Mobile User Interface Element Detection Via Adaptively Prompt Tuning

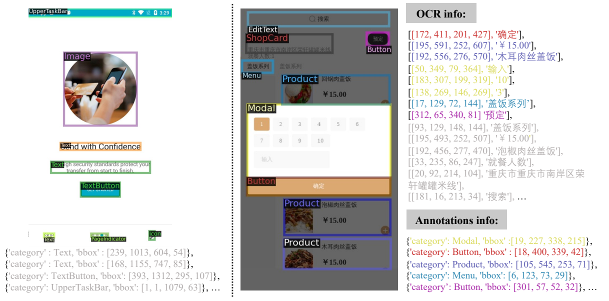
Zhangxuan Gu, Zhuoer Xu, Haoxing Chen, Jun Lan, Changhua Meng, Weiqiang Wang
Tiansuan Lab, Ant Group
{guzhangxuan.gzx,xuzhuoer.xze,chenhaoxing.chx,yelan.lj,changhua.mch,weiqiang.wwq}
@antgroup.com

Abstract

Recent object detection approaches rely on pretrained vision-language models for image-text alignment. However, they fail to detect the Mobile User Interface (MUI) element since it contains additional OCR information, which describes its content and function but is often ignored. In this paper, we develop a new MUI element detection dataset named MUI-zh and propose an Adaptively Prompt Tuning (APT) module to take advantage of discriminating OCR information. APT is a lightweight and effective module to jointly optimize category prompts across different modalities. For every element, APT uniformly encodes its visual features and OCR descriptions to dynamically adjust the representation of frozen category prompts. We evaluate the effectiveness of our plug-and-play APT upon several existing CLIP-based detectors for both standard and open-vocabulary MUI element detection. Extensive experiments show that our method achieves considerable improvements on two datasets. The datasets is available at github.com/antmachineintelligence/MUI-zh.

1. Introduction

While significant progress has been made in object detection [2, 17, 23, 24, 28], with the development of deep neural networks, less attention has been paid to its challenging variant in the Mobile User Interface (MUI) domain [1]. Instead of personal computers and books, people nowadays spend more time on mobile phones due to the convenience of various apps for daily life. However, there may exist some risks, including illegal gambling [10, 19], malware [31, 32], security [4, 8], privacy [14, 15], copy/fake [27] and fraudulent behaviors [6, 13] in apps, which need to be detected and alarmed as required by government authorities and app markets. In apps, these risks may occur in one element or even hide in the subpage after clicking one element. As a result, it is in great need of an accurate, robust, and even open-vocabulary MUI element detection approach in practice. Such technology can benefit a great variety of scenarios as mentioned above, towards building a better mobile ecosystem [13, 30].



{'category' : Text, 'bbox' : [239, 1013, 604, 54]},
{'category' : Text, 'bbox' : [168, 1155, 747, 85]},
{'category': TextButton, 'bbox': [393, 1312, 295, 107]},
{'category': UpperTaskBar, 'bbox': [1, 1, 1079, 63]}, ...

[[172, 411, 201, 427], '确定'],
[[195, 591, 252, 607], '￥15.00'],
[[192, 556, 276, 570], '木耳肉丝盖饭'],
[[50, 349, 79, 364], '输入'],
[[183, 307, 199, 319], '10'],
[[138, 269, 146, 269], '3'],
[[17, 129, 72, 144], '盖饭系列'],
[[312, 65, 340, 81] '预定'],
[[93, 129, 148, 144], '盖饭系列'],
[[195, 493, 252, 507], '￥15.00'],
[[192, 456, 277, 470], '泡椒肉丝盖饭'],
[[33, 235, 86, 247], '就餐人数'],
[[20, 92, 214, 104], '重庆市重庆市南岸区荣轩罐罐米线'],
[[181, 16, 213, 34], '搜索'], ...

{'category': Modal, 'bbox' :[19, 227, 338, 215]},
{'category': Button, 'bbox' : [18, 400, 339, 42]},
{'category': Product, 'bbox': [105, 545, 253, 71]},
{'category': Menu, 'bbox': [6, 123, 73, 29]},
{'category': Button, 'bbox': [301, 57, 52, 32]}, ...

VINS dataset

MUI-zh dataset

Figure 1. **Two MUI samples from VINS and MUI-zh dataset.** Compared to VINS, we additionally obtain the OCR descriptions as supplemental information in MUI-zh. Moreover, we further link OCR descriptions and element annotations with the same color.

This paper proposes MUI element detection as a variant object detection task and develops a corresponding dataset named MUI-zh. In general, object detection aims to classify and locate each object, such as an animal or a tool, in one raw image. While in MUI data, our primary concern is detecting elements, *e.g.*, products and clickable buttons in the screenshots. The main difference between the two tasks is that MUI data often have discriminative OCR descriptions as supplemental information for every element, significantly influencing detection results. To better explain it, we put two MUI data examples from VINS [1] and our MUI-zh in Figure 1. VINS only provides the category annotation and bounding box for every element, as the object detection dataset does. At the same time, MUI-zh additionally obtains the OCR descriptions and links them with elements for further usage. Since the OCR descriptions are texts and will be an additional input modality, it is natural to leverage recent Open-Vocabulary object Detection (OVD) models [3, 11, 20, 22, 36, 37, 40] as the MUI element detection baseline because of their rich vision-language knowledge learned from pretrained CLIP [21].

