

A General Context Learning and Reasoning Framework for Object Detection in Urban Scenes

Xuan Wang¹ ^a, Hao Tang² ^b and Zhigang Zhu^{1,3} ^c

¹*The Graduate Center - CUNY, New York, NY 10016, U.S.A*

²*Borough of Manhattan Community College - CUNY, New York, NY 10007, U.S.A*

³*The City College of New York - CUNY, New York, NY 10031, U.S.A*

xwang4@gradcenter.cuny.edu, htang@bmcc.cuny.edu, zzhu@ccny.cuny.edu

Keywords: Deep Learning, Context Understanding, Convolutional Neural Networks, Graph Convolutional Network

Abstract: Contextual information has been widely used in many computer vision tasks. However, existing approaches design specific contextual information mechanisms for different tasks. In this work, we propose a general context learning and reasoning framework for object detection tasks with three components: local contextual labeling, contextual graph generation and spatial contextual reasoning. With simple user defined parameters, local contextual labeling automatically enlarge the small object labels to include more local contextual information. A Graph Convolutional Network learns over the generated contextual graph to build a semantic space. A general spatial relation is used in spatial contextual reasoning to optimize the detection results. All three components can be easily added and removed from a standard object detector. In addition, our approach also automates the training process to find the optimal combinations of user defined parameters. The general framework can be easily adapted to different tasks. In this paper we compare our framework with a previous multistage context learning framework specifically designed for storefront accessibility detection and a state of the art detector for pedestrian detection. Experimental results on two urban scene datasets demonstrate that our proposed general framework can achieve same performance as the specifically designed multistage framework on storefront accessibility detection, and with improved performance on pedestrian detection over the state of art detector.

1 INTRODUCTION

Contextual information has been widely used in many computer vision tasks. Context refers to any information that is related to the visual appearance of a target (an object or an event). Context can be in the form of visual or non-visual information. In object recognition task, recognizing a single object may be challenging sometimes when the object is out of context. But contextual information can provide crucial cues for the target. An example is shown in Fig 1, showing a mouse on a desk. In video based tasks, such as video action recognition and video event recognition, temporal context can help predict what will happen in the future. A walking person is visible in previous frame in a video, but he or she may become partially occluded in current frame because of a car or a tele-

graph pole is in front of the target person. When this happens, contextual information from nearby frames (previous or next) can help locate and detect the occluded target person in current frame. In object detection tasks, other objects can influence the presence of a target object in the same scene. These contextual information can indicate the co-occurrence of the objects and the location of the objects. For example, a painting should be on the wall, not on the ground. If we know there is a desktop on a table, there is higher probability that there are a keyboard and a mouse next to the desktop. Other contextual information, such as locations, dates and environments, etc. could potentially increase the likelihood of the presence of an object or an event. In this work, we propose a general framework that employs different contextual information such as local context, semantic context and spatial context among different objects in an urban scene for object detection tasks.

In object detection, a bounding box is the standard way to describe the spatial location of an ob-

^a  <https://orcid.org/0000-0003-4265-9375>

^b  <https://orcid.org/0000-0002-6197-0874>

^c  <https://orcid.org/0000-0002-9990-1137>

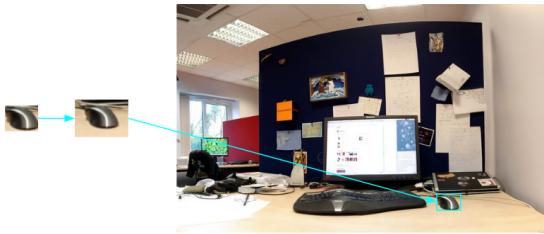


Figure 1: An example on the importance of contextual information for small object - A mouse next to a keyboard. From left to right: an isolated object, the object with a local context, and the object with a more global context.

ject. Large datasets, like MSCOCO (Lin et al., 2014) and ImageNet (Deng et al., 2009), use workers on Amazon’s Mechanical Turk (AMT) for crowdsourcing tasks. The quality of the labels is heavily relied on humans. Human labelers need to label the bounding box for an object by hand. Usually tight bounding boxes are fit to the target objects in order to maintain the label consistency. However, when an object is small, the tight bounding box may not provide sufficient local contextual information for recognition, sometimes even human observers cannot recognize the object because of the small size of the object. An example is shown in Fig. 1, we can barely recognize the object (a mouse), which is isolated from its context. When the local context (the area surrounding the mouse) is included, we can recognize the object as a mouse with less uncertainty. Furthermore, when we see the whole scene, we can easily recognize it is a mouse with a more global context even it is a small object in the image. A few pieces of work (Lim et al., 2021; Leng et al., 2021) show the contextual information from the surrounding areas of small objects provides critical clues for successful detection results. However, these researches utilize deep learning model training to extract and refine features from these small objects (Lim et al., 2021; Leng et al., 2021), which could increase computational cost potentially. In fact, one of the simplest ways to employ local context for small objects is to directly include their surrounding areas in the images that appear in, which directly provides contextual information for small objects. In this work, we apply an automatic local contextual labeling approach to enhance the original bounding boxes for small objects in order to employ local context *before* the model training step, by using the two most used definitions of small object in computer vision tasks.

Semantic context can also provide important information for detecting objects successfully. Without any visual cues, if we know the scene is at an urban street environment, we can easily guess there are

higher chance we shall detect pedestrians, bicycles, riders and cars, etc. The labels in the scene could provide prior knowledge of the co-occurrence relationship between labels. Several papers (Li et al., 2014; Li et al., 2016; Lee et al., 2018) show that a graph was proven to be very effective in modeling label correlation. Chen et al. (Chen et al., 2019) propose a framework to model the label dependencies for multi-label image recognition. Inspired by (Chen et al., 2019), we introduce a mechanism to allow an easy user configuration to automate the process for generating a contextual graph and searching the word embeddings from pretrained language model, for adapting the context learning model to various object detection/recognition tasks. A Graph Convolutional Network (GCN) (Kipf and Welling, 2016) learns over the contextual graph in our framework, to build a semantic space by using the word embeddings, and project the visual features extracted from the object detector into the semantic space for final classification.

