advantage of this approach would have been that, since fine labels in English text format are unavailable in the RPC dataset, working with text queries posed a challenge because we had to work with the coarse labels instead of fine labels, which diverges from our goal to be able to classify each product. Generating captions held the promise of providing text descriptions for fine classes, thereby offering a potential solution to this problem. Unfortunately, the automatically generated captions fell short of our expectations and did not improve performance or enable us to work with fine labels as intended. So we set it aside for now. However, employing manual or better text descriptions might still offer performance gains. Hence we consider it a future avenue for investigation and future work to test our hypothesis.

In the final step, when a set of queries is given for a single class, we obtain a set of embeddings, and the next question is what is the most effective way to leverage these multiple embeddings. To tackle this question, we explored three possible methods:

1. Ensemble over the embedding space by calculating the mean of the embeddings of the given class to obtain a single final embedding

2. Ensemble over the score space by independently evaluating the similarity scores for each

query of the given class, then take their average.

3. Select the embedding with the highest similarity score.

After conducting several experiments and evaluations, we found that, despite the first method being the most efficient, we chose the third option as it yielded the best results.

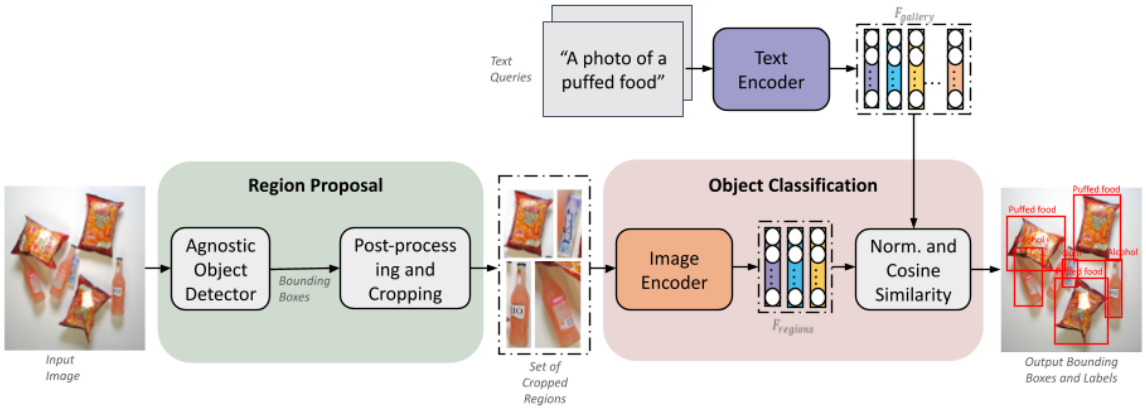

Figure 5: **Architecture of CLIP-based classifier with text queries as input.** All regions detected by a Region Proposal Network[21],by a pre-trained image encoder CLIP[29] to obtain feature vectors. The process is similar for both the input images and the query texts. The feature vectors of the query texts are extracted using the a text encoder that shares a joint embedding space with image encoder.. Subsequently, both the image and query embeddings are normalized and subjected to a dot product operation to determine the similarity score between them. Based on these scores, they are assigned to a label.

3.3.3 Variant 3: CLIP-based Classifier with Image Queries

This section focuses on explaining the functionality of our 2-stage model when it is used with image queries as illustrated in the Figure 6. The implementation details largely resemble those of text queries in the previous section. However, a key distinction arises when an image (One-shot) or a few images (Few-shot) serve as the query. Instead of using the text encoder to get the embeddings, we employ the same image encoder used on the input image's crops. Since the same model is used for query and input images, their embedding space is identical. Therefore, if a highly similar query and input images are fed to the model, their embeddings should align and yield a high similarity score.

In terms of prompt engineering, it differs from the text as well. Multiple distinct images of a single object can achieve broader coverage in the representation space. This, in turn, aims to enhance the overall performance by encompassing various potential locations of the object's embedding. Additionally, even though we did not explicitly implement it, a single image with various transformations, such as rotation, could also be used as a query to achieve the same purpose. We opted not to adopt this strategy in the RPC dataset because it already provides

multiple images of the same object captured from different angles. As shown in the evaluation section, we used 4 images taken from a different angle as queries for each object. To ensure consistency, these query images were cropped using their respective ground truth bounding boxes, to maintain a similar setting to the crops generated by our object proposer. Ensembling a set of queries given for a single class follows the identical approach utilized with text queries.

This variant bears a high resemblance to our first approach, with the only differences being using CLIP [29] image encoder instead of DINOv2 [28] or EfficientNet [35], and employing a normalization followed by a dot product to obtain the similarity score, instead of using a KNN classifier. One notable advantage of using CLIP is that it combines image and text queries, which was impossible in our first approach. This limitation arose because DINOv2 lacks a pre-trained text encoder that shares an embedding space with its image encoder. However, this specific area of ensemble learning, involving the combination of image and text queries, has yet to be explored in this project. We observed different similarity score distributions for texts and images, which led us to find separate thresholds for both methods. In a possible extension of this combination, one might need to scale the similarity scores to have the same distribution to leverage both query domains effectively.

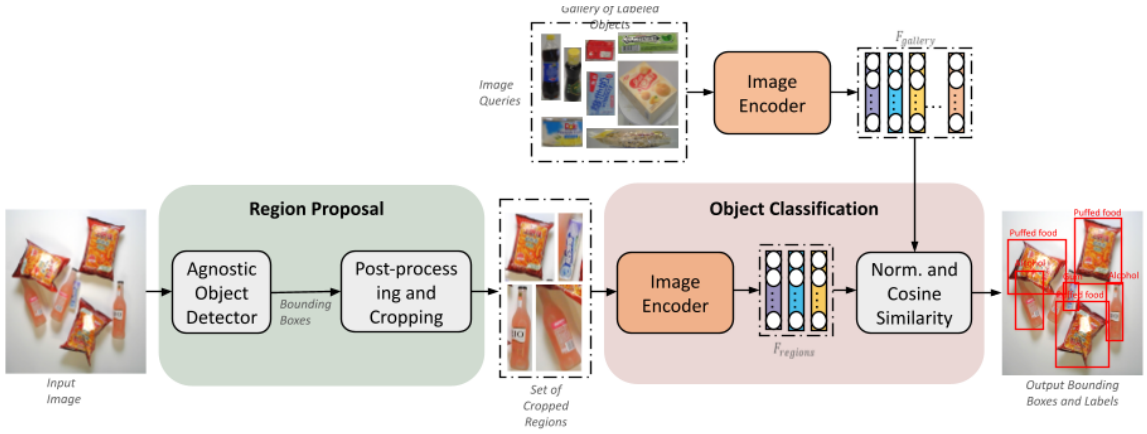

Figure 6: **Architecture of CLIP-based classifier with image queries.** All regions detected by a Region Proposal Network[21],by a pre-trained image encoder CLIP[29] to obtain feature vectors. The process is similar for both the image regions and the query images. Initially, the query images are cropped using their ground truth bounding box annotations, and then their feature vectors are extracted using the same encoder. Subsequently, both the image and query embeddings are normalized and subjected to a dot product operation to determine the similarity score between them. Based on these scores, they are assigned to a label.

Inference with Zero-shot Classification Setting

In this setting, the first step is determining all the classes for classification. It is a zero-shot classification because any novel class can be selected with a valid corresponding text or image query. Then, as explained in the implementation details section, embeddings of each query are calculated. This approach offers great flexibility since single or multiple texts, a single image (One-shot), or multiple images (Few-shot) can be chosen to represent a class. To classify the objects in a new input image, we get object proposals, then each of them receives a similarity score for each query. Object proposals that fail to achieve a sufficient score that surpasses the

threshold are discarded. We select one query from each class with the highest score for each of the remaining object proposals. Subsequently, a softmax function is applied to those similarity scores. This process leads to the assignment of the label with the highest similarity score to that region, with a probability equivalent to the respective objectness score multiplied by the value corresponding to the highest similarity score after the softmax function. One significant advantage of this method is that it requires no additional training. Novel classes can be easily added to the target labels, or existing ones can be modified or removed without further training.