OVD detectors usually detect and classify objects by cal-

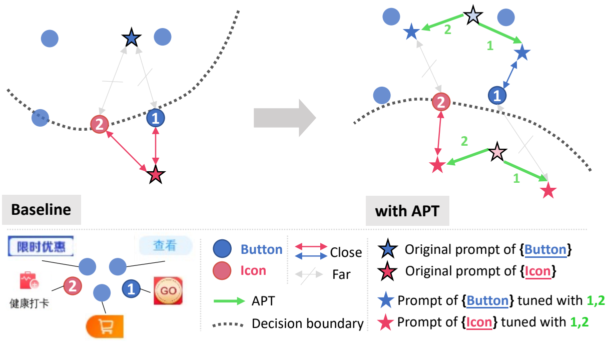

2
1
2
1
Baseline
with APT
限时优惠
查看
健康打卡
GO
Button
Icon
Close
Far
APT
Decision boundary
Original prompt of {Button}
Original prompt of {Icon}
Prompt of {Button} tuned with 1,2
Prompt of {Icon} tuned with 1,2

Figure 2. **Decision boundaries of baseline and adding APT during vision-language alignment.** The stars are category prompts, and the circles are element vision embeddings. Element 1 is misclassified by baseline while our APT tunes its category prompts adaptively and thus successfully matches it and its category.

culating the similarity between visual embeddings and textual concepts split from captions. However, according to our experiments, existing OVD methods can not achieve satisfactory performances on MUI datasets. The reason mainly comes from two aspects: Firstly, the samples for training OVD detectors are appearance-centric, while MUI data is not. Besides the appearance, the category of one MUI element is often closely related to its textual explanations obtained by OCR tools. Thus, OCR descriptions of one element can be viewed as a discriminative modality to distinguish itself from other categories, but neither exists nor is used in OVD models; Secondly, the category prompts with only category name is not optimal for vision-language alignment since they may not be precise enough to describe an MUI element. For example, we show four buttons (blue) and one icon (red) in Figure 2. The baseline (OVD detector) only uses "a photo of category name" to perform alignment and misclassify button 1 as an icon.

To alleviate the above issues, we propose a novel lightweight and plug-and-play Adaptively Prompt Tuning (APT) module in MUI element detection. Firstly, it takes OCR descriptions as input, using a unimodal block to obtain rich elements' information (*e.g.*, content and function) for vision-language alignment; Secondly, it adaptive encodes vision and OCR description features into embeddings to adjust the representation of frozen category prompts, which further reduces the impact of language ambiguity during matching. As shown in Figure 2, the gray dotted lines indicate the decision boundaries of the OVD baseline and its variant with APT during the recognizing phase. Element 1 is misclassified by the baseline since its embedding is close to the frozen category prompt of "icon" and far away from its groundtruth "button". Our APT adaptively tunes two category prompts (noted by the green arrow) for every element and successfully recognizes element 1. As a result, we demonstrate that the APT can achieve noticeable performance gains based on previous OVD detectors, which will benefit many mobile layout analyses [34, 35] and risk hunters [4, 10]. We summarize our contributions as follows.

- We develop a high-quality MUI dataset (called MUI-zh) containing 18 common categories with OCR descriptions as the supplemental information. Besides MUI-zh, we will also provide the OCR descriptions of the existing dataset VINS to facilitate future research.
- Inspired by the MUI data characteristics, we further proposed a novel Adaptive Prompt Tuning (APT) module to finetune category prompts for standard and open-vocabulary MUI element detection.
- Experiments on two datasets demonstrate that our APT, as a plug-and-play module, achieves competitive improvements upon four recent CLIP-based detectors.

2. Related Works

2.1. Object Detection

Object detection aims to detect and represent objects at a bounding box level. There are two kinds of object detection methods, *i.e.,* two-stage [2,24], and single-stage [17,23,28]. Two-stage methods first detect objects, then crop their region features to further classify them into the foreground or background. In contrast, the one-stage detectors directly predict the category and bounding box at each location.

2.2. Open-vocabulary Object Detection

Relying heavily on visual-language pretrained models [21], open-vocabulary object detection approaches aim to locate and classify novel objects that are not included in the training data. Recently, OVD methods [3,7,11,20,22,36,37, 40] follow two-stage fashion: class-agnostic proposals are firstly generated by RPN [24] trained on base categories, then the classification head is required to recognize novel classes with the knowledge from pretrained CLIP [21].

The representative solutions include OVR-CNN [33] and ViLD [11]. Taking Faster RCNN [24] as the backbone, OVR-CNN [33] trains a projection layer on image-text pairs with contrastive learning, while ViLD [11] proposes to explicitly distill the knowledge from the pretrained CLIP visual encoder. Advanced to them, Detic [40] tries to self-train the detector on ImageNet21K [25] for OVD. Recently, VL-PLM [36] use self-training in both two stages on unlabeled data and MEDet [3] proposes an online proposal mining method to refine the vision-language alignment. Following Detic [40], Object-centric OVD [22] combines knowledge distillation and contrastive learning, achieving the best performance on COCO [16] with the extra weakly supervised data from ImageNet21K. One closely related work is RegionCLIP [37], which leverages a CLIP model to match image regions with template texts on large-scale data from the web and then uses pseudo pairs to train the fine-grained alignment between image regions and text spans.

2.3. Prompts Learning

The large vision-language model, *e.g.*, CLIP [21], has significantly improved many few-shot or zero-shot computer vision tasks. They are often pretrained on a large amount of image-text pairs collected from the web and can be easily transferred to numerous downstream tasks with either finetuning [18, 26] or prompt learning [39]. From [21], we can observe that a task-specific prompt can boost performance significantly but needs carefully tuning prompts by humans. As its extension, CoOp [39] proposes context optimization with learnable vectors for automating prompt learning in few-shot classification, relieving the burden of designing hand-craft prompts by humans. Moreover, its further extension CoCoOp [38] learns a lightweight neural network to generate for each image an input-conditional token, which improves the generalization ability to wider novel categories in image classification tasks.

Recently, DetPro [7] and PromptDet [9] adapt CoOp [39] to OVD by designing particular strategies to handle foreground and background proposals within images. Although the vision embeddings learned in our APT are somehow inspired by CoCoOp [38], we are the first to propose a unified module for tuning prompts on two modalities, *i.e.*, OCR descriptions and vision features.

3. Mobile User Interface Dataset

In this section, we first introduce the existing MUI datasets and our developed MUI-zh. Then we briefly describe how to match OCR descriptions and elements.

3.1. Dataset Preparation

Early work on the MUI dataset explored how to support humans in designing applications. For example, Rico [5], a dataset of Android apps, was released five years ago. It consists of 72k MUI examples from 9722 apps, spanning 27 categories in the Google Play Store. However, the annotations of Rico are noisy, sometimes even incorrect, according to [1]. As its extension, VINS [1] uses MUI designs and wireframes to enable element detection and retrieval.

