# Context in Computer Vision: A Taxonomy, Multi-stage Integration, and a General Framework

by

Xuan Wang

A dissertation submitted to the Graduate Faculty in Computer Science in partial fulfillment of the requirements for the degree of Doctor of Philosophy, The City University of New York

2024

This manuscript has been read and accepted by the Graduate Faculty in Business in satisfaction of the dissertation requirement for the degree of Doctor of Philosophy.

**Professor Zhigang Zhu**

Date

Chair of Examining Committee

**Professor Mikael Vejdemo-Johansson**

Date

Executive Officer

**Professor Hao Tang**

**Professor Jie Gong**

**Professor Jie Wei**

**Dr. William H. Seiple**

Supervisory Committee

The City University of New York

Abstract

# Context in Computer Vision: A Taxonomy, Multi-stage Integration, and a General Framework

by

Xuan Wang

Adviser: Professor Zhigang Zhu

Contextual information has been widely used in many computer vision tasks, such as object detection, video action detection, image classification, etc. Recognizing a single object or action out of context could be sometimes very challenging, and context information may help improve the understanding of a scene or an event greatly. However, existing approaches design specific contextual information mechanisms for different detection tasks.

In this research, we first present a comprehensive survey of context understanding in computer vision, with a taxonomy to describe context in different types and levels. Then we proposed MultiCLU, a new multi-stage context learning and utilization framework, which is applied to storefront accessibility detection and evaluation. The MultiCLU has four stages: Context in Labeling (CIL), Context in Training (CIT), Context in Detection (CID) and Context in Evaluation (CIE). Our experiment results show that the proposed framework can achieve significantly better performance than the baseline detector. As the fourth stage, we further design a new evaluation criteria for storefront accessibility dataset, which could provide a new way to think the

evaluation standard in real world applications. For better data collecting and model refinement, we also utilize the MultiCLU storefront detection engine in a smart DoorFront platform for collecting new data and refining the deep learning models.

Furthermore, we generalize our MultiCLU into the GMC framework, a general framework for multi-stage context learning and utilization, which can be applied to various current deep learning models for different visual detection tasks. The GMC framework incorporates three major components (corresponding to the first three stages of the MultiCLU for storefront): local context representation, semantic context fusion, and spatial context reasoning. All three components can be easily added and removed from a standard object detector, which is demonstrated in a number of object recognition tasks including the storefront accessibility detection and the City Pedestrian detection tasks. The GMC framework is further extended to semantic segmentation tasks such as panoptic segmentation, which turns out to be both straightforward and effective. The outcomes of this research seek to provide a generalized approach on streamlining context learning in real world applications at various stages of the processing more flexibly and adapting to different tasks more efficiently.

# Acknowledgements

I would like to thank my advisor Professor Zhigang Zhu at City College of New York and the CUNY Graduate Center, who has guided me through the research with continuous support throughout the research. Professor Zhu provides me with critical feedback and keeps me on the track throughout my PhD research. His patience and support helps me overcome many difficult situations during research. I also want to thank Professor Hao Tang (Borough of Manhattan Community College and the CUNY Graduate Center) for the research guidance and support whenever it is needed, especially for data collection and technical solutions.

Many thanks to other committee members, including Professor Jie Gong (Rutgers), Professor Jie Wei (CCNY & GC) and Dr. William Seiple (Lighthouse Guild), who have been offering valuable feedback and time. Thanks to my lab mate Mr. Jiawei Liu who leads the development of the DoorFront application and collaborated with my research to make the app more smart. Thanks to other members of City College Visual Computing Laboratory. We have spent a lot of time together and helped each other. Thanks also to my friends who have helped me and encouraged me through these tough years. Their support helps me overcome the obstacles and stay focused on my graduate study.

I would like to thank the funding agencies that supported this research. These include the National Science Foundation (NSF) via the Smart & Connected Community Program (Award # 1737533), the Partnerships for Innovation Program (Award #1827505) and the CISE-MSI Program (Award

#2131186); the US Air Force Office for Scientific Research (AFOSR) via Dynamic Data Driven Applications Systems (Award #FA9550-21-1-0082), and the Office of the Director of National Intelligence (ODNI) via the Intelligence Community Center for Academic Excellence (IC CAE) at Rutgers University (Awards #HHM402-19-1-0003 and #HHM402-18-1-0007).