Inference with Image Retrieval Setting In this context, we refer to "image retrieval" as searching for objects in an input image similar to a given query. This task is closely aligned with our problem definition. We can search for any class by defining a valid query or query for that specific class. Implementation of this approach is similar to Zero-shot classification setting. However, the key distinction lies in the final step, where we do not apply softmax since we have only one class. Instead, we use the similarity score, and if it exceeds the threshold, we consider it a successful proposal for the object we are looking for, with a probability equivalent to the respective objectness score multiplied by the normalized similarity score, which is scaled to a range between 0 and 1. We give less weight to this setting during the evaluation, primarily because finding suitable baselines and conducting a comprehensive evaluation took longer.

4 Experiments and Results.

4.1 Setup

Datasets. We use the RPC dataset [39] for our evaluations since it provides a comprehensive assortment of relevant inventory items of various product categories. This dataset contains 200 fine object categories and 17 coarse categories, which are: puffed food, dried fruit, dried food, instant drink, instant noodles, dessert, drink, alcohol, milk, canned food, chocolate, gum, candy, seasoner, personal hygiene, tissue, stationery. In total, the RPC dataset [39] encompasses 83,739 images, among which 53,739 are single-product exemplar images, and 30,000 are checkout images as shown in Table 1. The exemplar images of single products constitute the training set. In contrast, the validation and test set are comprised of checkout images with various product instances randomly chosen and placed on the counter. The validation and test set, thus, resemble realistic, cluttered, and complex checkout scenarios. The hierarchical structure of 200 fine-grained categories and 17 coarse categories can be exploited as auxiliary supervision information for better performance. For more details, please refer to the original paper [39].

Table 1: **Key statistics for the RPC Dataset [39].** Training set only includes exemplar images, while the validation and test set are comprised of checkout images.

Split	# Images	# Objects	# Objects/Image	# Categories/Image
Training set	53,739	53,739	1	1
Validation set	6,000	73,602	12.27	6.33
Test set	24,000	294,333	12.26	6.31

Metrics. Our experiments employ standard object detection metrics: mean Average Precision (mAP), Precision, and Recall. To obtain the mAP, we average the average precision over all classes. We additionally utilize the @k metric to assess prediction accuracy within the top k predictions. If the true label is within the top k predictions (sorted by confidence), we categorize the prediction as a true positive.

4.1.1 Baselines

We chose two open-vocabulary detectors as our baseline: GroundingDINO [23], CORA [40], and RegionCLIP [20]. All three baselines were trained on COCO [22], achieving state-of-the-art results. We used the original codebase on GitHub and initialized the models with the pre-trained weights trained on COCO [22], which the authors published. We evaluate all models on mAP on the 17 coarse classes of the RPC dataset [39]. We additionally experiment with a new hand-crafted prompt method for GroundingDINO [23]. More specifically, we change 10 prompts to resemble the products' properties, such as *blue cup dessert* and *green package puffed food*, instead of classes.

Table 2 depicts the results of our baselines on the RPC dataset [39]. The results show that all our baseline methods fail to maintain high performance on the RPC dataset [39]. GroundingDINO achieved merely 1.7% mAP, increasing to 4.9% mAP when using manually created descriptive prompts. However, this significant performance gain leads to mixed concepts within the models' embedding space and thus entangled outputs; for example, GroundingDINO would sometimes predict a nonexistent label like *blue cup dessert green package puffed food*. This unresolved problem [37] has constrained us from further elevating the performance of this model.

Contrastingly, CORA yields an extremely low mAP of 0.5% on the RPC dataset, whereas regionCLIP attains a modest improvement with an mAP of 4.7%. The low performance of GroundingDINO and CORA can be partially explained by their object prediction as they consistently predict fewer objects. On the contrary, RegionCLIP tends to overpredict the number of objects. Moreover, GroundingDINO and RegionCLIP predicted huge bounding boxes encompassing multiple objects in an image, accompanied by ambiguous labels.

Table 2: **Quantitative results of baselines on RPC dataset [39].** We run the models with provided pre-trained models in a zero-shot setting on the 17 base classes in the validation set of RPC dataset.

Model	GroundingDino [23]	GroundingDINO w/ prompts	CORA [40]	RegionCLIP [20]
mAP	0.017	0.049	0.005	0.047

The inadequate performance of all baseline models indicates the substantial domain gap between the RPC dataset and standard datasets like COCO [22] and LVIS [11]. As these standard datasets encompass generic objects, the resulting models lack exposure to unique product characteristics of the RPC dataset. Moreover, the imbalanced disparity of product packages for categories within the RPC dataset (Figure 1) also contributes to the subpar performance observed, thus requiring tailored solutions to address this challenging dataset.

Table 3: **Results of using ViT-H as region proposal network in stage 1 of our approach on the RPC dataset [39].**

Method	Precision	Recall
ViT-H [21]	0.38	0.98
ViT-H + post-processing	0.93	0.92

4.2 Stage 1: Region Proposal

4.2.1 ViT-H as a Region Proposal Network.

Experiment. The region proposal network is the first step of our two-stage models. The performance of this network is crucial for our method since the second stage relies entirely on it. For this stage, we utilize ViT-H [21] and evaluate its performance on RPC (val) [39]. As ViT-H is class-agnostic, we consider the problem as binary detection, using only foreground and background labels, and evaluate it on Precision and Recall. We further evaluate the effect of the post-processing step described in subsection 3.2

Results. Table 3 shows the result of our experiments. We obtain a recall of 0.97 precision of 0.38. Figure 7 illustrates the distributions of the metric values of ViT-H [21]. These plots show that ViT-H obtained high recall values in almost every image. However, precision values are generally low and deviate significantly across images. However, as seen in Table 3, while our post-processing method reduces the recall by a small margin, it improves precision significantly to 0.93.

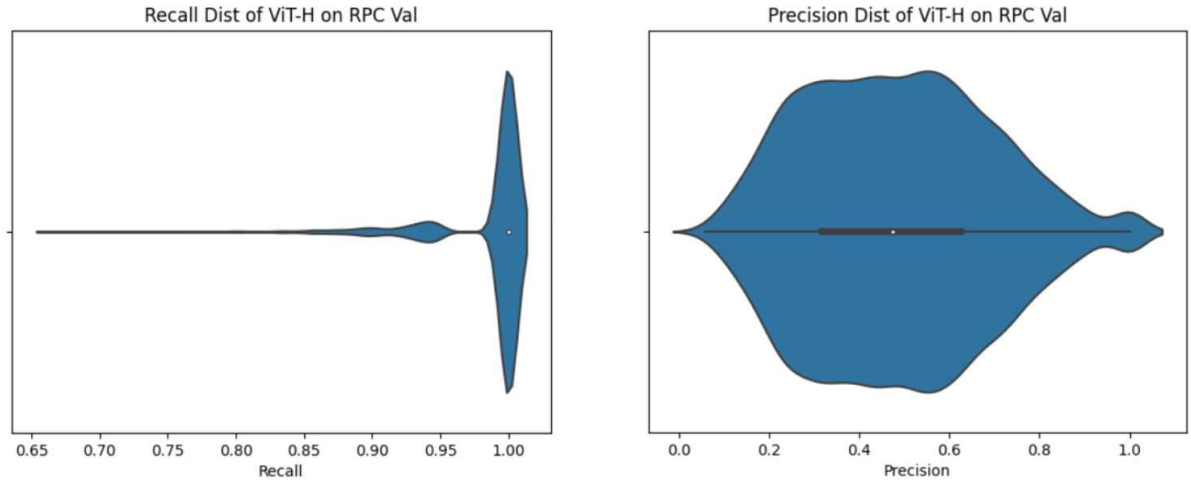

Figure 7: **Distribution of obtained precision and recall for ViT-H over test images plotted in a violin plot.** To generate these plots, precision and recall values are calculated for each image. ViT-H can yield satisfactory recall values across all images consistently.

4.2.2 SAM as a Region proposal Network

Experiment. Similarly to subsubsection 4.2.1, we have tried to leverage the Segment Anything Model (SAM) [16] for the RPC (val) dataset [39]. We fully segment the input with SAM using the Vit-B backbone in this experiment. Each segmentation is then converted into bounding boxes by constructing the smallest possible rectangle that encompasses all pixels of a segment. Note that the resulting bounding boxes are not further processed. We use the default hyper-parameters and pre-trained weights provided by the authors for the multipoint-prompt segmentation method.