Nowadays, more and more tinyapps (in apps) are developed by merchants, and their elements, also as MUI data, have a noticeable domain gap with the elements in Rico and VINS. In order to fully understand MUI data, we develop MUI-zh, an MUI detection dataset from tinyapp screenshots. MUI-zh has 5769 images of tinyapp screenshots, including 50k elements within 18 categories. Besides element location and category, we also provide essential OCR descriptions and locations for every screenshot as supplemental information for classification. Another reason for developing MUI-zh is that the existing language of MUI datasets is English. Detectors trained on them can not be used in another language, such as Chinese, due to the domain gap/bias during vision-language alignment. Our MUI-zh collects high-quality tinyapp screenshots in Chinese, which enriches the MUI data for different languages.

3.2. OCR Descriptions Matching

After we collect and annotate enough MUI screenshots, we have to link the OCR descriptions and elements for further usage. How to relate OCR and elements with their locations is an open question. Intuitively, it is possible to link them by calculating and ranking their Intersection Over Union (IoU) according to two series bounding boxes inspired by non-maximum suppression (NMS). For every element box, we select the OCR boxes whose IoU scores are larger than a threshold (*e.g.*, 0.5) as its descriptions without replacement. Note that OCR tools may separate one sentence into many phrases, and as a result, an element may also be linked to more than one OCR description. Another special case is when an element box does not have any description, we assign it an empty word.

Generally speaking, IoU measures how much two proposals overlap and whether they can be assigned with the same instance in the object detection task. However, MUI elements like products and buttons are more likely to include their OCR descriptions (often occupy only a small region, *e.g.*, 10% of element) within the box. In this case, the IoU (0.1) is smaller than the threshold and this element fails to match its description, which is unacceptable. To tackle this problem, we utilize Intersection Over Minimum (IoM) instead of IoU during OCR matching. IoM replacing the area of union with the area of the minimum box in IoU is suitable for MUI data. For the case mentioned above, the IoM is 1, which means we successfully link the element and its OCR descriptions. Note that we also conduct OCR matching on VINS and release the results.

4. Methodology

In this section, we first briefly present how existing OVD models detect elements of MUI data in Section 4.1 and then show the architecture of APT and how it works in Section 4.2. Finally, we claim how to assemble APT on four existing detectors in Section 4.3. The whole pipeline of MUI element detection is shown in Figure 3.

4.1. Detectors on MUI

Preprocessing: Given a batch of MUI data, the training pipeline of recent two-stage CLIP-based detectors follows almost the same scheme (detect-then-classify). They first use a class-agnostic RPN [24] to obtain element proposals and perform their innovations and improvements during the

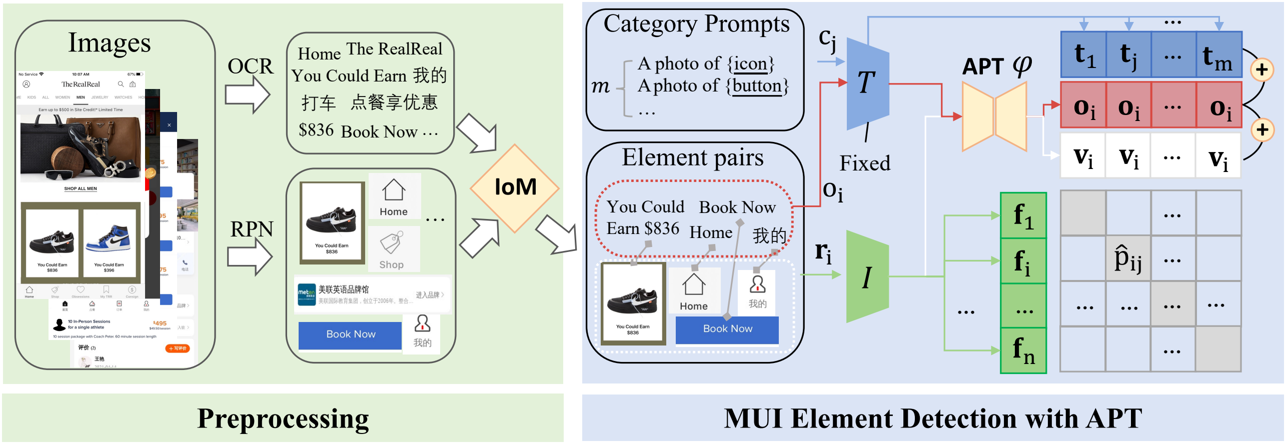


Figure 3. **Overview of MUI element detection pipeline associated with our proposed APT.** We first use OCR tools and class-agnostic RPN for input images to obtain OCR descriptions and element proposals. An IoM module matches and links the elements and OCR descriptions in the preprocessing phase. Existing CLIP-based detectors usually encode element proposals and category prompts into vision (green) and text (blue) embeddings by image encoder $\boldsymbol{I}$ and text encoder $\boldsymbol{T}$ for similarity calculation. Our APT additionally uses OCR descriptions (red) and vision embeddings to tune the text embedding for better alignment. Best viewed in color.

classification step. Our APT additionally considers OCR descriptions while aligning the element with categories.

Training: Existing CLIP-based detectors mainly focus on the training of the classifier. Specifically, they first construct human-made prompts (*e.g.*, "a photo of category name") and feed them to the frozen language encoder $\boldsymbol{T}(\cdot)$ of pre-trained CLIP [21] as the text embeddings. At the same time, a trainable CLIP visual encoder $\boldsymbol{I}(\cdot)$ is adapted to the detector for encoding element proposals into vision embeddings. Finally, the classifier learns to match these pair-wise embeddings via contrastive learning and cross-entropy loss.

Specifically, assuming an image I has n element proposals obtained by RPN, and we notate their features as $\{\mathbf{r}_i\}_{i=1}^n \in \mathcal{R}^d$. The classifier's goal is to match the element proposals with category prompts $\{c_j\}_{j=1}^m$ for m different categories. Relying on the powerful CLIP, the text (vision) embedding $\mathbf{t}_j$ ($\mathbf{f}_i$) of category j (proposal i) is generated by feeding c_j ($\mathbf{r}_i$) into the encoders, respectively:

$$\mathbf{t}_j = \boldsymbol{T}(c_j); \mathbf{f}_i = \boldsymbol{I}(\mathbf{r}_i). \tag{1}$$

For a paired proposal i and its groundtruth category j during training, we can calculate the predicted probability as:

$$p_{ij} = \frac{\exp(\cos(\mathbf{t}_j, \mathbf{f}_i)/\tau)}{\sum_{k=1}^m \exp(\cos(\mathbf{t}_k, \mathbf{f}_i)/\tau)}, \tag{2}$$

where τ is a temperature hyper-parameter. Finally, the cross-entropy loss is applied to optimize the network parameters except for $\boldsymbol{T}$ on proposal i:

$$L_i = -\log(p_{ij}). \tag{3}$$

The reason for freezing $\boldsymbol{T}$ is to fully utilize the knowledge learned by the CLIP pretrained on large-scale data according to [7, 22]. We also conduct experiments to verify it.