Objects appear together, and they usually have spatial relations between each other in a real-world scene. For example, a keyboard and a mouse usually appear together and a mouse is probably appeared on the right side of the keyboard. Yang et al. (Yang et al., 2015) propose a Faceness-Net for face detection using spatial relation between face parts, such as the hair should appear above the eyes, and the nose should appear below the eyes, etc. Another work (Yang et al., 2019) proposes a spatial-aware network to model the relative location among different objects in a scene to boost the object detection performance. A few recent papers (Wang et al., 2022; Chakra and Zelek, 2022) uses specific spatial relations for storefront accessibility detection and scene graph generation. However, all these methods use specific, hard-coded spatial relations for their specific tasks, and these approaches cannot be easily generalized to other tasks without significant re-coding. In order to provide the generality of the spatial reasoning, topological relationships could be beneficial for modeling relations between different objects. In this work, we propose a more general approach to model the spatial relation between objects for object detection that can be used for different tasks. We utilize the user configuration mechanism to maximize the flexibility for object relation definition, without the modification of the code.

Different contextual information has been used in specific computer tasks, such as data augmentation (Dvornik et al., 2018), semantic reasoning during training (Zhu et al., 2021; Chen et al., 2019; Wang et al., 2022) and post processing (Fang et al., 2017; Wang et al., 2022), but there are lack of research on a general framework that can guide the context learn-

ing from data labeling, model training and post processing. In our previous work (Wang et al., 2022), we proposed a context learning framework for storefront accessibility detection through all these stages. However, the framework was designed with specific context learning mechanisms for storefront accessibility detection, and if we want to use it for a different task, we have to make significant changes in the code. In this work, we propose a general context learning and reasoning framework for various object detection tasks.

In summary, we propose a general context learning and reasoning framework for object detection of various tasks, which has three components: local contextual labeling (LCL), contextual graph generation (CGG) and spatial contextual reasoning (SCR). Local contextual labeling is applied to the objects that satisfy the definition of small objects. Contextual graph generation is applied to model the semantic relations of objects during training. Spatial contextual reasoning provides general spatial relations that could be used in different object detection tasks. The main contributions of this paper are:

- A general context learning and reasoning framework is proposed from data labeling, model training and spatial reasoning.
- An automated process is implemented for each component with simple user defined parameters.
- Each component can be applied individually and in combination, and it is easy to add and remove from a standard object detector.
- Our approach enables training automation to find the optimal user defined parameter combination.

The paper is organized as follows. Section 2 discusses related work. Section 3 proposes our general context learning and reasoning framework and describes each component in detail. Section 4 presents our experiments, including experimental setting (Section 4.1), dataset description (Section 4.2), experimental results (Section 4.3) and the ablation studies of our framework (Section 4.4). Section 5 provides a few concluding remarks.

2 RELATED WORK

2.1 Context Learning in Computer Vision

Context information has been widely used in many computer vision tasks. Many tasks, such as image

classification (Mac Aodha et al., 2019), object detection (Du et al., 2012; Fang et al., 2017; Sun and Jacobs, 2017; Zhu et al., 2016; Zhu et al., 2021; Wang et al., 2022), data augmentation (Dvornik et al., 2018), video event recognition (Wang and Ji, 2015; Wang and Ji, 2017) and video action detection (Yang et al., 2019), have employed different forms of contextual information. These context information includes local context (Dvornik et al., 2018; Du et al., 2012), global context (Zhu et al., 2016), semantic context (Wang and Ji, 2015; Wang and Ji, 2017), spatial context (Sun and Jacobs, 2017; Yang et al., 2019) and temporal context (Wang and Ji, 2015; Wang and Ji, 2017; Yang et al., 2019). Dvornik et al. (Dvornik et al., 2018) show that the environment surrounding the object provides crucial information about the correct location in order to augment useful dataset for minor object category. A serial work (Wang and Ji, 2015; Wang and Ji, 2017) use both semantic context and temporal context to build a hierarchical model to recognize events in videos. Fang et al. (Fang et al., 2017) use a knowledge graph to improve object detection performance. Sun et al. (Sun and Jacobs, 2017) use the spatial co-occurrence of curb ramps at intersection to detect missing curb ramps in urban environments. Although different context has been used widely in various computer vision tasks, to our best knowledge, there is no general framework available to guide context learning and reasoning over the whole deep learning process (data labeling, model training and post processing), and across different tasks. Our proposed framework employs different forms of context information through the entire deep learning process, and each component is easy to add and remove from an object detector.

2.2 Object Detection in Urban Scene

Many methods have been proposed for object detection in urban scene. These includes text detection and recognition (Du et al., 2012; Zhu et al., 2016), zebra crossing detection (Ahmetovic et al., 2015), curb detection (Cheng et al., 2018; Sun and Jacobs, 2017) and storefront accessibility detection (Wang et al., 2022). Du et al. (Du et al., 2012) and Zhu et al. (Zhu et al., 2016) focus on detecting text in a street environment. Cheng et al. (Cheng et al., 2018) propose a framework to detect road and sidewalk using stereo vision in the urban regions. Another work (Sun and Jacobs, 2017) aims to find missing curb ramps at street intersection in the city by using the pair-wise existence of the curb ramps. Our recent work (Wang et al., 2022) proposes a multi-stage context learning framework for storefront accessibility detection, by using

the specific relations between categories. All these researches are either lack of employing context information or using specifically designed context learning mechanisms. In this paper we propose a general context learning and reasoning framework which could be adapted to various object detection tasks.