Results. Figure 8 illustrates the distribution of precision and recall with an overall average precision of 0.37. This low average precision is due to the over-segmentation problem, creating excessive small bounding boxes. While the performance could most likely be further improved with further post-processing, we did not re-use SAM as a region proposal method due to hardware constraints and long inference time.

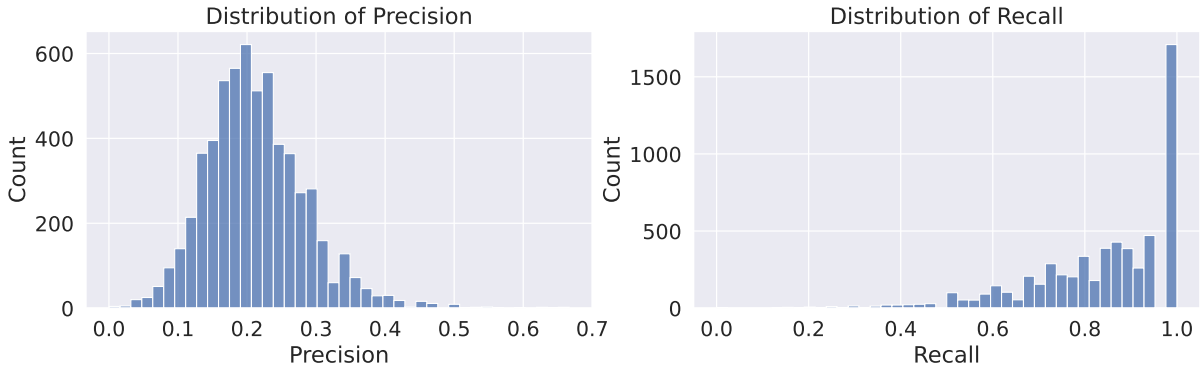

Figure 8: Precision and Recall of the naive SAM-region proposal method on the RPC dataset.

4.3 Stage 2: Object Classification

4.3.1 Variant 1: Image Based Object Retrieval

In this section, we perform several experiments on Variant 1. If not stated otherwise, we use mAP, Precision@k, and Recall@k as metrics. We set the threshold for the object proposal method to 0.2, filtering all bounding boxes below that threshold. We manually filter out all bounding boxes with dimensions smaller than 56 pixels, as this is the smallest bounding box in the training set. Furthermore, we resize each detected bounding box to 512x512, crop them to 448x448, and normalize them by the mean and standard deviation of ImageNet [31]. For feature matching, we set 0.4 as the similarity threshold and sample 4 instances per class randomly from the gallery. We did not train any models and initialized all with the respective pre-trained weights published by the authors.

Experiment - Feature Extractors. We evaluate Variant 1 (subsubsection 3.3.1) with DINOv2-S, DINOv2-L [28] and EfficientNet-M [35] as feature extraction backbone, and ViT-H [21] as object proposal method on the RPC (val) dataset [39].

Results - Feature Extractors. Table 4 shows the performance of Variant 1 with different feature extraction backbones. As can be seen, the supervised EfficientnetV2 [35] outperforms the self-supervised DINOv2 [28]. We achieve high recall, precision, and mAP values around 0.96, 0.84, and 0.76, substantially higher than our baselines (see subsubsection 4.1.1. This is because instead of being bound by the object classification, Variant 1 is only bound by the object detection, which already achieves very high performance subsection 4.2. The object classification outperforms the baseline by a huge margin because we leverage a gallery, which is not used by the baselines [20, 23, 40]. This gallery provides our approach a competitive advantage as we provide the method with the required knowledge at inference time.

We also visualize the feature space of DINOv2 [28] and EfficientnetV2 [35] in Figure 9. In this visualization, feature vectors representing dataset objects are color-coded based on their respective class names. Notably, data points belonging to the same class are clustered in both instances. Nonetheless, EfficientnetV2 can differentiate clusters with a significantly more significant margin than DINOv2. This capacity to more effectively differentiate between clusters substantially decreases the likelihood of mislabeling objects, thereby reducing the number of false positives compared to DINOv2.

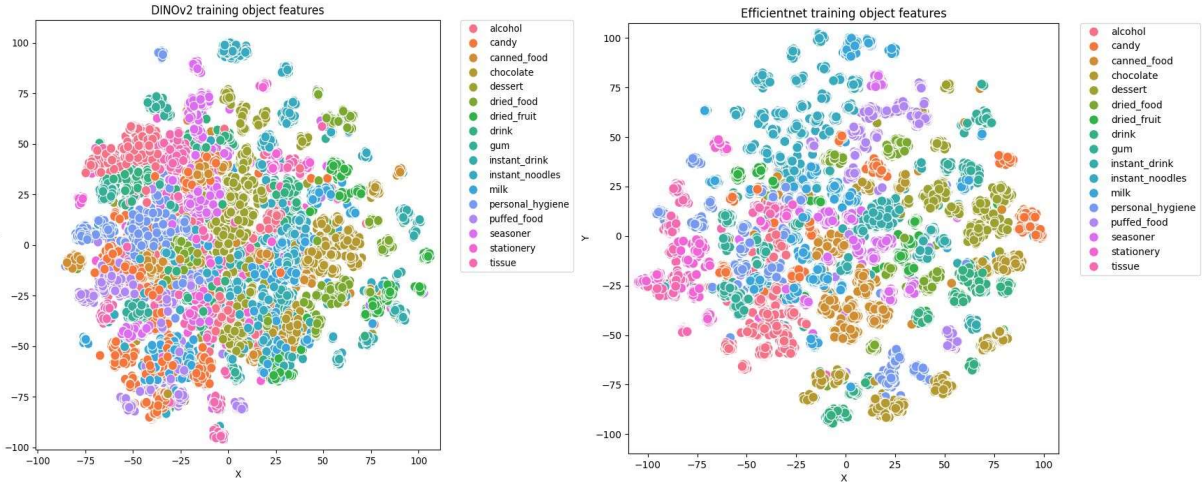

Figure 9: **Comparison between feature spaces generated by different image encoders with t-SNE.** Different colors in the graph represent different categories of products in the RPC dataset. Though both models cluster and distinguish between classes, EfficientnetV2 [35] could render a large margin between clusters compared with DINOv2 [28], thus providing better performance.

Experiment - Number of Sampled Gallery Features. The number of sampled gallery features is a crucial hyper-parameter for our feature matching. As the number of samples in the training set increases, our feature matching can capture a broader region of the feature space and generalize more. To observe the effect of the number of samples, we increased it from 1 to 4 for each class category.

Results - Number of Sampled Gallery Features. Table 4 shows the results of using 1 and 4 samples. The probability to sample an object more similar to the target object increases with more samples. Thus we observe a notable surge in performance between 1 sample and 4 samples.

Experiment - Camera Angles. The RPC (val) dataset [39] has multiple images of each product

Table 4: **Variant 1: Comparison between different feature extractors.** In general, increasing the number of sampled images tends to lead to better performance. Also, EfficientNet outperforms both versions of DINOv2 in all settings, demonstrating its capability to capture better features for this specific dataset.

Model		Fine(200)									Base(17)								
		mAP@k			Recall@k			Precision@k			mAP@k			Recall@k			Precision@k		
Backbone	# Samples	1	3	5	1	3	5	1	3	5	1	3	5	1	3	5	1	3	5
DINOv2-S [28]	1	0.16	0.26	0.31	0.86	0.91	0.92	0.26	0.38	0.44	0.37	0.57	0.66	0.92	0.95	**0.96**	0.54	0.73	0.8
DINOv2-L	1	0.21	0.34	0.4	0.87	0.87	0.94	0.32	0.47	0.54	0.51	0.68	0.74	0.93	**0.96**	**0.96**	0.67	0.82	0.88
EffNet-M [35]	1	0.28	0.42	0.5	0.91	0.94	0.94	0.4	0.55	0.62	0.59	0.77	0.82	0.95	**0.96**	**0.96**	0.73	0.87	0.91
DINOv2-S	4	0.31	0.44	0.5	0.92	0.94	0.95	0.45	0.59	0.65	0.53	0.68	0.73	0.95	**0.96**	**0.96**	0.68	0.81	0.85
DINOv2-L	4	0.38	0.53	0.59	0.93	0.95	0.95	0.53	0.67	0.73	0.65	0.76	0.78	0.95	**0.96**	**0.96**	0.77	0.87	0.9
EffNet-M	4	**0.53**	**0.67**	**0.73**	**0.95**	**0.96**	**0.96**	**0.65**	**0.78**	**0.82**	**0.76**	**0.84**	**0.87**	**0.96**	**0.96**	**0.96**	**0.84**	**0.91**	**0.93**

Table 5: **Results of the different image sampling methods over mAP score.** In each camera setting, we select one image per four available cameras. We observed slight improvements in each model when training images were sampled concerning camera angles.