Inference: OVD models predict the category with the probability obtained by Equation 2 for the element detection task. While performing experiments in the open-vocabulary setting, we extend the category prompts to cover both base and novel classes following [37].

4.2. Adaptively Prompt Tuning

As we mentioned in Section 1, existing CLIP-based detectors are not generalizable to MUI categories due to the ignorance of OCR descriptions and the difficulty of aligning various-appearance elements to one frozen manual category prompts. To deal with these two weaknesses, we propose an Adaptively Prompt Tuning (APT) module by mapping OCR descriptions (red) and vision embeddings (green) into the space of text embeddings to adaptively tune the category prompts for every element proposal as shown in Figure 3. The figure shows that the mapped embeddings (red and white) are fused to adjust the frozen text embeddings (blue) for final alignment with vision embeddings (green).

For simplicity, we use $\varphi(\cdot)$ to denote the APT and formulate the training pipeline for image I as:

$$\mathbf{o}_i = \varphi(\boldsymbol{T}(o_i)); \mathbf{v}_i = \varphi(\mathbf{f}_i); \hat{\mathbf{t}}_{ji} = \mathbf{t}_j + \mathbf{o}_i + \mathbf{v}_i; \tag{4}$$

$$\hat{p}_{ij} = \frac{\exp(\cos(\hat{\mathbf{t}}_{ji}, \mathbf{f}_i)/\tau)}{\sum_{k=1}^m \exp(\cos(\hat{\mathbf{t}}_{ki}, \mathbf{f}_i)/\tau)}, \tag{5}$$

where $\{o_i\}_{i=1}^{n}$ are the OCR descriptions for n proposals. In this way, we can optimize the whole model except for $\boldsymbol{T}$ with cross-entropy loss:

$$\hat{L}_i = -\log(\hat{p}_{ij}). \tag{6}$$

Note that during inference, we also tune the text embeddings in the same way as training with Equation 4.

Since our goal is to map supplemental information into the embedding space for prompt tuning, it is natural to uniformly encode OCR descriptions and vision embeddings to encourage knowledge sharing and interaction from different modalities. As we know, APT is the first unimodal prompt tuning method, holding higher performances than individually encoding two modalities with different network parameters, as shown in our experiments.

Inspired by CoCoOp [38], we construct APT as a lightweight network with only two bottlenecks, which contains a fully-connected layer (fc) associated with a batch norm (bn) and a relu activation. It follows standard encoder-decoder fashion, and the fc is utilized to reduce/enlarge the number of feature channels (16x). Since the input channel of the visual feature is 1024, the total number of parameters of APT is about 128k, including the weights and bias, which have little influence on training and inference speed.

At the end of APT, we also explore how to fuse modality information in three ways: element-wise sum, element-wise multiply, and fusion with fc. Recall that the attention mechanism [29] is also influential in modality fusion and feature extraction. When we choose element-wise sum as the fusion function, our APT works as an attention layer for different modalities except for the self-attention part calculated on $\mathbf{t}_j$ in equation 4, which is a constant. If we use fc to learn the weights for fusion, then $\mathbf{t}_j$ can also be learned, which means our APT, in this case, has the same function of attention layers. According to our experiments, we use element-wise sum as the fusion function due to the slightly higher performance and lower calculating complexity.

In conclusion, we highlight that our goal of APT is *to adaptively tune frozen category prompts with the context from every element's OCR description and specific vision information*. Another interesting thing is that there exist many variants of APT. For example, what if we tune the category prompts only with vision embeddings and tune vision embeddings with OCR descriptions? Moreover, can we tune vision embeddings by self-attention and OCR descriptions while leaving category prompts fixed? To explore the influence of different tuning methods mentioned above, we conduct experiments in Section 5.3.

4.3. Assembling APT to CLIP-based Detectors

DetPro [7], PromptDet [9], Object-centric [22], and RegionCLIP [37] are recent CLIP-based frameworks for OVD. As we mentioned, our APT tunes category prompts without changing model architectures and thus can be used directly by many OVD methods. Here we explain how and where to equip them with APT in detail. Firstly, RegionCLIP [37] and Object-centric [22] use the fixed manual prompts, and we can easily add APT upon them at the end of the network during classification. For PromptDet [9] and DetPro [7], they both use the CoOp [39] to generate trainable category prompts instead of manual ones. Our APT adjusts that trainable category prompts for fair comparisons in this case.

5. Experiments

In this section, we first introduce the implementation details for datasets and models in Section 5.1. Our main results are APT upon CLIP-based detectors for both standard and open-vocabulary MUI element detection as shown in Section 5.2. Moreover, we evaluate the ablations to study model components in Section 5.3. Since the bounding boxes annotated by Rico [5] are noisy according to [1], we only conduct experiments on MUI-zh and VINS for comparison. Finally, we evaluate our APT for the object detection task on COCO [16] in Section 5.4.

5.1. Implementation Details

Datasets. We evaluate our method on two MUI element detection datasets, namely MUI-zh and VINS [1]. MUI-zh is a high-quality MUI element detection dataset with screenshots collected from mobile tinyapps. Its training set contains 4769 images and 41k elements, while the validation set has 1000 images and 9k elements within 18 categories. Another popular MUI dataset is VINS [1], which contains 3826 training and 981 validation images with 20 categories. For open-vocabulary element detection, we set the product, icon, button, card, tips, and menu as the base categories and the remaining 12 elements as novel ones on MUI-zh. As for VINS, we set background-image, card, text and spinner as four novel categories and others as base categories.
Training details and metrics. We evaluate MUI element detection performance on MUI-zh and VINS for both standard and open-vocabulary settings. During training, the default visual encoder of all models we used in the experiments is ResNet50 [12] from pretrained CLIP [21]. Note that the language encoder is frozen following [7, 22]. For MUI element detection, SGD is used with a batch size of 64, an initial learning rate of 0.002, and a maximum iteration of 12 epochs on 8 A100 GPUs. For open-vocabulary element detection, RPN is trained with the base categories of two datasets. The temperature τ is 0.01. The widely-used object detection metrics, including Mean Average Precision (mAP) for novel and all categories are used.