Finally, I would like to thank my family, especially my parents and my wife. It wouldn't have been possible to reach this far without their love and support.

# Contents

# List of Figures

# List of Tables

# Chapter 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 an object detection 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. Co-occurrence of the objects can influence the presence of a target object or event. Spatial relation between objects (e.g., painting is on the wall) provides cues to the location of the target. Semantic context from the scene potentially indicates how likely of an object or event to be found in certain scenes but not others. Temporal information such as nearby frames, previous clips can help to predict what will happen in the future. Non-visual information in the meta-data of image collection (such as dates, environments, locations) can also be used as context information - be it spatial, semantic or temporal. Understanding how context can be applied in various ways would be helpful to design effective context-based vision methods. In this research, we first present a comprehensive survey of context understanding in computer vision, with a taxonomy to describe context in different types (spatial, temporal and others) and levels (prior knowledge, global, local).

Many context based approaches use deep learning methods. Different kinds of network architectures have been used as backbones in context based integration. Various convolutional network architectures have been proposed to train the deep convolutional neural networks (ConvNets). Among reviewed literatures, ResNet and VGGNet are mostly used architectures in context based approaches. Many researches are either use the existing ResNet and VGGNet in employing context information, or use a modified version to better incorporate with context related to the tasks. Other architecture like Graph Convolutional Network (GCN), is used for modelling the spatial relation between target and others, and semantic relation between different object categories because of its unique graph structure. Current labeling approaches in object detection tasks heavily rely on human labelers to create labels on their datasets. To obtain the consistency of the labels, there are predefined description of the target classes and instructions on how to draw labels on images. Usually tight bounding boxes are fit to the target objects. However, to our best knowledge, there is no previous work, if any, that guide the context learning through labeling, training and post processing. In this research, we propose MulitCLU, a framework for multi-stage context learning and utilization (MultiCLU) with individual component that apply to data labeling, model training and post processing, which can be applied individually and in combination.

There are various urban image datasets for different computer vision tasks. Cityscapes [31] is a large-scale street level dataset that is mainly used for semantic urban scene understanding tasks. The Street View Text Dataset, known as SVT [129], is another open source outdoor street level imagery for text detection and recognition of business signage and business names. However, both datasets don't include annotations for storefront accessibility features. For providing accessibility detection features in a complicated street-level environment, we identify three categories of objects in helping people who are blind or have low vision (BLV) to identify the storefront accessibility: 1) doors (for store entrances), 2) Knobs (for accessing the entrances) and 3) stairs (for leading to the entrances). In this research, we collected our own storefront accessibility image (SAI) dataset for detection and evaluation in this work. We apply the MultiCLU deep learning model to detect storefront accessibility objects, by applying specific relations between storefront accessibility objects.

With the MultiCLU storefront detection engine, We further introduce a Smart Doorfront platform, an AI-enabled storefront accessibility annotation platform, by integrating our MultiCLU engine with crowdsourcing image labeling. Since our specially designed storefront image detection model MultiCLU is built upon the state-of-the-art object detector and uses of context information among storefront accessibility objects, our Smart DoorFront platform can perform pre-labeling once a human labeler captures a new Google Street View images, which automates the labeling process and reduce the time for annotation by human labelers. The new data collected in turn can be used to improve the performance of the MultiCLU deep learning engine for storefront accessibility detection.

We further generalize the specially designed MultiCLU framework into a more general design, called GMC, a general framework for multi-stage context learning and utilization with various base detector architectures and for different visual detection tasks.. The GMC framework has also been applied to both the SAI task and the City Pedestrian detection task, with minimal modification in coding. We provide user-defined parameters in configuration for all the context components. The GMC framework is further extended to semantic segmentation tasks such as panoptic segmentation, which turns out to be both straightforward and effective. Our framework not only benefits the detection of small-scale pedestrians and occluded pedestrians by using contextual labeling, all the components can also benefit each other and further improve the performance. The outcomes of this research seek to provide a generalized approach on guiding context learning in real world applications so adapting to different tasks would be more efficient.

In the following, we will first discuss the differences of context understanding between human vision and computer vision, and explain why context reasoning is still challenging but critical for computer vision. Examples are presented to explain why context is important for both human and machine. Then we highlight the contribution of this research, and lay out the organization of the thesis proposal.