Model			Fine(200)									Base(17)								
			mAP@k			Recall@k			Precision@k			mAP@k			Recall@k			Precision@k		
Backbone	# Images	Sampling	1	3	5	1	3	5	1	3	5	1	3	5	1	3	5	1	3	5
DINOv2-S [28]	4	Random	0.31	0.44	0.50	0.92	0.94	0.95	0.45	0.59	0.65	0.53	0.68	0.73	0.95	**0.96**	**0.96**	0.68	0.81	0.85
DINOv2-L	4	Random	0.38	0.53	0.59	0.93	0.95	0.95	0.53	0.67	0.73	0.65	0.76	0.78	0.95	**0.95**	**0.96**	0.77	0.87	0.90
EffNet-M [35]	4	Random	0.53	0.67	0.73	0.95	**0.96**	**0.96**	0.65	0.78	0.82	0.76	0.84	**0.87**	**0.96**	**0.96**	**0.96**	0.77	0.87	0.90
DINOv2-S	1*4	Per Cam	0.34	0.49	0.55	0.93	0.95	0.95	0.49	0.63	0.69	0.55	0.69	0.75	0.95	**0.96**	**0.96**	0.70	0.82	0.86
DINOv2-L	1*4	Per Cam	0.40	0.55	0.62	0.93	0.95	0.95	0.54	0.68	0.75	0.66	0.77	0.81	0.95	**0.96**	**0.96**	0.78	0.87	0.90
EffNet-M	1*4	Per Cam	**0.55**	**0.70**	**0.75**	**0.95**	**0.96**	**0.96**	**0.67**	**0.80**	**0.84**	**0.77**	**0.85**	**0.87**	**0.96**	**0.96**	**0.96**	**0.85**	**0.91**	**0.93**

taken from different camera positions. However, this variation can be challenging as models must recognize products from multiple angles. Our pre-trained image encoders have particular difficulty, as their extracted features focus on primary characteristics such as camera position and product shape, rather than specific image cues. This issue is demonstrated in Figure 10 and Figure 11, illustrating the similarity of feature vectors extracted by pre-trained models from the same camera angle, despite differences in the product.

We developed a sample selection method for creating sampling features to address this. Instead of randomly sampling an image from all angles, we select one image per camera angle, making our sampling feature space more diverse and resistant to camera angle changes.

Results - Camera Angles. Table 5 shows the improvement of selected images with respect to camera angle over selecting them randomly. Selecting images per camera improved the mAP score by 1-5% in each experiment.

Experiment and Results - COCO Benchmark [22] We evaluated Variant 1 model using ViT-H and DINOv2-M on COCO dataset. Testing on the validation set showed an mAP of 18.5% with one sampled image per category in the object gallery and 25.2% with ten samples. The model performed relatively well on animal-related categories like bear, giraffe and zebra, with APs around 70% but struggled with 1% mAP on abstract classes or similar concepts like snowboard and surfboard, potentially due to limited exposure in DINOv2's object categories.

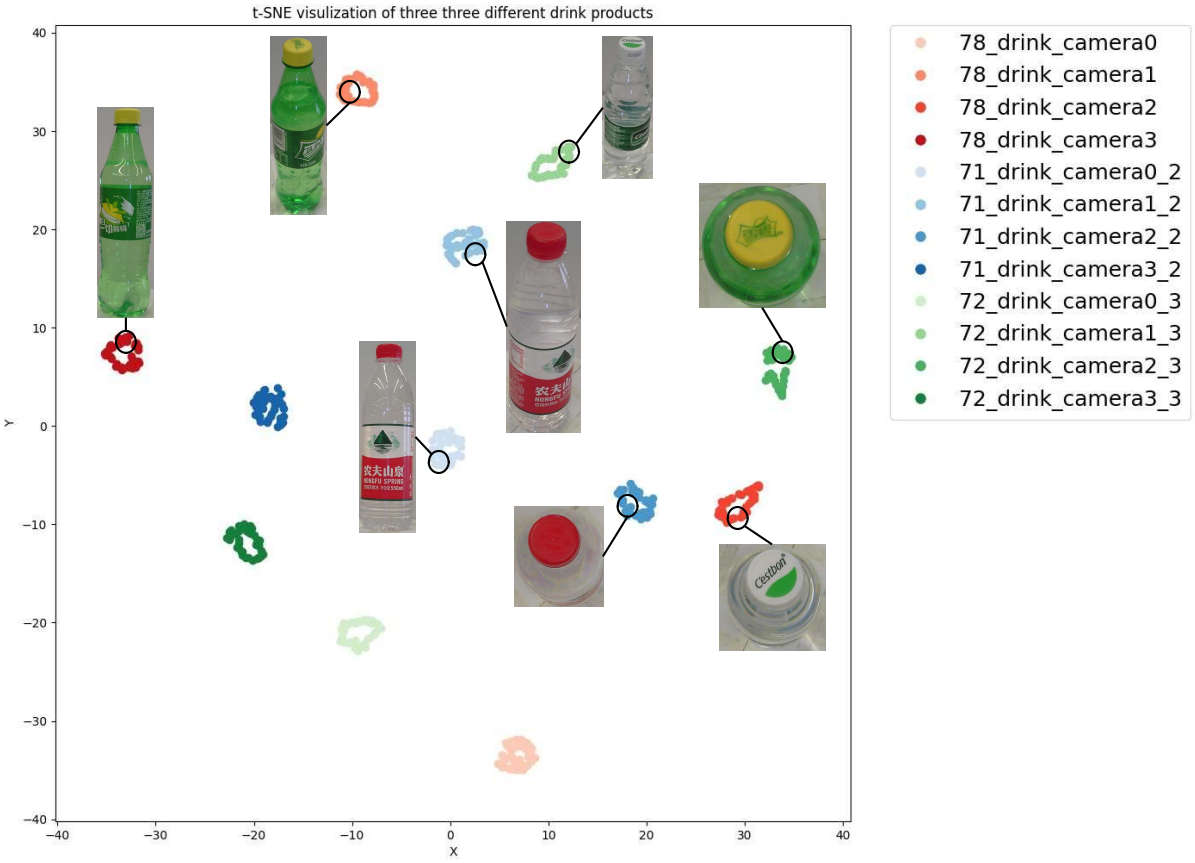

Figure 10: **t-SNE visualization of three different bottle products.** Each color indicates a different product whose dataset labels are given in the legend. Each hue of the same color depicts a different camera angle.

4.3.2 Variant 2: CLIP-based Classifier with Text Input

Experiments. We assess Variant 2 on both the COCO [22] and RPC (val) datasets [39] using mAP as metric. We also conducted an additional experiment, assessing three variations of CLIP using only the ground truth annotations to focus on the second stage of this variant.

Results. The results in Table6, reveal that when using ground truth bounding boxes, CLIP performs better on the COCO dataset [22], possibly because its data distribution more closely matches CLIP's training data than that of the RPC dataset. However, using VitDet [21] as object proposal backbone leads to a significant drop in CLIP's performance on the COCO dataset, with mAP scores decreasing from 0.46 to 0.13. The drop is smaller on the RPC (val) dataset, from 0.28 to 0.20. This suggests that VitDet does not perform optimally on COCO and may limit the model's overall performance. The performance disparity might also be due to the RPC-specific optimizations negatively affecting performance when applied to the COCO dataset. ViT-L/14 showed the best performance among the three CLIP variations and was subsequently used throughout this work.