Pedestrian detection is a special form of object detection task. Several papers (Cai et al., 2016; Zhang et al., 2017; Zhou and Yuan, 2018; Wu et al., 2020) are focused on pedestrian detection in urban scene. Convolutional Neural Networks(CNNs) have become the dominant approaches not only in object detection, but also in pedestrian detection. Although CNNs based pedestrian detectors have shown considerable progress, it is still challenging for detecting small-scale pedestrians and occluded pedestrians. A recent work (Wu et al., 2020) aims to improve the detection of small-scale pedestrians by enhancing the representations of small-scale pedestrians using the representation from large-scale pedestrians. To our best knowledge, none of these methods are employing local context for small-scale pedestrian detection and occluded pedestrian detection. Among these methods, Faster R-CNN (Ren et al., 2015) became the most popular framework that is deployed for pedestrian detection (Cai et al., 2016; Zhang et al., 2017; Zhou and Yuan, 2018; Wu et al., 2020). In this paper, we compare our proposed general context learning and reasoning framework with a baseline Faster R-CNN, which shows that our framework not only benefits the detection of small-scale pedestrians and occluded pedestrians by using contextual labeling, our proposed contextual components can also benefit each other and further improve the detection results.

3 Proposed Framework

Our proposed general context learning framework is shown in Fig.2. The overall framework includes three main components: local contextual labeling (LCL), contextual graph generation (CGG) and spatial contextual reasoning (SCR). Each component can be applied individually and in combinations to an object detector. We first utilize the local context for small objects in the local contextual labeling component (Section 3.1). In contextual graph generation component (Section 3.2), we automatically build the contextual co-occurrence graph using the prior label presence knowledge from training data, to describe object relations between different categories. Object categories are represented using word embeddings extracted from a pretrained language model (Pennington et al., 2014). Then the word embeddings are fed

into a Graph Convolutional Network (GCN) (Kipf and Welling, 2016) by learning the relations using contextual co-occurrence graph. Then we project the extracted region features from object detector into the semantic space built by the GCN. We further propose a spatial contextual reasoning component (Section 3.3) to optimize the detected candidates by using the general spatial relations between detected objects. Finally, we introduce a training automation mechanism (Section 3.4) for finding the optimal user defined parameter combinations. In the following, we will detail each component of our proposed general context learning and reasoning framework.

3.1 Local Contextual Labeling

First, we utilize surroundings of small objects as their *local context* in the Local Contextual Labeling component. In computer vision tasks, "small" objects are not very clearly defined. An small object, such as a spoon, can be a large object in the image because of the shooting angle, shooting environment, etc. (Fig. 3). In COCO dataset (Lin et al., 2014), small objects are defined as less than or equal to 32×32 pixels with a fixed image size of 640×480 . Another definition (Chen et al., 2017) of a small object is that objects are small when the overlap area between ground truth bounding box and the image is less than 0.58%. Because of the reliability and acceptance by other researchers, we use these two definitions of small objects as the standard for labeling automation. We extend the bounding box B of an object O in image I if the object satisfies with the COCO standard for a small object:

$$B'_O = \begin{cases} (1 + \alpha)B_O, & \text{if } B_O < 32 \times 32 \\ B_O, & \text{otherwise} \end{cases} \quad (1)$$

If the small object satisfies with the second standard - the Small Object Dataset (SOD) Standard (Chen et al., 2017), we extend the bounding box B of the object O in image I by:

$$B'_O = \begin{cases} (1 + \beta)B_O, & \text{if } \frac{B_O}{R_I} < 0.58\% \\ B_O, & \text{otherwise} \end{cases} \quad (2)$$

In the two equations above, B_O and B'_O denote the original and the updated bounding boxes, respectively, of the ground truth label for the small object. The parameters α and β are the extending factors (in percentage) from the original bounding boxes for the COCO standard and SOD standard, respectively. R_I is the resolution of the input image I . We provide flexibility for users to select a contextual labeling standard if the small objects satisfy with both definitions.

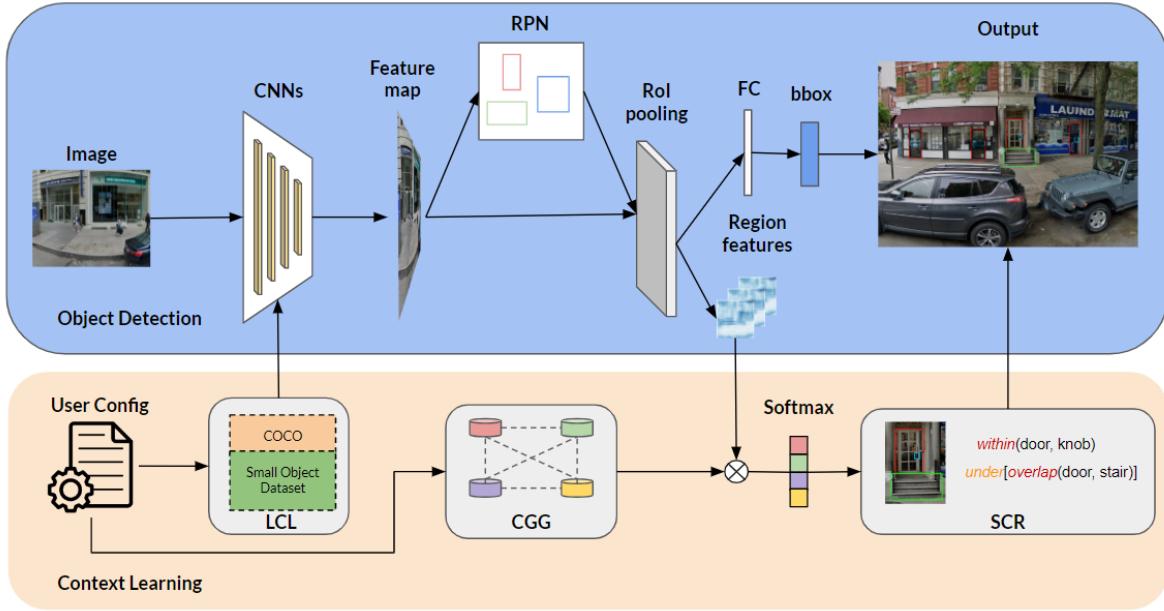


Figure 2: The overview of our general context learning and reasoning framework. Three contextual components: local contextual labeling (LCL), contextual graph generation (CGG) and spatial contextual reasoning (SCR). We design a user configuration mechanism for automating the process for various recognition tasks. Our contextual components can be applied individually and in combination. "⊗": dot product. "FC": Fully-connected layer.