5.2. Main Results of MUI Element Detection

MUI element detection. As shown in Table 1, we list two groups of detection approaches on both MUI-zh and VINS.

Methods	Publication	MUI-zh	VINS
VINS [1]	CHI'21	-	63.21
Faster RCNN [24]	NeurIPS'15	44.63	68.89
Cascaded RCNN [2]	CVPR'18	46.76	72.85
+ COCO pretrain		48.80	75.77
DetPro [7]	CVPR'22	44.55	71.67
[7]+APT	-	48.62(+4.07)	77.73(+6.06)
PromptDet [9]	ECCV'22	40.14	68.94
[9]+APT	-	45.07(+4.93)	76.43(+7.49)
Object-centric [22]	NeurIPS'22	45.87	72.36
[22] +APT	-	50.78(+4.91)	79.48(+7.12)
RegionCLIP [37]	CVPR'22	45.51	71.53
[37]+APT	-	**51.23**(**+5.72**)	**80.84**(**+9.31**)

Table 1. **Results (mAP%) of MUI element detection.** We list the performance of six popular object detection approaches based on ResNet50. Besides them, we additionally report the performance gains of our APT module over four recent CLIP-based models.

The first group is standard object detection methods like Faster RCNN [24] and Cascaded RCNN [2], while the second group contains four CLIP-based models.

The table shows that recently proposed Object-centric OVD [22] and RegionCLIP [37] achieve much better performances than standard object detection models since MUI data need more attention on vision-language alignment. Moreover, our APT improves about 4-5% (6-9%) mAP on CLIP-based detectors on MUI-zh (VINS), which is a significant enhancement and shows APT's effectiveness in the MUI element detection task. Among these detectors, RegionCLIP [37] equipped with APT achieves the best performances (51.23% and 80.84% on MUI-zh and VINS).

Open-vocabulary MUI element detection. One more advantage of CLIP-based detectors compared to object detection ones is that they can detect objects not in the predefined categories. To this end, we also conduct experiments on open-vocabulary MUI element detection, and the results are in Table 2. Here we compare four recent methods with and without our APT on two datasets. The table shows that APT achieves noticeable improvements upon the listed methods. More specifically, among four CLIP-based methods, Object-centric OVD [22] with APT outperforms others on the MUI-zh, while RegionCLIP associated with APT gets the best performance on VINS.

Note that even though we have 80% (16/20) base categories on VINS, the performances of these methods on novel categories still need improvement compared to OVD detectors on COCO novel categories. There are mainly two reasons. Firstly, compared to COCO, the category names of MUI data have much less relation, which causes difficulty for knowledge transfer and embedding alignment. For example, the knowledge of recognizing cats can be easily transferred to classify dogs, while it is challenging to utilize the knowledge of recognizing cards for classifying icons in MUI data. For example, Drawer and Switch are two MUI categories in VINS. However, they have at least two meanings (polysemy words) with various appearances (real-world object and MUI element), making the transfer difficult. Another limitation is the size of the dataset. For open-vocabulary detection on COCO, there are thousands of (>110k) annotated object-caption pairs, not to mention numerous unpaired data from the web for self-supervised training. At the same time, the recent MUI datasets only have less than 10k samples for training.

Methods	MUI-zh		VINS	
	Novel(12)	All(18)	Novel(4)	All(20)
DetPro [7]	0.54	15.99	2.03	54.68
[7]+APT	1.41 (+0.87)	16.93 (+0.94)	2.74(**+0.71**)	54.95(+0.27)
PromptDet [9]	0.78	17.02	2.59	54.86
[9] +APT	1.83 (+1.05)	18.12(**+1.10**)	3.02 (+0.43)	55.18(+0.32)
Object-centric [22]	1.31	17.28	3.19	55.34
[22]+APT	**2.36**(**+1.05**)	**18.36**(+1.08)	3.76 (+0.57)	55.49(+0.15)
RegionCLIP [37]	1.06	17.34	3.61	55.79
[37] +APT	2.10 (+1.04)	18.23 (+0.89)	**4.23**(+0.62)	**56.80**(**+1.01**)

Table 2. **Results (mAP%) of open-vocabulary MUI element detection.** We report the performances of four CLIP-based methods on two datasets. Note that the number of novel categories of MUI-zh is 12, while VINS has four novel classes. Our APT improves the results of both novel and all categories.

Various APT architectures		MUI-zh	VINS
Ablation	APT($\mathbf{v}_i + \mathbf{o}_i$)	**51.23**	**80.84**
	w/o $\mathbf{o}_i$	47.96 (-3.27)	75.97(-4.87)
	w/o $\mathbf{v}_i$	48.91 (-2.32)	76.32(-4.52)
Weights	Share weights	**51.23**	**80.84**
	Individual weights	51.01(-0.22)	79.65 (-1.19)
Layers	2 (fc+bn+relu)	**51.23**	**80.84**
	3 (fc+bn+relu)	51.19 (-0.04)	80.79 (-0.05)
Tuning	$\mathbf{t}_j + \mathbf{v}_i + \mathbf{o}_i$ vs. $\mathbf{f}_i$	**51.23**	**80.84**
	$\mathbf{t}_j + \mathbf{o}_i$ vs. $\mathbf{f}_i + \mathbf{v}_i$	51.08(-0.15)	78.23 (-2.61)
	$\mathbf{t}_j + \mathbf{v}_i$ vs. $\mathbf{f}_i + \mathbf{o}_i$	50.23(-0.10)	78.35(-2.49)
	$\mathbf{t}_j$ vs. $\mathbf{f}_i + \mathbf{v}_i + \mathbf{o}_i$	51.00 (-0.23)	77.97 (-2.87)
Fusion	Element-wise sum	**51.23**	**80.84**
	Element-wise multi	48.83 (-2.40)	77.65 (-3.19)
	Attention(Concat + fc)	51.17 (-0.06)	80.59 (-0.25)
Encoder	Freeze $\boldsymbol{T}$	**51.23**	**80.84**
	Trainable $\boldsymbol{T}$	45.11(-6.12)	73.13(-7.71)

Table 3. **Ablation studies of APT and its variants.** We evaluate APT variants from several views, including their architecture, ablation and tuning methods.