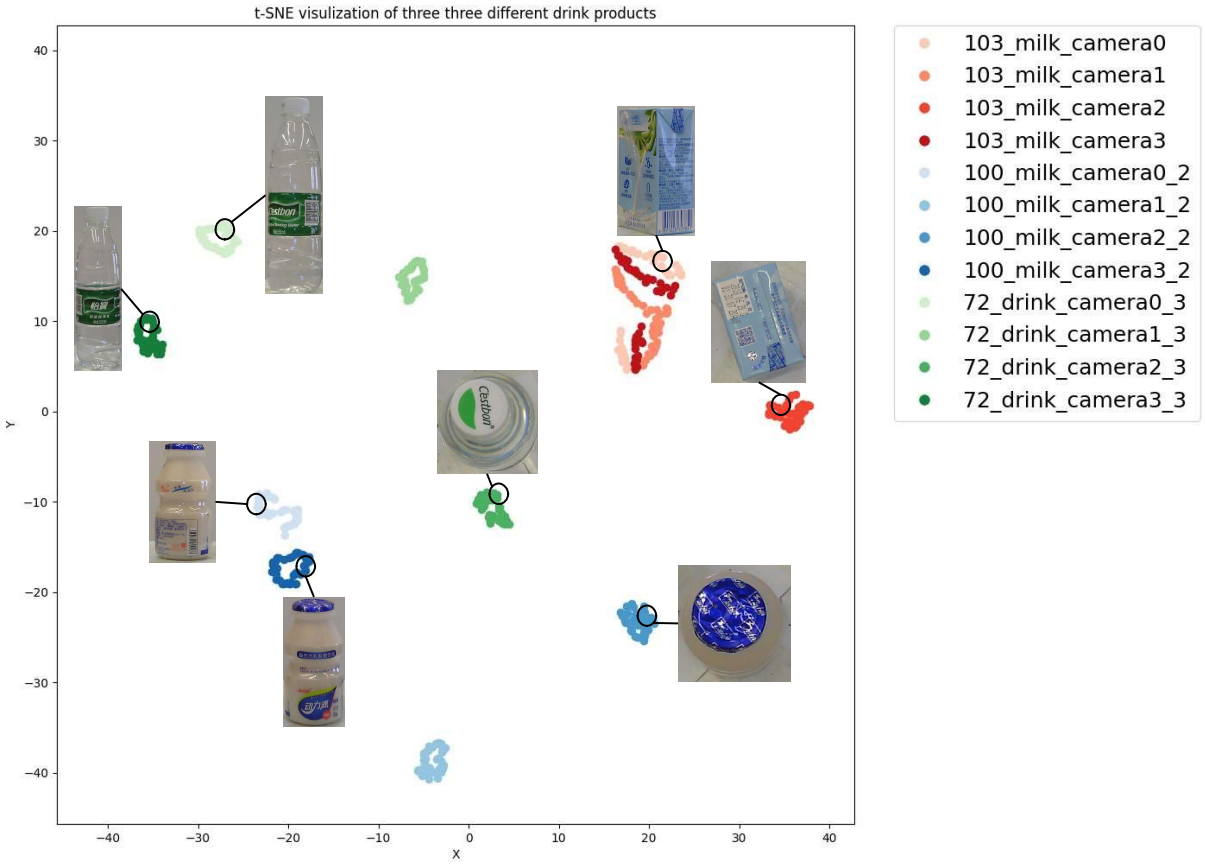

Figure 11: **t-SNE visualization of three different products.** Each color indicates a different product. Each hue of the same color depicts a different camera angle.

4.3.3 Variant 3: CLIP-based Classifier with Image Input

Experiment. We evaluate Variant 2 on both COCO [22] and the RPC (val) dataset [39] using mAP as a metric. For the RPC (val) dataset we consider both the 200 fine classes and 17 coarse classes. We sample different views as the RPC (val) dataset provides multi-view images. We also differentiate between using ground truth bounding boxes and the predicted bounding boxes of VitDet [21].

Results. As depicted in Table 7, our method yielded 0.43 mAP on the coarse labels and 0.25 mAP on the fine labels, improving performance with more samples. However, our model underperformed on the COCO dataset compared to the RPC dataset. Further investigation indicated that the random selection of low-quality or unrepresentative query images may have been a factor. Thus, we suggest that better image selection could enhance performance.

Using images as queries also provided an advantage over text queries. In the RPC dataset, only coarse labels are in English, precluding us from examining the fine labels.

Table 6: **Variant 2: Zero-shot Classification mAP Results using Text query.**

Model		Dataset	
Backbone	Object Proposals	COCO	RPC
CLIP ViT-L/14 [29]	VitDet [21]	0.13	0.20
CLIP ViT-L/14	Ground Truth	**0.46**	**0.28**
CLIP RN50x64	Ground Truth	-	0.25
CLIP ViT-B/32	Ground Truth	-	0.22

Table 7: **Variant 3: One-shot and Few-shot Classification mAP Results using Image query.**

Model		Datasets		
		RPC		COCO
# Sample	Object Proposals	Fine(200)	Coarse(17)	-
1	Ground Truth	0.17	0.54	0.12
4	Ground Truth	**0.34**	**0.62**	-
1	VitDet [21]	0.12	0.37	0.05
4	VitDet	**0.25**	**0.43**	-

4.3.4 Performance of 2 Stage Model

In this section, we analyzed the performances of our different methods described in the section 3. To tackle our problem, we constructed variants of models using different types of inferences, such as zero-shot, one-shot, and few-shot. Table 8 summarizes our results. Results in the table indicate that image queries are better suited than text queries and that the few-shot setting has more potential to elevate the performance.

5 Other Explorations and Future Work

This section focuses on other ideas we have explored that are outside our primary approach. Despite the limited depth of our exploration of these ideas, we hope they can still provide value as starting points for further research.

Table 8: **Comparison between 3 variants of our two-stage model on RPC dataset [39].** Variant 1, especially when using 4 sampled images as input, outperforms Variant 2 by a large margin.

		Model				mAP	
Variant	Backbone	Input Type	# Sample	Feature Dim.	Inference Type	Fine(200)	Base(17)
Variant 1	EffNet-M [35]	Image	1	1280	Zero-Shot	**0.20**	**0.38**
Variant 1	EffNet-M	Image	1	1280	One-Shot	**0.28**	**0.59**
Variant 1	EffNet-M	Image	4	1280	Few-Shot	**0.53**	**0.76**
Variant 2	CLIP [29]	Text	1	512	Zero-Shot	-	0.22
Variant 3	CLIP	Image	1	512	One-Shot	0.28	0.16

5.1 Query Search

Our previous experiments showed that the primarily text-query-based baseline methods perform poorly on the RPC dataset. One possible explanation could be that the text description fails to describe the objects sufficiently. Of course, this could also be a problem with the text embedder rather than the text descriptions themself. Therefore to explore this problem and investigate the generalisability of the models 'understanding' of the text embedding space, the following method is proposed:

Instead of doing traditional prompt engineering, as in searching for a better object description in natural language/text space, we want to do it directly in the embedding space. This would provide the advantage that since were searching in a (for this purpose) continuous space and have the capability of computing gradients, we can employ gradient-based optimization methods. Specifically, given an annotated image, we can optimize the text embedding to find the hopefully somewhat general embedding that yields the desired detection results. Figure 12 provides a graphical representation of this process. More formally given a model $f_{\text{detect}}(x_{image}, f_{text\_embed}(x_{text\_description})) = \hat{y}$ decomposed into f_{detect} and f_{text_embed} and a sample $(x_{image}, x_{text\_description}, y)$ with x_{image} the image to do detection on, $x_{text\_description}$ the description of the object to detect, and y the ground truth defections, we want to compute:

$$x^*_{text\_emb} = \underset{x_{text\_emb}}{arg\,min}\ \mathcal{L}(f_{detect}(x_{image}, x\_text\_emb), y) \tag{1}$$