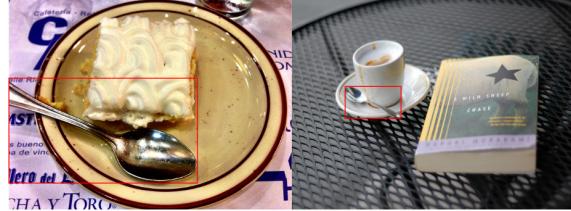


Figure 3: An example of a spoon in a image. The spoon is relatively large in left image but it is small in right image.

We keep both the original bounding boxes and the enlarged bounding boxes for all the small objects that satisfy with the user selected standard, in order to include local contextual information and improve the robustness of the detection. We will provide experimental settings in detail in Section 4.1.

3.2 Contextual Graph Generation

Graph Convolutional Network (GCN) (Kipf and Welling, 2016) has been used to model the semantic relationship between objects to solve different computer vision tasks, such as scene graph generation (Yang et al., 2018; Johnson et al., 2018) and image classification (Chen et al., 2019). A GCN takes feature description H of all nodes n and a contextual graph A to describe the relation between all nodes n . When a convolutional operation is applied, the func-

tion can be written as:

$$f(H, A) = \sigma(AHW) \quad (3)$$

where W and σ denote the weight and the non-linear activation function, respectively.

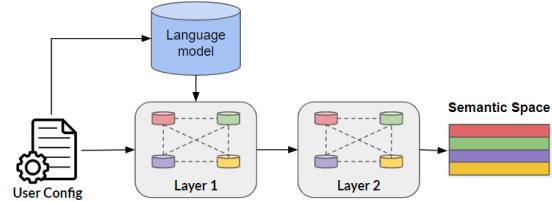


Figure 4: The visualization of CGG component.

Our Context Graph Generation component is shown in Fig 4. When our framework reads the category information from the user configuration, it first searches the word embeddings $H_{labels} \in \mathbb{R}^{n \times d}$ from a pretrained language model (Pennington et al., 2014) as the input of the GCN network, where n is the number of label categories and d is the dimension of the word embeddings. Then the contextual graph is automatically generated. The GCN learns a semantic relation over the contextual graph to build the semantic space. The generated semantic space from label feature representation is $H'_{labels} \in \mathbb{R}^{n \times D}$, where D is the dimension of the extracted region features from

the object detector. As shown in Fig 2, we project the region features $f_{regions} \in \mathbb{R}^{D \times N}$ into the semantic spaces H'_{labels} . The final output is:

$$\mathbf{P}_{regions} = softmax(H'_{labels} f_{regions}) \quad (4)$$

where $\mathbf{P}_{regions}$ represents the classification probability distribution for each proposed region, and $\mathbf{P}_{regions} \in \mathbb{R}^{n \times N}$.

In order to describe object relations between different categories, we use the label occurrence dependency in the form of conditional probability inspired by (Chen et al., 2019). $P(L_j | L_i)$ denotes the probability of occurrence of label L_j when label L_i appears. We automatically generate the contextual graph $A \in \mathbb{R}^{n \times n}$ between different categories by using the prior label occurrence knowledge from the training data, where n is the number of label categories. Note that a background label is also included to represent regions that do not belong to any of the categories.

3.3 Spatial Contextual Reasoning

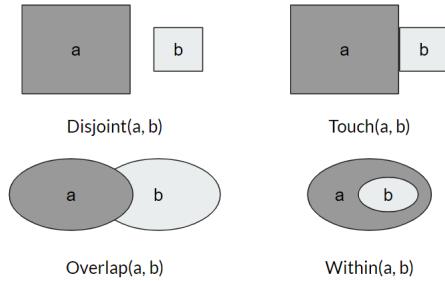


Figure 5: The visualization of common used topological relationships from (Clementini et al., 1993) and (Egenhofer and Franzosa, 1991).

Spatial relations between different objects, such as the object position and co-occurrence of the objects, have been encapsulated in spatial context. In order to provide the generality of the spatial reasoning, we use topological relationships to model relations between different objects. The topological relations can reveal the general relationship between a subject and object pair by using certain predicate, such as *above*, *under* and *within*, etc. The visualization of topological relationship is shown in Fig 5.

We use a predicate $pred$ to describe the directional relation between a subject and object pair $[S, O]$ along with the topological relationship t . The general relation R can be described as:

$$R[S, O] = pred[t(S, O)] \quad (5)$$

For example, in urban settings, a stair usually is located under a door. There might be overlaps or have

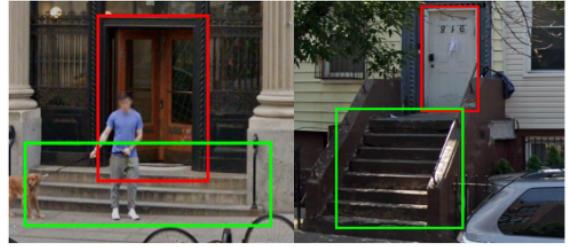


Figure 6: Visualization for stair-door relations in urban settings. Left: The label of stair is overlapped with the label of the door. Right: The stair has a spatial misalignment with the door.

spatial misalignment between the door and the stair. Examples are shown in Fig. 6. The general relationship for a door and a stair can be described using Eq. 5 as:

$$R[door, stair] = under[overlap(door, stair)] \quad (6)$$

Note that the general spatial relation is inversable between a subject-and-object pair, such as a door is above the stair and a stair is under a door. We further define a search area to search the detected object centroid with it if the object satisfies the condition in Eq. 5 with the detected subject. Then if the object is detected within the search area, we propose the detected object as a detection and send for evaluation. If multiple objects are detected in the search area, we propose the max score prediction to the evaluation. We provide the flexibility for users to configure the general spatial relation for the categories in their own dataset. The user-defined parameters are summarized in Table 1.

3.4 Training Automation

As mentioned in Section 3.1 and Section 3.3, we provide the capability for users to define contextual labeling thresholds and contextual spatial reasoning search areas. Although the deep learning network cannot automatically train these parameters, we enable a iterative training process in order to find the optimal user defined parameter combinations. The training pipeline is shown in Fig 7. Users can provide a threshold range for each parameter (i.e., labeling extension and search area), and set the number of training iterations for finding the optimal parameter combinations.