5.3. Ablation Studies

We perform experiments for the ablation studies on two datasets. First, we show the impact of progressively integrating our two tuning modalities: the OCR descriptions $\mathbf{o}_i$ and vision embeddings $\mathbf{v}_i$, to the baseline RegionCLIP in Table 3. Then we explore different settings of weights, layers, tuning methods, and fusion functions, respectively.

Analysis for Components. As shown in Table 3, we first use only the vision embeddings $\mathbf{v}_i$ to tune the category prompts, which decreases about 3.3% (4.9%) mAP on

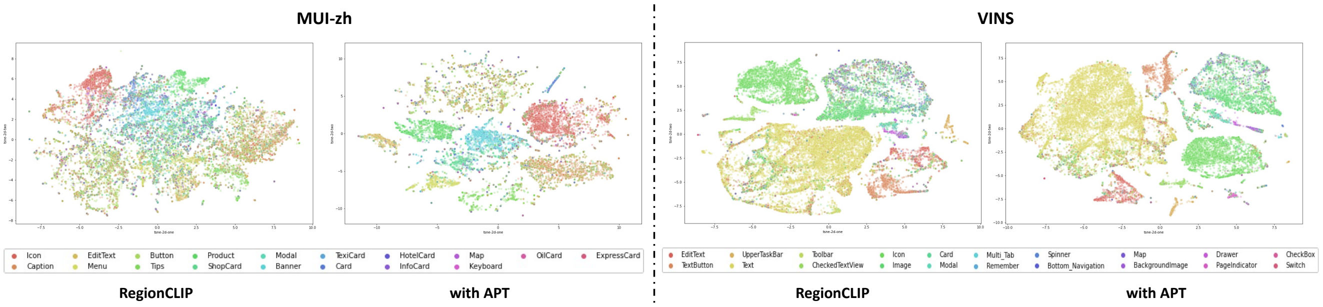


Figure 4. **T-SNE visualizations.** We perform RegionCLIP without and with APT on two datasets. T-SNE is utilized to visualize their region embeddings. It shows that APT contributes a lot to vision-language alignment. Best viewed in color and in-zoom.

Method	Novel (17)	Base (48)	Generalized(17+48)		
			Novel	Base	All
RegionCLIP [37]	35.2	57.6	31.4	57.1	50.4
[37] +APT($\mathbf{v}_i$)	**36.3**	57.3	**32.1**	57.2	50.0
[37] +APT($\mathbf{v}_i + \mathbf{o}_i$)	35.9	**57.7**	31.8	**57.3**	**50.6**

Table 4. **Results (mAP%) of OVD on COCO dataset.** We evaluate APT upon RegionCLIP (backbone ResNet50) following the standard base/novel split setting for a fair comparison.

MUI-zh (VINS). It means that the OCR descriptions of one element contribute a lot to its classification result. In the next row, we only equip RegionCLIP with OCR descriptions $\mathbf{o}_i$. Removing $\mathbf{v}_i$ leads to a 2.3% (4.5%) decrease in mAP for two datasets, which means adaptively tuning prompts according to the appearance is also crucial to the final performance. Overall, the whole improvements of APT upon baseline RegionCLIP indicate its effectiveness.

Analysis for weights sharing. Our APT is suitable for two different modalities, as verified in Table 3. We can observe that using a unified network for encoding OCR and vision embeddings is slightly better than two individual ones with the same architecture. The reason may be that OCR descriptions of one element often describe its appearance. Thus, a lightweight unified network can naturally map two modalities into one semantic space for prompt tuning.

Analysis for layers. Besides the weights of APT, we also want to explore its layer numbers. We compare two settings: 2 or 3 bottlenecks (fc+bn+relu) as presented in Table 3. With one extra layer, the network performance decreases a little. So our APT can be lightweight (only two layers with 128k parameters) and not time-consuming.

Analysis for tuning methods. An important part of APT is how to tune the prompts. Since our objective function is the similarity between category prompts and vision embeddings, there are four main ways: tuning only category prompts, tuning only vision embeddings and tuning both as shown in Table 3. We choose only to tune the category prompts with both OCR and vision embeddings, which gets the best performance. We believe the frozen category prompts rather than the trainable vision embeddings should be tuned adaptively in the MUI data domain.

Analysis for fusion functions. How to fuse the embeddings from different modalities also impacts element detection results. We compare element-wise sum, multiply and concentration with fc. Among them, the element-wise sum obtains the best performance with no extra parameters. We also believe employing element-wise sum for embedding fusion makes our APT work like an attention layer, excluding the self-attention part calculated on fixed text embeddings.

Analysis for text encoder. We also show the results of whether to freeze the text encoder $\boldsymbol{T}$ in this table. While we train $\boldsymbol{T}$ with MUI data, a large performance drop appears. As a result, we follow [7, 22] to freeze $\boldsymbol{T}$ in this paper.

5.4. Generalization on Object Detection

Although our APT is specially designed for MUI element detection with extra OCR information, it can also be modified to tune the category prompts on object detection tasks. To this end, we additionally conduct OVD experiments on COCO [16]. We follow the data split of [37] with 48 base categories and 17 novel categories and we also use the processed data from [37] with 110k training images and 4836 test images. Since objects in COCO usually have no OCR descriptions, we directly use their category names as the OCR descriptions, and thus we can build APT on RegionCLIP for the OVD task.

As shown in Table 4, our APT slightly outperforms RegionCLIP on all metrics (*e.g.*, 31.7 vs. 31.4 on novel categories) in the generalized setting. Compared with RegionCLIP in the standard OVD setting, our APT improves novel categories by about 0.7 mAP but only helps a little on the base categories. We find that the improvements in novel categories are larger than base ones, which indicates the effectiveness of APT in knowledge transfer. With these studies, we conclude that our APT positively impacts MUI element detection and object detection tasks.