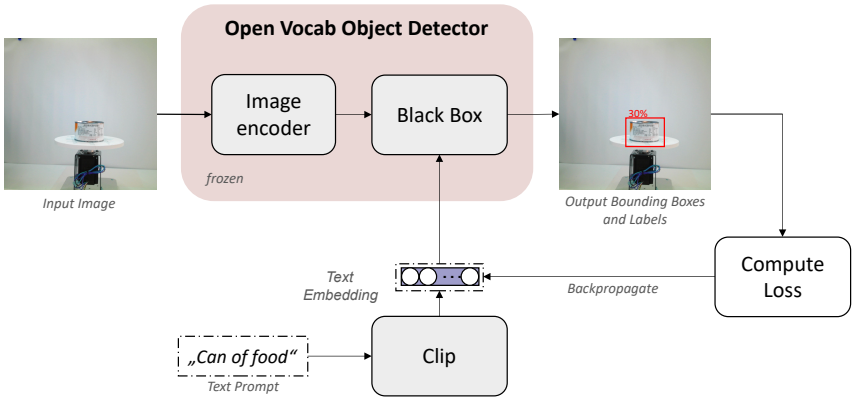

Figure 12: Depiction of query search

5.1.1 Experiment

First, a model to perform a query search needs to be selected. Initially, we intended to use SAM [16] for this, but since its text encoder part was never trained/released, it is impossible. Using the RegionCLIP[20] or our text-based CLIP model will not produce any exciting insight due to matching image to text query being based on a dot product. The GroundingDINO[23] uses Bert[1] to generate the text embeddings, but since word-level embeddings instead of sentence-level

embeddings are used here, this makes the embedding space challenging to traverse. Therefore the CORA model[40] was selected since it suffers from none of those mentioned above problems. (The only limitation is that its reference implementation is more challenging to work with). The embeddings are initialized with the model's text embedder and the object description. Finally, we experimented with different first-order optimizers (standard gradient decent, adam, momentum, rmsprop) and various hyperparameters (learning rate, momentum).

5.1.2 Results and Discussion

While we originally intended to investigate the transferability of text embeddings primarily generated this way, finding a good text embedding for a single image is already problematic. Specifically, when using the above-described optimizers, we could not find a set of hyperparameters that could produce any significant improvements over the default text embedding. Usually, after a couple of optimization steps, a plateau is reached.

This finding, of course, does not yet say anything about the existence of a point in embedding space that better describes the object. It merely shows that more elaborate optimization schemes are necessary if such a point exists. Our recommendations for other optimization schemes are to try L-BFGS as a quasi-second-order method, add randomization, and do multiple runs to overcome the possible local minima in which the optimization process gets stuck. Something else to investigate would be to replace the ReLU layers with LeakyReLU. While this would decrease the model's performance (if not retrained), it dramatically improves the ability to backpropagate through the network.

5.2 Future Work

Firstly, the baseline methods' evaluation demonstrated a clear need for further refinement of these SOTA methods and training with more extensive, comprehensive datasets. Secondly, we want to provide some suggestions for further improving the methods we have presented.

To improve the models performance on more crowded images, the image cropping process could be enhanced. Precisely segmenting the content of the bounding boxes (using a model such as SAM [16]) would allow for the removal of the background and possibly increase performance.

Similar to the previous idea, such a segmentation could be highly advantageous when cropping in feature space.

Another opportunity for further research could be investigating converting the two-stage model into a one-stage model. This could be done by removing the region proposal network and generating region proposals by a "brute force" approach. For further refinement, larger regions could act as proxies for the smaller regions.

Another interesting avenue could be exploring meta-learning for further refining the feature embeddings. This would also complement the open-vocabulary nature due to the emphasis on performing well in novel classes.

Finally, the query search idea could be explored with the advent of newer methods. Specifically, this would be interesting since one could assume that given a model that generalizes better, its text embedding space would also be more meaningful.

6 Conclusion

In this project, we explored object detection in the supermarket domain, explicitly addressing the complexities associated with visually diverse same-type products. We introduced a training-free two-stage model for effective object detection in response to this challenge.

The first stage of the model incorporated a class-agnostic object proposal method [21] to identify objects within an input image. This stage was bolstered by a post-processing step, notably improving the model's performance by enhancing precision.

In the second stage, we experimented with three different variants: image-based object retrieval, a CLIP [29]-based classifier with text input, and a CLIP-based classifier with image input. The image-based object retrieval method performed best of these variants, yielding high precision and recall values. This method also facilitated the integration of few-shot learning algorithms, further amplifying the overall model's performance.

Simultaneously, we ventured into open vocabulary settings, enabling our model to identify objects within images based on visual similarities instead of rigid pre-defined class labels. This approach demonstrated the ability to identify all class instances within a given image, thus providing flexibility and adaptability in object detection.

The two-stage model delivered promising results when applied with the image-based object retrieval variant, particularly in the visually rich and diverse supermarket environment. However, we recognize that there remains room for refinement. Further research and advancements are necessary to improve the model's real-world application and address existing limitations.

Prospective enhancements could involve integrating more annotated data or utilizing hierarchical category structures as auxiliary supervision. Such exploration could offer additional improvements to the model's performance, solidifying its effectiveness in object detection within the intricate setting of supermarket inventory.

Acknowledgement We want to express our gratitude to Mathias, Sebastian, and Max from PreciBake for their invaluable guidance and support throughout this project. Their expertise and encouragement have been instrumental in shaping our research. Additionally, we would like to extend our most profound appreciation to Tony Wang, who has generously offered instructive assistance in constructing this report.

List of Figures

List of Tables

Bibliography

[1] S. Alaparthi and M. Mishra. "Bidirectional Encoder Representations from Transformers (BERT): A sentiment analysis odyssey." In: *arXiv preprint arXiv:2007.01127* (2020) 23.

[2] M. A. Bravo, S. Mittal, and T. Brox. "Localized vision-language matching for open-vocabulary object detection." In: *DAGM German Conference on Pattern Recognition*. Springer. 2022, pp. 393–408 6.

[3] K. Buettner and A. Kovashka. "Enhancing the Role of Context in Region-Word Alignment for Object Detection." In: *arXiv preprint arXiv:2303.10093* (2023) 6.

[4] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko. "End-to-end object detection with transformers." In: *ECCV*. Springer. 2020, pp. 213–229 5, 6.

[5] H.-C. Cho, W. Y. Jhoo, W. Kang, and B. Roh. "Open-Vocabulary Object Detection using Pseudo Caption Labels." In: *arXiv preprint arXiv:2303.13040* (2023) 6.

[6] Y. Du, F. Wei, Z. Zhang, M. Shi, Y. Gao, and G. Li. "Learning to prompt for open-vocabulary object detection with vision-language model." In: *CVPR*. 2022, pp. 14084–14093 6.

[7] C. Feng, Y. Zhong, Z. Jie, X. Chu, H. Ren, X. Wei, W. Xie, and L. Ma. "Promptdet: Towards open-vocabulary detection using uncurated images." In: *European Conference on Computer Vision*. Springer. 2022, pp. 701–717 6.

[8] M. Gao, C. Xing, J. C. Niebles, J. Li, R. Xu, W. Liu, and C. Xiong. "Open vocabulary object detection with pseudo bounding-box labels." In: *European Conference on Computer Vision*. Springer. 2022, pp. 266–282 6.

[9] R. Girshick. "Fast r-cnn." In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 1440–1448 5, 6.

[10] X. Gu, T.-Y. Lin, W. Kuo, and Y. Cui. "Open-vocabulary object detection via vision and language knowledge distillation." In: *arXiv preprint arXiv:2104.13921* (2021) 6.

[11] A. Gupta, P. Dollar, and R. Girshick. "Lvis: A dataset for large vocabulary instance segmentation." In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 5356–5364 15.

[12] A. Jenkins. *What is Inventory Management? Benefits, Types, & Techniques*. Sept. 2020. URL: `https://www.netsuite.com/portal/resource/articles/inventory-management/inventory-management.shtml` 4.

[13] C. Jia, Y. Yang, Y. Xia, et al. "Scaling up visual and vision-language representation learning with noisy text supervision." In: *International conference on machine learning*. PMLR. 2021, pp. 4904–4916 6.

[14] P. Kaul, W. Xie, and A. Zisserman. "Multi-Modal Classifiers for Open-Vocabulary Object Detection." In: *arXiv preprint arXiv:2306.05493* (2023) 6.

[15] A. Kirillov, E. Mintun, N. Ravi, et al. "Segment anything." In: *arXiv preprint arXiv:2304.02643* (2023) 4.