4 Experiments

In this section, we present our experiment results. We first describe the experimental settings (Section 4.1) in detail. We apply our framework on two datasets in

Table 1: Summary of the provided user-defined parameters for the spatial contextual reasoning component.

| Parameters | Definition |
|-------------------------------------|--|
| [Subject, Object] | Subject and object pair |
| pred (optional) | Directional relationships between subject and object |
| t | Topological relationship between subject and object |
| Overlap_threshold (optional) | The threshold of overlap percentage between subject and object |
| Search_height (optional) | The height of search area for object |
| Search_width (optional) | The width of search area for object |
| Labeling_standard | The standard for small object label enlargement |
| Enlarge_percentage | The enlarging percentage for small object labels |
| Relation_descriptor | The contextual graph generation method |

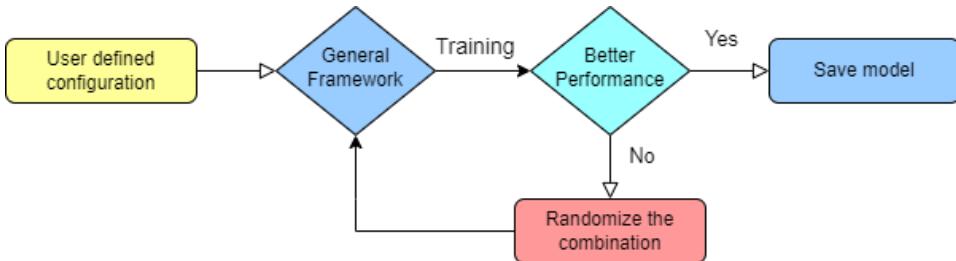


Figure 7: The training automation pipeline. We randomized the user defined parameter combinations and retrain the models.

urban scenes (Section 4.2), the Storefront Accessibility Image (SAI) Dataset, and the CityPersons Dataset, without any changes of the code. We compare the performance with a baseline object detector Faster R-CNN (Ren et al., 2015) and our previous multiCLU (Wang et al., 2022) framework for storefront accessibility detection. In experimental results (Section 4.3), we first compare the mean average precision (mAP) overall categories. We then compare the precision (%) and recall (%) for each category in the dataset. Furthermore, we provide results of our training automation for finding the optimal user defined parameters. Finally, we provide ablation analysis (Section 4.4) of each component based on the results of our contextual components individually and in combination for storefront accessibility detection. And we also demonstrate the generality of our framework by comparing the performance with baseline detector Faster R-CNN(Ren et al., 2015) for pedestrian detection, by using a urban scene pedestrian dataset (Zhang et al., 2017).

4.1 Experimental Settings

We use Faster R-CNN (Ren et al., 2015) as the underling detector for both storefront accessibility detection and pedestrian detection. We adopt ResNet-50 (He et al., 2016) and Feature Pyramid Network (FPN) (Lin et al., 2017) as the backbone feature extractor, which is pretrained on the COCO dataset. Our GCN model for contextual graph generation consists of two

layers with the output dimension of 1024. LeakyReLU (Maas et al., 2013) is used as the activation function for GCN. We use 300-dim word embeddings from GloVe (Pennington et al., 2014) as the input label feature vector for GCN model. Stochastic Gradient Descent (SGD) is used as the optimizer during training. The momentum is set to 0.95 and the weight decay is set to 1e-4, respectively. The initial learning rate is set to 0.005, and it drops by 0.25 for every 8 epochs. The total training epochs is 40 in total for storefront accessibility detection and 60 for pedestrian detection. In order to compare the performance between our proposed framework with our previously designed MultiCLU (Wang et al., 2022), we initially use the same settings as described in (Wang et al., 2022). We use the Small Object Dataset(SOD) standard to enlarge the labels for small objects in the SAI dataset. The enlarge percentage is set to 15 percent (i.e., $\beta=0.15$). we use the same small object standard for CityPersons dataset and the enlarge percentage is set to 10 percent (i.e., $\beta=0.10$). The configurations of general spatial contextual reasoning for both tasks are shown in Table 2.

4.2 Dataset Description

Storefront Accessibility Image Dataset. We use the SAI dataset described in (Wang et al., 2022), which consists of 3 main categories (doors, knobs, stairs) for storefront accessibility in an urban environment. The dataset is collected from Google Street View

Table 2: Default user parameter settings for our experiments on the SAI Dataset(Wang et al., 2022) and the CityPersons Dataset (Zhang et al., 2017). O_T: Overlap_threshold.

| Task | [Subject, Object] | Predicate | Topology | O_T | Search_area_height | Search_area_width |
|-------------|-------------------|-----------|----------|-----|-------------------------------------|--------------------------------|
| SAI | [door, knob] | - | within | - | - | - |
| | [door, stair] | under | overlap | 0.2 | $0.2height_{door} + height_{stair}$ | $width_{door} + width_{stair}$ |
| CityPersons | [person, bicycle] | under | overlap | 0.4 | $0.5height_{person}$ | $width_{bicycle}$ |

Table 3: Results on recall(%), precision(%) and F1 score (%) per category for various combinations of the three general contextual components, compared with a baseline and a previous methods on the SAI dataset. The best results are in bold and the second best underlined.