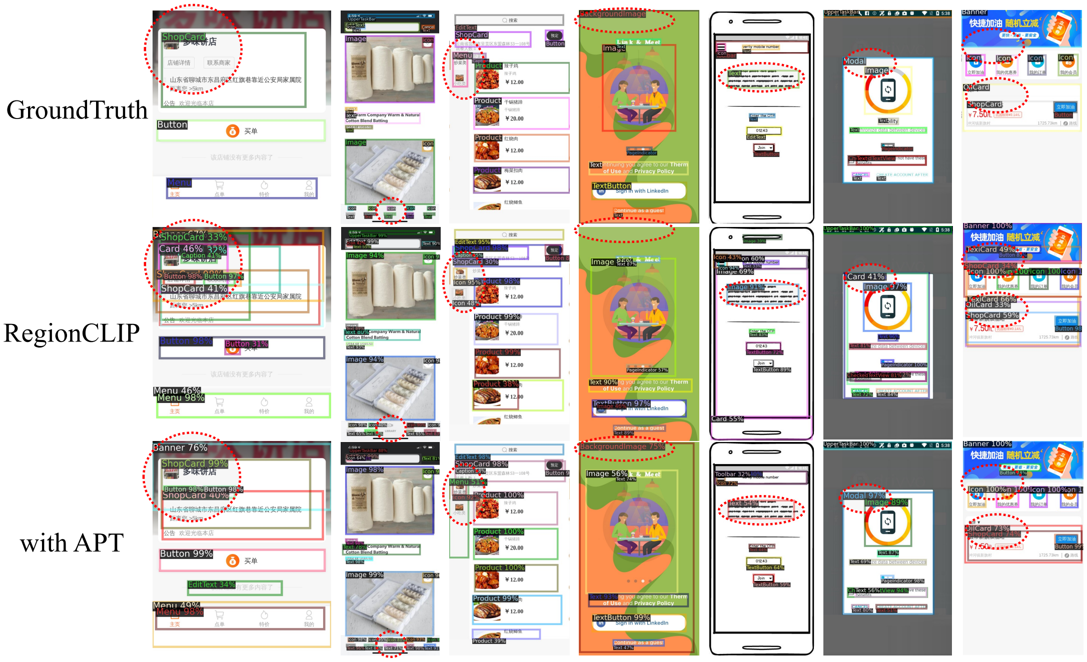


Figure 5. **Visualizations of MUI element detection.** We successively visualize the images and element bounding boxes of ground truth, RegionCLIP and with our APT. Note that we highlight the differences with the red dotted circle. Best viewed in color.

5.5. Visualizations

T-SNE plots of region vision embeddings. We have shown that APT can significantly improve performance over the baseline RegionCLIP. However, because the CLIP-based models implicitly learn the alignments by calculating similarity, it is interesting to see their region vision embeddings after training. We show the t-SNE plots of RegionCLIP and APT region embeddings (after non-linear dimensionality reduction) of MUI categories on two validation datasets in Figure 4. We can observe that APT promotes intra-class compactness and inter-class separation, which benefits the vision-language alignment. For example, in MUI-zh, our APT separates products and banners better than RegionCLIP. As for VINS, our model can successfully classify edittexts and textbuttons, while RegionCLIP can not.

Detection on MUI data. The detection visualizations of RegionCLIP and our APT on two MUI datasets are shown in Figure 5. We successively visualize the images, ground truth element boxes, RegionCLIP and ours. The red dotted circles in this figure highlight the differences. For example, RegionCLIP misclassifies texts in the fifth column and modal in the sixth column, while ours does not. It shows that our APT can better detect elements in MUI datasets.

6. Conclusion

In this work, we introduced APT, a lightweight and effective prompts tuning module for MUI element detection. Our APT contains two modality inputs, *i.e.*, element-wise OCR descriptions and visual features. They are fused and encoded within the APT to obtain embeddings for category prompt tuning. It significantly improves performance on existing CLIP-based models and achieves competitive results on two MUI datasets. We also released MUI-zh, a new MUI dataset with matched OCR descriptions. In summary, our model and dataset can benefit various real-world domains, such as robot interaction, information retrieval, targeted advertising, and attribute extraction on mobile phones. We hope our work could inspire designing new frameworks to tackle the challenging MUI element detection tasks.

Limitations. Our work has several limitations that can be further investigated. (1) The open-vocabulary capabilities of existing models on the MUI data could be further improved compared to the results on OVD datasets as mentioned in Section 5.2. (2) Existing methods all rely on the frozen language encoder from CLIP. We believe the performance drop of unfreezing the language encoder may be due to the small dataset size.

7. Acknowledgements

We thank the contribution of Qiangqiangzhu and his team Tinyapp Ecological Business Team in Ant Group for establishing MUI-zh dataset. Besides, we also thank the Business Risk Management—eKYB Team from the Ant Group for their help in this paper.

References

[1] Sara Bunian, Kai Li, Chaima Jemmali, Casper Harteveld, Yun Fu, and Magy Seif Seif El-Nasr. Vins: Visual search for mobile user interface design. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, 2021. 1, 3, 5, 6

[2] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *CVPR*, 2018. 1, 2, 6

[3] Peixian Chen, Kekai Sheng, Mengdan Zhang, Yunhang Shen, Ke Li, and Chunhua Shen. Open vocabulary object detection with proposal mining and prediction equalization. *arXiv preprint arXiv:2206.11134*, 2022. 1, 2

[4] Zhuo Chen, Jie Liu, Yubo Hu, Lei Wu, Yajin Zhou, Xianhao Liao, and Ke Wang. Illegal but not malware: An underground economy app detection system based on usage scenario. *arXiv preprint arXiv:2209.01317*, 2022. 1, 2

[5] Biplab Deka, Zifeng Huang, Chad Franzen, Joshua Hibschman, Daniel Afergan, Yang Li, Jeffrey Nichols, and Ranjitha Kumar. Rico: A mobile app dataset for building data-driven design applications. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*, 2017. 3, 5

[6] Feng Dong, Haoyu Wang, Li Li, Yao Guo, Tegawendé F Bissyandé, Tianming Liu, Guoai Xu, and Jacques Klein. Frauddroid: Automated ad fraud detection for android apps. In *Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pages 257–268, 2018. 1