[16] A. Kirillov, E. Mintun, N. Ravi, et al. "Segment anything." In: *arXiv preprint arXiv:2304.02643* (2023) 17, 23, 24.

[17] W. Kuo, A. Piergiovanni, D. Kim, et al. “Mammut: A simple architecture for joint learning for multimodal tasks.” In: *arXiv preprint arXiv:2303.16839* (2023) 6.

[18] J. Li, D. Li, S. Savarese, and S. Hoi. *BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models*. 2023. arXiv: 2301.12597 [cs.CV] 11.

[19] J. Li, D. Li, C. Xiong, and S. Hoi. *BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation*. 2022. arXiv: 2201.12086 [cs.CV] 11.

[20] L. H. Li, P. Zhang, H. Zhang, et al. “Grounded language-image pre-training.” In: *CVPR*. 2022, pp. 10965–10975 6, 7, 15, 18, 23.

[21] Y. Li, H. Mao, R. Girshick, and K. He. “Exploring plain vision transformer backbones for object detection.” In: *arXiv preprint arXiv:2203.16527* (2022) 4, 8, 11–13, 16, 17, 20–22, 25.

[22] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. “Microsoft coco: Common objects in context.” In: *ECCV*. Springer. 2014, pp. 740–755 5, 15, 19–21.

[23] S. Liu, Z. Zeng, T. Ren, et al. “Grounding dino: Marrying dino with grounded pre-training for open-set object detection.” In: *arXiv preprint arXiv:2303.05499* (2023) 6, 7, 15, 18, 23.

[24] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. “Ssd: Single shot multibox detector.” In: *ECCV*. Springer. 2016, pp. 21–37 5.

[25] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo. “Swin transformer: Hierarchical vision transformer using shifted windows.” In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, pp. 10012–10022 5.

[26] Z. Ma, G. Luo, J. Gao, L. Li, Y. Chen, S. Wang, C. Zhang, and W. Hu. “Open-vocabulary one-stage detection with hierarchical visual-language knowledge distillation.” In: *CVPR*. 2022, pp. 14074–14083 6.

[27] M. Minderer, A. Gritsenko, A. Stone, et al. “Simple open-vocabulary object detection with vision transformers. arXiv 2022.” In: *arXiv preprint arXiv:2205.06230* () 6.

[28] M. Oquab, T. Darcet, T. Moutakanni, et al. *DINOv2: Learning Robust Visual Features without Supervision*. 2023 11, 13, 17–19.

[29] A. Radford, J. W. Kim, C. Hallacy, et al. “Learning transferable visual models from natural language supervision.” In: *International conference on machine learning*. PMLR. 2021, pp. 8748–8763 6, 8, 10–13, 22, 25.

[30] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. “You only look once: Unified, real-time object detection.” In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 779–788 5.

[31] O. Russakovsky, J. Deng, H. Su, et al. “Imagenet large scale visual recognition challenge.” In: *International journal of computer vision* 115 (2015), pp. 211–252 17.

[32] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. “Grad-cam: Visual explanations from deep networks via gradient-based localization.” In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 618–626 6.

[33] B. W. Silverman and M. C. Jones. “E. fix and jl hodges (1951): An important contribution to nonparametric discriminant analysis and density estimation: Commentary on fix and hodges (1951).” In: *International Statistical Review/Revue Internationale de Statistique* (1989), pp. 233–238 10.

[34] H. Song and J. Bang. “Prompt-Guided Transformers for End-to-End Open-Vocabulary Object Detection.” In: *arXiv preprint arXiv:2303.14386* (2023) 6.

[35] M. Tan and Q. Le. “Efficientnetv2: Smaller models and faster training.” In: *International conference on machine learning*. PMLR. 2021, pp. 10096–10106 9, 11, 13, 17–19, 22.

[36] M. Tan, R. Pang, and Q. V. Le. “Efficientdet: Scalable and efficient object detection.” In: *CVPR*. 2020, pp. 10781–10790 5.

[37] *text prompts with multiple object categories: concepts are combined issue #85 idea-research/groundingdino2023*. May 2023. URL: https://github.com/IDEA-Research/GroundingDINO/issues/85 15.

[38] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. “Attention is all you need.” In: *Advances in neural information processing systems* 30 (2017) 5.

[39] X.-S. Wei, Q. Cui, L. Yang, P. Wang, L. Liu, and J. Yang. *RPC: a large-scale and fine-grained retail product checkout dataset*. 2022 4, 7, 9, 14–18, 20–22.

[40] X. Wu, F. Zhu, R. Zhao, and H. Li. “CORA: Adapting CLIP for Open-Vocabulary Detection with Region Prompting and Anchor Pre-Matching.” In: *ArXiv* abs/2303.13076 (2023) 6, 7, 15, 18, 24.

[41] S. S. A. Zaidi, M. S. Ansari, A. Aslam, N. Kanwal, M. Asghar, and B. Lee. “A survey of modern deep learning based object detection models.” In: *Digital Signal Processing* 126 (2022), p. 103514 5.

[42] Y. Zang, W. Li, K. Zhou, C. Huang, and C. C. Loy. “Open-vocabulary detr with conditional matching.” In: *European Conference on Computer Vision*. Springer. 2022, pp. 106–122 6.

[43] A. Zareian, K. D. Rosa, D. H. Hu, and S.-F. Chang. “Open-vocabulary object detection using captions.” In: *CVPR*. 2021, pp. 14393–14402 6.

[44] S. Zhao, Z. Zhang, S. Schulter, L. Zhao, B. Vijay Kumar, A. Stathopoulos, M. Chandraker, and D. N. Metaxas. “Exploiting unlabeled data with vision and language models for object detection.” In: *European Conference on Computer Vision*. Springer. 2022, pp. 159–175 6.

[45] X. Zhou, R. Girdhar, A. Joulin, P. Krähenbühl, and I. Misra. “Detecting twenty-thousand classes using image-level supervision.” In: *European Conference on Computer Vision*. Springer. 2022, pp. 350–368 6.

Appendix

6.1 Visualization of predictions

We visualized exemplary predictions of our first variant of models. Specifically, we used our best-performing model that utilizes Efficientnet and K-NN and uses one training image from each camera angle of each category. In all visualizations, we plot ground truth annotations on the left figure with blue color. The right figures depict predictions of the model which are drawn in either green or red color indicating the correctness of the prediction. While red indicates the predicted label is wrong, green indicates a correct prediction.

We visualized samples in both base (Figure 14, Figure 16) and fine label(Figure 13, Figure 15) settings.

Figure 13 illustrates an example of our models' efficacy. Even in crowded scenarios where objects' bounding boxes collide, our model can detect objects correctly. This figure also shows the effectiveness of the object proposal model. It successfully detected all objects in the input and provide to the second stage.

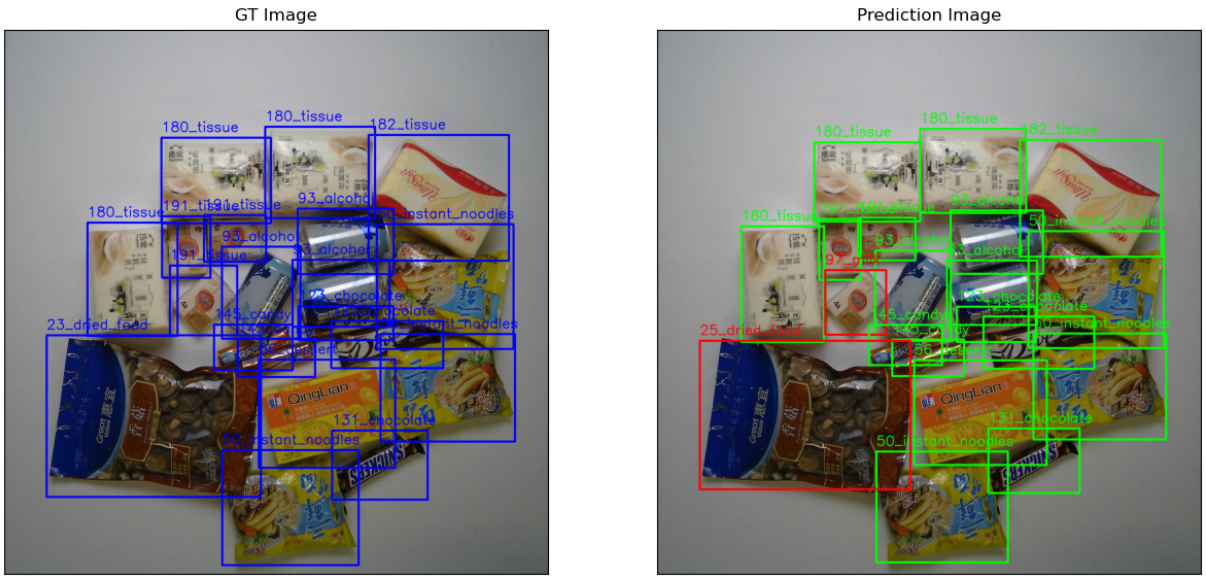

Figure 13: **An example output of our method in crowded scenario** Our model can perform well even in crowded object scenarios.