| Model | Precision ↑ | | | Recall ↑ | | | mAP ↑ | Recall ↑ | F1 Score ↑ |
|--------------------------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| | Door | Knob | Stair | Door | Knob | Stair | | | |
| Faster R-CNN(Ren et al., 2015) | 75.6 | 17.7 | 66.0 | 87.5 | 47.6 | 73.1 | 53.1 | 69.4 | 60.2 |
| MultiCLU (Wang et al., 2022) | 75.6 | <u>51.2</u> | 70.0 | 92.3 | 80.4 | 83.0 | 66.4 | 85.2 | 74.6 |
| +LCL | 78.1 | 41.3 | 66.8 | 88.9 | 77.7 | 74.5 | 62.1 | 80.4 | 70.1 |
| +CGG | 78.0 | 19.0 | 68.5 | 90.1 | 53.0 | 79.4 | 55.2 | 74.2 | 63.3 |
| +SCR | 77.8 | 18.6 | 67.2 | 88.8 | 52.4 | 74.5 | 54.5 | 71.9 | 62.0 |
| +LCL+CGG | 78.4 | 50.0 | 69.2 | 90.8 | 75.0 | 79.4 | 65.9 | 81.7 | <u>73.0</u> |
| +CGG+SCR | 78.2 | 21.2 | 69.6 | 90.3 | 55.8 | 80.8 | 56.3 | 75.6 | 64.5 |
| +LCL+SCR | 79.2 | 41.2 | 67.8 | 89.2 | 77.8 | 74.5 | 62.7 | 80.5 | 70.5 |
| Proposed Framework (C3) | <u>78.2</u> | 52.3 | <u>69.6</u> | <u>92.0</u> | <u>79.9</u> | <u>82.3</u> | 66.7 | <u>84.7</u> | 74.6 |



Figure 8: The label example from SAI dataset(Wang et al., 2022). Red: Ground truth label of the door. Cyan: Ground truth label of the knob. Green: Ground truth label of the stair.

of New York city using Google Street View API. The final dataset is central cropped images from the panorama images in which the storefronts are clearly seen. Overview of the SAI dataset is shown in Table 4. There are 1102 images in total, with 992 images in the training set and 110 images in the testing set. An example of labeled storefront objects is shown in Fig 8.

CityPersons Dataset. The CityPersons dataset is a subset of Cityscapes (Cordts et al., 2016), which only consists of person annotations. The dataset has four categories: pedestrian, rider, sitting person and person(other). Overview of the dataset is shown in Table 4 as well. An example of labeled pedestrians is shown in Fig 9.

Table 4: Overview of two urban scene datasets: SAI Dataset and CityPersons Dataset. #C: Number of Categories; #T: Number of Samples in the Training Set; #V: Number of Samples in the Validation Set

| Datasets | #C | #T | #V |
|----------------------------------|----|------|-----|
| SAI (Wang et al., 2022) | 3 | 992 | 110 |
| CityPersons (Zhang et al., 2017) | 4 | 2975 | 500 |



Figure 9: The label example from CityPersons Dataset (Zhang et al., 2017). Red: Pedestrian. Blue: Rider. Yellow: Sitting person.

4.3 Experimental Results

Comparison with baseline Faster R-CNN(Ren et al., 2015) and MultiCLU(Wang et al., 2022). We first compare our proposed general context learning framework with the baseline detector Faster R-CNN (Ren et al., 2015) and the previous proposed MultiCLU (Wang et al., 2022) on the SAI dataset. We measure the mean average precision (mAP) and recall over standard 0.5 IoU threshold. The results (Table 3) show that our proposed framework gain

great improvement over Faster R-CNN on both mAP (+13.6%) and recall (+15.3%). Our general context framework also achieves slight better performance on mAP (+0.3%) but a slightly lower performance on recall (-0.5%) comparing with the MultiCLU framework with specially designed context mechanisms. Overall, our general context framework achieves the same performance as the specifically designed MultiCLU on F1 score, which is a 14.4% increase from the baseline model Faster R-CNN.

Comparison between various contextual components. We further compare the performance between the different combination of three contextual components. We measure the small objects in the SAI dataset using the same approach in (Wang et al., 2022). If local contextual labeling is enabled, we use both original labels and enlarged labels for small objects that satisfy the standard. If both labels are detected for same small object, we only count one to avoid duplicated detection. The results are shown in Table 3. When only apply a single contextual component, The recall is improved over baseline from 2.8% to 11%, and mAP is improved from 1.4% to 9%. We can also observe that when applying a single contextual component, local contextual labeling has greater impact than the other two components.

When combinations of two contextual components are applied, all combinations surpass the performance over the baseline detector, from +3.2% to 12.8% on mAP and 6.2% to 12.3% on recall. When the combinations have the Local Context Labeling (LCL) component, they outperform the other combinations with large margins on both mAP (+6.4% to 9.6%) and recall (+4.9% to 6.1%). The results indicates that including the contextual information surround the small objects can aid the successful detection of the small objects, the important doorknobs in this example. Furthermore, when comparing with the single LCL component, both Context Graph Generation (CGG) and Spatial Context Reasoning (SCR) have positive impact, and they further improve the results over a single LCL component on both mAP and recall. When comparing between applying both CGG and SCR and applying them individually, the combination slightly improves both mAP and recall over the single CGG and single SCR component. When all three components (C3) are applied, our proposed framework achieves the best result, with 13.6% improvement on mAP and 15.3% improvement on recall. We can also observe that our general framework improves mAP on all categories over MultiCLU(Wang et al., 2022), but with only very slightly decreased recall. This indicate that the specific designed MultiCLU could introduce more false posi-

tives than correct predictions, hence our framework achieves better precision and slightly worse recall.

Table 5: Result comparison on training automation to find the optimal user defined parameters combination on SAI dataset.

| (Enlarge/Overlap/heights) | mAP ↑ | Recall ↑ |
|---------------------------|-------------|-------------|
| Default(0.15, 0.2, 0.2) | 66.7 | 84.7 |
| (0.16, 0.17, 0.16) | 68.4 | 85.7 |
| (0.08, 0.12, 0.18) | 66.3 | 83.4 |
| (0.09, 0.06, 0.17) | 65.9 | 83.9 |
| (0.14, 0.11, 0.13) | 66.5 | 83.1 |
| (0.19, 0.13, 0.11) | 66.1 | 83.6 |

Results on finding optimal combination of user defined parameters. We further use iterative training to find the optimal combinations of user defined parameters. As described in MultiCLU(Wang et al., 2022), the default enlarge percentage for the small object label is 15%, the overlap between door and stair is 20% and the search area height is 20% height of detected door plus the height of detected stair. We further random select the combinations by setting a threshold range for each parameter. We set [0.05, 0.2] as the threshold for both enlarge percentage and overlap percentage. [0.1, 0.2] is set as the threshold for the height of subject (door). We keep 2 decimals for the random selection and iterate the training for 6 times. The results (Table 5) show that when enlarge percentage is below 15 percent, both mAP and recall decrease comparing with default settings. The results also show that it is possible to find a better combination of user defined parameters comparing with default parameter settings.