[7] Yu Du, Fangyun Wei, Zihe Zhang, Miaojing Shi, Yue Gao, and Guoqi Li. Learning to prompt for open-vocabulary object detection with vision-language model. In *CVPR*, 2022. 2, 3, 4, 5, 6, 7

[8] Parvez Faruki, Ammar Bharmal, Vijay Laxmi, Vijay Ganmoor, Manoj Singh Gaur, Mauro Conti, and Muttukrishnan Rajarajan. Android security: a survey of issues, malware penetration, and defenses. *IEEE communications surveys & tutorials*, 17(2):998–1022, 2014. 1

[9] Chengjian Feng, Yujie Zhong, Zequn Jie, Xiangxiang Chu, Haibing Ren, Xiaolin Wei, Weidi Xie, and Lin Ma. Promptdet: Towards open-vocabulary detection using uncurated images. In *ECCV*, 2022. 3, 5, 6

[10] Yuhao Gao, Haoyu Wang, Li Li, Xiapu Luo, Guoai Xu, and Xuanzhe Liu. Demystifying illegal mobile gambling apps. In *Proceedings of the Web Conference 2021*, pages 1447–1458, 2021. 1, 2

[11] Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. Open-vocabulary object detection via vision and language knowledge distillation. *arXiv preprint arXiv:2104.13921*, 2021. 1, 2

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

[13] Yangyu Hu, Haoyu Wang, Ren He, Li Li, Gareth Tyson, Ignacio Castro, Yao Guo, Lei Wu, and Guoai Xu. Mobile app squatting. In *Proceedings of The Web Conference 2020*, pages 1727–1738, 2020. 1

[14] Eunhoe Kim, Sungmin Kim, and Jaeyoung Choi. Detecting illegally-copied apps on android devices. In *2013 International Conference on IT Convergence and Security (ICITCS)*, pages 1–4. IEEE, 2013. 1

[15] Jialiu Lin, Shahriyar Amini, Jason I Hong, Norman Sadeh, Janne Lindqvist, and Joy Zhang. Expectation and purpose: understanding users' mental models of mobile app privacy through crowdsourcing. In *Proceedings of the 2012 ACM conference on ubiquitous computing*, pages 501–510, 2012. 1

[16] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 2, 5, 7

[17] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *ECCV*, 2016. 1, 2

[18] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Advances in neural information processing systems*, 32, 2019. 3

[19] Qian Luo, Jiajia Liu, Jiadai Wang, Yawen Tan, Yurui Cao, and Nei Kato. Automatic content inspection and forensics for children android apps. *IEEE Internet of Things Journal*, 7(8):7123–7134, 2020. 1

[20] Zongyang Ma, Guan Luo, Jin Gao, Liang Li, Yuxin Chen, Shaoru Wang, Congxuan Zhang, and Weiming Hu. Open-vocabulary one-stage detection with hierarchical visual-language knowledge distillation. In *CVPR*, 2022. 1, 2

[21] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 1, 2, 3, 4, 5

[22] Hanoona Rasheed, Muhammad Maaz, Muhammad Uzair Khattak, Salman Khan, and Fahad Shahbaz Khan. Bridging the gap between object and image-level representations for open-vocabulary detection. In *NIPS*, 2022. 1, 2, 4, 5, 6, 7

[23] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, 2016. 1, 2

[24] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. 1, 2, 3, 6

[25] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. 2

[26] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. Vl-bert: Pre-training of generic visual-linguistic representations. *arXiv preprint arXiv:1908.08530*, 2019. 3

[27] Chongbin Tang, Sen Chen, Lingling Fan, Lihua Xu, Yang Liu, Zhushou Tang, and Liang Dou. A large-scale empirical

study on industrial fake apps. In *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*, pages 183–192, 2019. 1

[28] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *ICCV*, 2019. 1, 2

[29] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 5

[30] Nicolas Viennot, Edward Garcia, and Jason Nieh. A measurement study of google play. In *The 2014 ACM international conference on Measurement and modeling of computer systems*, pages 221–233, 2014. 1

[31] Haoyu Wang, Junjun Si, Hao Li, and Yao Guo. Rmvdroid: towards a reliable android malware dataset with app metadata. In *2019 IEEE/ACM 16th International Conference on Mining Software Repositories (MSR)*, pages 404–408. IEEE, 2019. 1

[32] Liu Wang, Ren He, Haoyu Wang, Pengcheng Xia, Yuanchun Li, Lei Wu, Yajin Zhou, Xiapu Luo, Yulei Sui, Yao Guo, et al. Beyond the virus: A first look at coronavirus-themed mobile malware. *arXiv preprint arXiv:2005.14619*, 2020. 1

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

[34] Haoran Zhang, Xuan Song, Yin Long, Tianqi Xia, Kai Fang, Jianqin Zheng, Dou Huang, Ryosuke Shibasaki, and Yongtu Liang. Mobile phone gps data in urban bicycle-sharing: Layout optimization and emissions reduction analysis. *Applied Energy*, 242:138–147, 2019. 2

[35] Mingming Zhang, Guanhua Hou, and Yeh-Cheng Chen. Effects of interface layout design on mobile learning efficiency: a comparison of interface layouts for mobile learning platform. *Library Hi Tech*, (ahead-of-print), 2022. 2

[36] Shiyu Zhao, Zhixing Zhang, Samuel Schulter, Long Zhao, Anastasis Stathopoulos, Manmohan Chandraker, Dimitris Metaxas, et al. Exploiting unlabeled data with vision and language models for object detection. *arXiv preprint arXiv:2207.08954*, 2022. 1, 2

[37] Yiwu Zhong, Jianwei Yang, Pengchuan Zhang, Chunyuan Li, Noel Codella, Liunian Harold Li, Luowei Zhou, Xiyang Dai, Lu Yuan, Yin Li, et al. Regionclip: Region-based language-image pretraining. In *CVPR*, 2022. 1, 2, 3, 4, 5, 6, 7

[38] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *CVPR*, 2022. 3, 5

[39] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348, 2022. 3, 5

[40] Xingyi Zhou, Rohit Girdhar, Armand Joulin, Philipp Krähenbühl, and Ishan Misra. Detecting twenty-thousand classes using image-level supervision. In *ECCV*, 2022. 1, 2