Figure 14 depicts an example of error propagate in our two-stage method. An extra prediction by object proposal model caused a false positive sample at the end.

Figure 15 and Figure 16 depict an example of improvements we get when we reduce our fine classes to base. Our model can detect three more objects correctly when base classes are used. It also shows that even if a prediction might be wrong for a fine class, its prediction stays in the same base class.

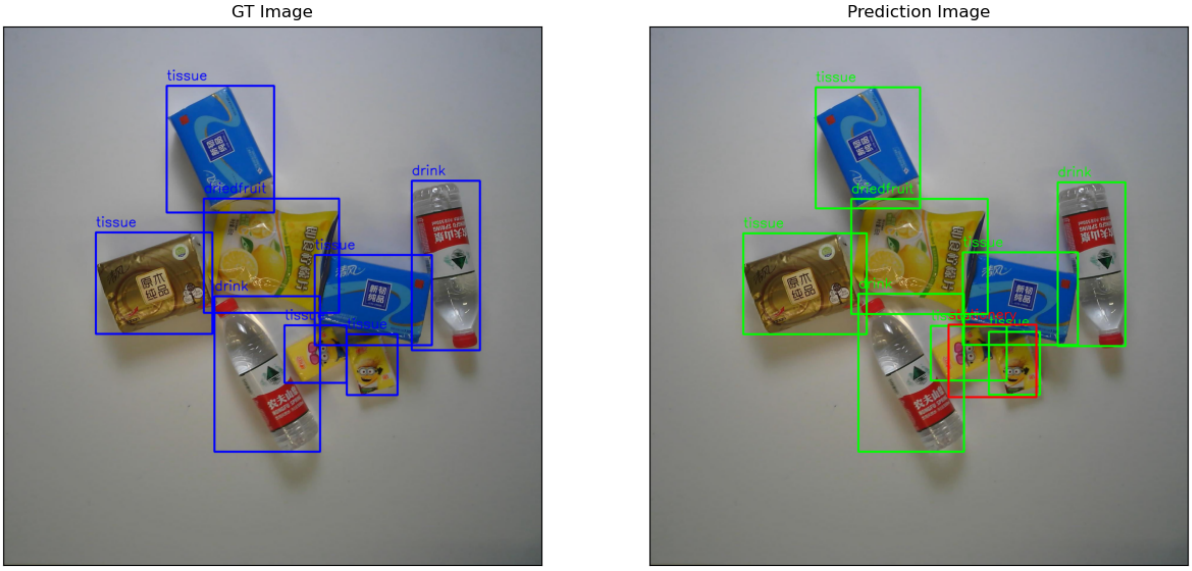

Figure 14: **An example output of our method. That predicts base classes.** Although each of the bounding items is correctly found, an extra prediction of the object proposal model caused a false positive at the output.

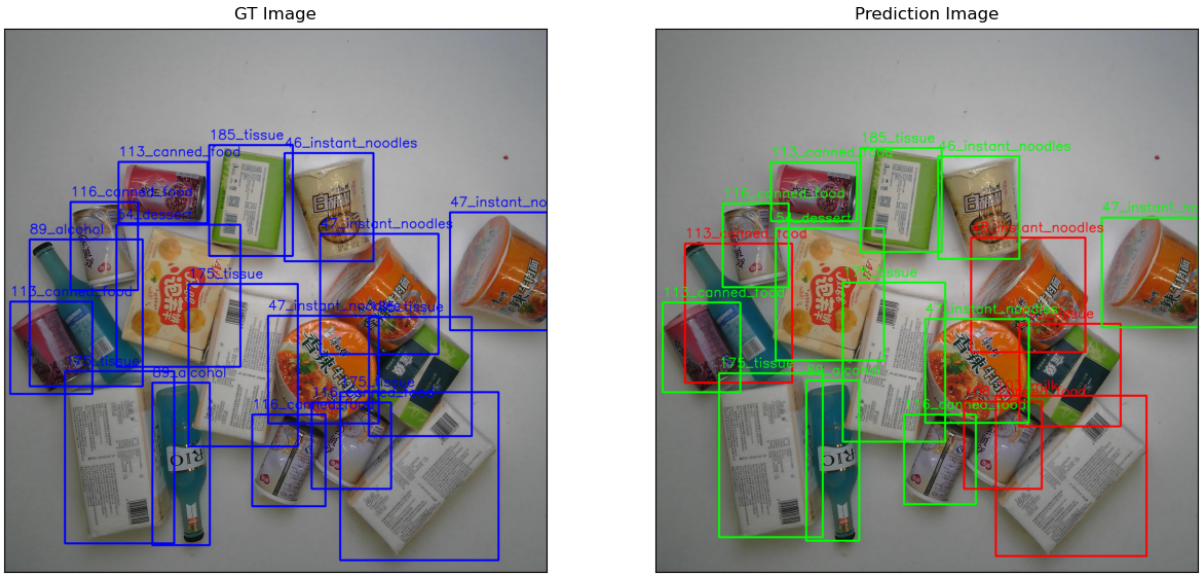

Figure 15: **An example output of our method with fine class predictions.**

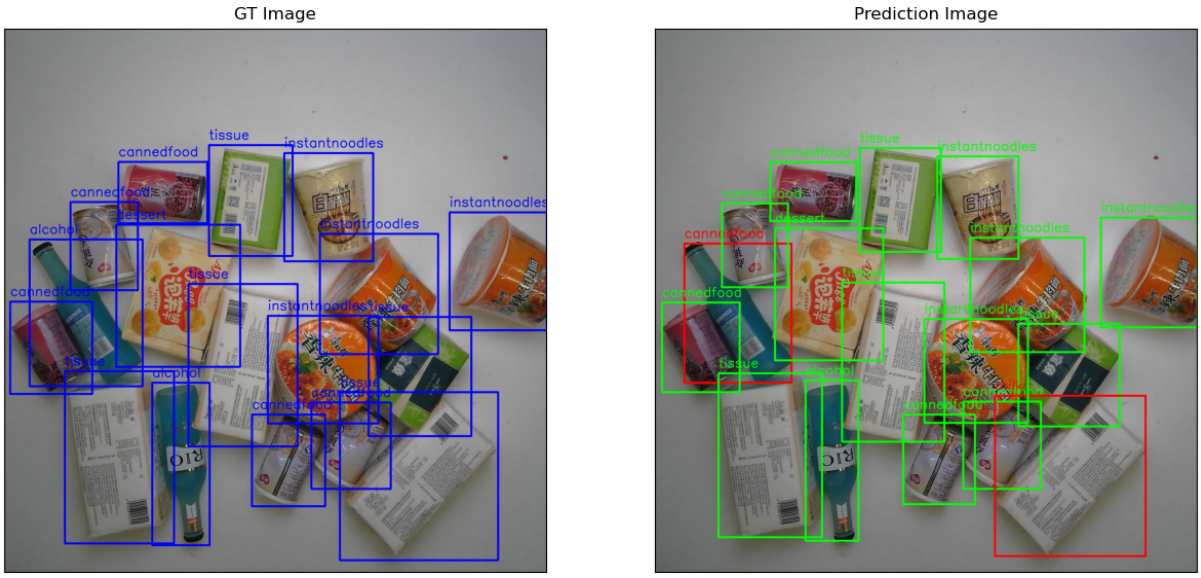

Figure 16: **An example output of our method with base class predictions.** Using base classes improved the final results.