Comparison on pedestrian detection with various combinations of contextual components. We further evaluate our general context learning and reasoning framework on pedestrian detection, by comparing with the baseline detector Faster R-CNN(Ren et al., 2015), without any change in coding. We first compare the evaluation results on reasonable and heavy subsets of the data with the baseline using the standard evaluation metric in pedestrian detection MR^{-2} (the lower, the better). These subsets are defined as: Reasonable: $h \in [50, \infty]$, $v \in [0.65, 1]$; Heavy: $h \in [50, \infty]$, $v \in [0, 0.65]$, where h and v denote the height and visible ratio of pedestrians, respectively. When only apply the single LCL component, the performance improve 1.1% on the reasonable subset and 1.7% on the heavy subset (Table 6). We further add in the fine-grained category (rider) in CityPersons dataset during training in order to enable the CGG and SCR components. Similar to the SAI detection results, when the combinations have

Table 6: Comparisons with the baseline detector on the CityPersons validation set.

| Model | LCL | CGG | SCR | Reasonable ↓ | Heavy ↓ |
|---------------------------------|-----|-----|-----|--------------|-------------|
| Faster R-CNN (Ren et al., 2015) | - | - | - | 13.4 | 36.9 |
| Single Component | ✓ | - | - | 12.3 | 35.6 |
| | - | ✓ | - | 13.3 | 37.1 |
| | - | - | ✓ | 13.0 | 36.5 |
| Two Components | ✓ | ✓ | - | 12.2 | 35.2 |
| | - | ✓ | ✓ | 13.2 | 36.5 |
| | ✓ | - | ✓ | 12.0 | 36.0 |
| Proposed framework (C3) | ✓ | ✓ | ✓ | 12.0 | 35.2 |

the LCL component, the result is better than the other combinations. Both CGG and SCR have minor impact on pedestrian detection,. It might because the low correlation between pedestrians and other objects in urban scene. Overall, our proposed framework with all the three components achieves the best performance on both the reasonable subset (-1.4%) and the heavy subset (-1.7%), comparing with the baseline detector and other combinations.

4.4 Ablation Studies

We further studied the contribution of each general contextual component. As we applied various combinations of three contextual components, We can clearly see the impact of each component from Table 3 and Table 6, for both storefront accessibility detection and pedestrian detection.

Before sending an image into the detector, local contextual labeling uses selected standard definitions of small objects to automatically expand the ground truth label, in order to include local contextual information for small objects for the network to learn. Our evaluation results show that enough local contextual information has great impact on small objects. We can observe that when LCL is applied, the framework gain great improvement on both mAP (12.8%) and recall (12.3%) for the SAI dataset. Although there is no great improvement for pedestrian detection, LCL also shows greater impact than the other two components.

We apply contextual graph generation during the network training. We use word embeddings from pre-trained language model and the contextual graph generated from prior knowledge from the training set as the input of Graph Convolutional Network. The GCN learned over the word embeddings and the contextual graph to build a semantic space. We then project the region features extracted from object detector into the semantic space for final prediction of each region. As the result shown in Table 3 and Table 6. The CGG component does not have the same impact as the LCL component for the SAI dataset, and even smaller for

pedestrian detection. This might because the contextual graph is using the prior co-occurrence knowledge from the training set between categories, where SAI dataset has higher correlated categories compare to CityPersons dataset.

We further propose a spatial contextual reasoning (SCR) component, using general topological relationships to model the relations between subject and object pairs. As shown in Table 3, although the performance has slight improvement when applying to the baseline detector, the SCR component can benefit the other two components when apply in combinations. The SCR also has a minor impact on pedestrian detection task. It might because the general spatial reasoning have minor impact on the pedestrians even when the fine-grained category is added for spatial reasoning. It could also because the lack of data on fine-grained categories for CityPersons dataset. Our proposed framework exhibited improvement over any other combinations, The result also shows that contextual components can benefit from each other, hence maximize the performance over the baseline.

5 CONCLUSION

In this work, we proposed a general context learning and reasoning framework. We compared our framework for the storefront accessibility detection task and the pedestrian detection task, with a baseline detector Faster R-CNN (Ren et al., 2015) and our previously proposed context learning framework particularly designed for storefront accessibility detection(Wang et al., 2022). Our new general framework can apply to various visual tasks without any changes, and in a general manner, guide context learning from data labeling, contextual graph during training and general spatial reasoning during post processing. Our results show that our proposed framework applied to the same storefront data can achieve same performance as the previous context learning framework specifically designed for storefront accessibility

detection. The results also show that the framework, when applying to a different dataset CityPersons, can achieve better performance over the baseline detector for pedestrian detection. We demonstrate that our contextual components can be applied individually and in combinations, and easily add and remove from the object detector. In future works, the effectiveness of the contextual components in various visual detection tasks will be investigated, which could be more generalized and adaptive for other different visual detection tasks. We will also investigate how to better model relations between visual context and non-visual context. We hope our work could provide a generalized approach on guiding context learning in real world applications so adapting to different tasks would be more efficient.

ACKNOWLEDGEMENTS

The work is supported by NSF via the Partnerships for Innovation Program (Award #1827505) and the CISE-MSI Program (Award #1737533), AFOSR Dynamic Data Driven Applications Systems (Award #FA9550-21-1-0082), and ODNI via the Intelligence Community Center for Academic Excellence (IC CAE) at Rutgers University (Awards #HHM402-19-1-0003 and #HHM402-18-1-0007).

REFERENCES

- Ahmetovic, D., Manduchi, R., Coughlan, J. M., and Mascetti, S. (2015). Zebra crossing spotter: Automatic population of spatial databases for increased safety of blind travelers. In *Proceedings of the International ACM SIGACCESS Conference on Computers and Accessibility*, pages 251–258.
- Cai, Z., Fan, Q., Feris, R. S., and Vasconcelos, N. (2016). A unified multi-scale deep convolutional neural network for fast object detection. In *Proceedings of the European Conference on Computer Vision*, pages 354–370. Springer.
- Chakra, D. A. and Zelek, J. (2022). The topology and language of relationships in the visual genome dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4860–4868.
- Chen, C., Liu, M.-Y., Tuzel, O., and Xiao, J. (2017). R-cnn for small object detection. In Lai, S.-H., Lepetit, V., Nishino, K., and Sato, Y., editors, *Computer Vision – ACCV 2016*, pages 214–230, Cham. Springer International Publishing.
- Chen, Z.-M., Wei, X.-S., Wang, P., and Guo, Y. (2019). Multi-label image recognition with graph convolutional networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5177–5186.
- Cheng, M., Zhang, Y., Su, Y., Alvarez, J. M., and Kong, H. (2018). Curb detection for road and sidewalk detection. *IEEE Transactions on Vehicular Technology*, 67:10330–10342.
- Clementini, E., Felice, P. D., and Oosterom, P. v. (1993). A small set of formal topological relationships suitable for end-user interaction. In *International Symposium on Spatial Databases*, pages 277–295. Springer.
- Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., and Schiele, B. (2016). The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 277–295. Springer.
- Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.
- Du, Y., Duan, G., and Ai, H. (2012). Context-based text detection in natural scenes. In *Proceedings of the IEEE International Conference on Image Processing*, pages 1857–1860. IEEE.
- Dvornik, N., Mairal, J., and Schmid, C. (2018). Modeling visual context is key to augmenting object detection datasets. In *Proceedings of the European Conference on Computer Vision*, pages 364–380.
- Egenhofer, M. J. and Franzosa, R. D. (1991). Point-set topological spatial relations. *International Journal of Geographical Information System*, 5(2):161–174.
- Fang, Y., Kuan, K., Lin, J., Tan, C., and Chandrasekhar, V. (2017). Object detection meets knowledge graphs. In *Proceedings of the International Joint Conferences on Artificial Intelligence*.
- He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 770–778.
- Johnson, J., Gupta, A., and Fei-Fei, L. (2018). Image generation from scene graphs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1219–1228.
- Kipf, T. N. and Welling, M. (2016). Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.
- Lee, C.-W., Fang, W., Yeh, C.-K., and Wang, Y.-C. F. (2018). Multi-label zero-shot learning with structured knowledge graphs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1576–1585.
- Leng, J., Ren, Y., Jiang, W., Sun, X., and Wang, Y. (2021). Realize your surroundings: Exploiting context information for small object detection. *Neurocomputing*, 433:287–299.
- Li, Q., Qiao, M., Bian, W., and Tao, D. (2016). Conditional graphical lasso for multi-label image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 248–255.

- on Computer Vision and Pattern Recognition*, pages 2977–2986.
- Li, X., Zhao, F., and Guo, Y. (2014). Multi-label image classification with a probabilistic label enhancement model. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, volume 1, pages 1–10.
- Lim, J.-S., Astrid, M., Yoon, H.-J., and Lee, S.-I. (2021). Small object detection using context and attention. In *Proceedings of the International Conference on Artificial Intelligence in Information and Communication*, pages 181–186.
- Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., and Belongie, S. (2017). Feature pyramid networks for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2117–2125.
- Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In *Proceedings of the European Conference on Computer Vision*, pages 740–755.
- Maas, A. L., Hannun, A. Y., and Ng, A. Y. (2013). Rectifier nonlinearities improve neural network acoustic models. In *Proceedings of the ICML Workshop on Deep Learning for Audio, Speech and Language Processing*.
- Mac Aodha, O., Cole, E., and Perona, P. (2019). Presence-only geographical priors for fine-grained image classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9596–9606.
- Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1532–1543.
- Ren, S., He, K., Girshick, R. B., and Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39:1137–1149.
- Sun, J. and Jacobs, D. W. (2017). Seeing what is not there: Learning context to determine where objects are missing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5716–5724.
- Wang, X., Chen, J., Tang, H., and Zhu, Z. (2022). Multiclu: Multi-stage context learning and utilization for storefront accessibility detection and evaluation. In *Proceedings of the International Conference on Multimedia Retrieval*, ICMR ’22, page 304–312, New York, NY, USA. Association for Computing Machinery.
- Wang, X. and Ji, Q. (2015). Video event recognition with deep hierarchical context model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4418–4427.
- Wang, X. and Ji, Q. (2017). Hierarchical context modeling for video event recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(9):1770–1782.
- Wu, J., Zhou, C., Zhang, Q., Yang, M., and Yuan, J. (2020). Self-mimic learning for small-scale pedestrian detection. In *Proceedings of the 28th ACM International Conference on Multimedia*, MM ’20, page 2012–2020, New York, NY, USA. Association for Computing Machinery.
- Yang, J., Lu, J., Lee, S., Batra, D., and Parikh, D. (2018). Graph r-cnn for scene graph generation. In *Proceedings of the European Conference on Computer Vision*, pages 670–685.
- Yang, S., Luo, P., Loy, C.-C., and Tang, X. (2015). From facial parts responses to face detection: A deep learning approach. In *Proceedings of the IEEE international conference on computer vision*, pages 3676–3684.
- Yang, X., Yang, X., Liu, M.-Y., Xiao, F., Davis, L. S., and Kautz, J. (2019). Step: Spatio-temporal progressive learning for video action detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 264–272.
- Zhang, S., Benenson, R., and Schiele, B. (2017). Citypersons: A diverse dataset for pedestrian detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3213–3221.
- Zhou, C. and Yuan, J. (2018). Bi-box regression for pedestrian detection and occlusion estimation. In *Proceedings of the European Conference on Computer Vision*.
- Zhu, A., Gao, R., and Uchida, S. (2016). Could scene context be beneficial for scene text detection? *Pattern Recognition*, 58:204–215.
- Zhu, C., Chen, F., Ahmed, U., Shen, Z., and Savvides, M. (2021). Semantic relation reasoning for shot-stable few-shot object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8782–8791.