## 1.1 Roles of Context

Humans use visual context effortlessly to perceive the real world. What we see is not only based on the signals that our eyes send to our brain, but is influenced strongly by the context. The visual stimulus is presented in, on our previous knowledge, and expectations. Intrinsic features (shapes, colors, texture, etc.) of an object against a background of the scene in the retinal images of our eyes provides enough information to determine what the object is. Human can also easily recognize the object under normal conditions. However, when an object appears in isolation from its surrounding scene, recognizing the object becomes unreliable. Fig. 1.1 shows an example of an object in isolation and the same object in context. When the keyboard is taking out from the office environment, human can barely recognize it. But within the office scene, we can identify the object in front of the monitor is a keyboard, even the surrounding area is blurry. Context provides critical information to help us visually find and recognize objects faster and more accurately.

Context encapsulates rich information not only on how natural scenes and objects are related to each other, but also the relative positions of objects with respect to a scene or co-occurrence of objects within a scene. Besides the visual form of context, no-visual information can also provide important cues. For example, without looking at an image, and if we know that there is a ship in the image, we can easily guess there is a river or sea in the image. Human can even draw a picture with only description of an object or an event. Human can benefit from either the visual information between

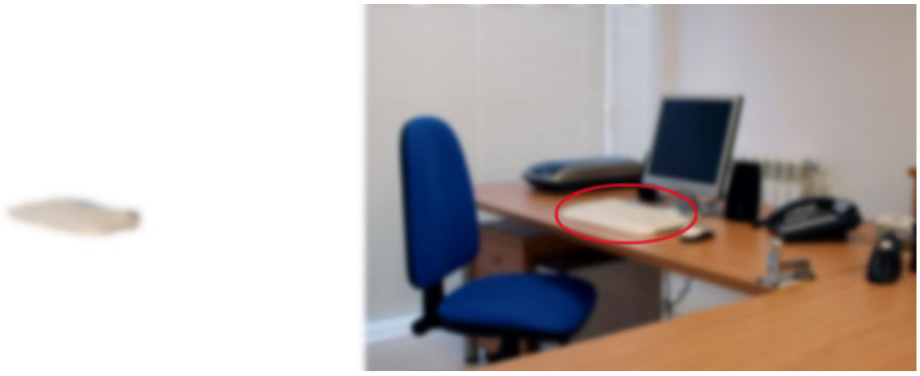


Figure 1.1: An example from [86]. Left: An isolated object. Right: The object in its relevant scene.

objects and scenes, or the relations between semantically related objects.

Nevertheless, contextual information in natural scenes provides critical information to help us visually find and recognize objects faster and more accurately. An object that cannot be recognized in isolation can be identified when it appears in relevant context scene. Context besides an object itself can serve as a supplement available for the object. Human can also infer information about the scene that will be useful for interpreting other parts of the scene. We can easily build a hierarchical relations between objects using not only visual context but also no-visual context.

So far it seems that humans can always perform better than machines. One potential reason for this performance gap is that humans and machines have qualitatively different learning mechanism. Machines are typically learned on images containing objects in a certain context with limited amount of data, whereas humans view objects in different context in real world daily. Another reason is that computer vision is trying to mimic human vision, but with the help of our brain, human vision is more advanced.

We not only can learn context and object separately, but also easily build up connections and relations between objects and their context. In contrast, computer vision is still challenging to model the relation between objects and context. When the object has a weak correlation with its surrounding context, the context can be difficult to learn in the presence of more informative object features. On the other hand, if the object has a strong correlation with its context (e.g., living room usually contains a TV), the object can be learned effectively along with the context. These variations make machines hard to learn context systematically and independently of the object. Context also has different presentations between human vision and computer vision.

Although we can use context effortlessly with our human vision system, context reasoning about objects and relations is still challenging and critical to computer vision. Fig. 1.2 shows that a machine algorithm can recognize the object clearly: a rider (black box), a bike (Orange box) and a helmet (Yellow box). With only capturing these kind information, parallel relations (a man rides a bike) and hierarchical relations (helmet affiliated to head) are missing.

Given an image or a video of the real world, the final goal of a computer vision system is to determine what visual elements and structures are presented, how these elements are related to each other, and to have a complete understanding of what is happening in the visual input. Visual understanding is difficult to define and evaluate, therefore researchers have concentrated on solving more focused, specialized, low-level problems like object detection or image classification. Object recognition does not occur as an isolated pro-

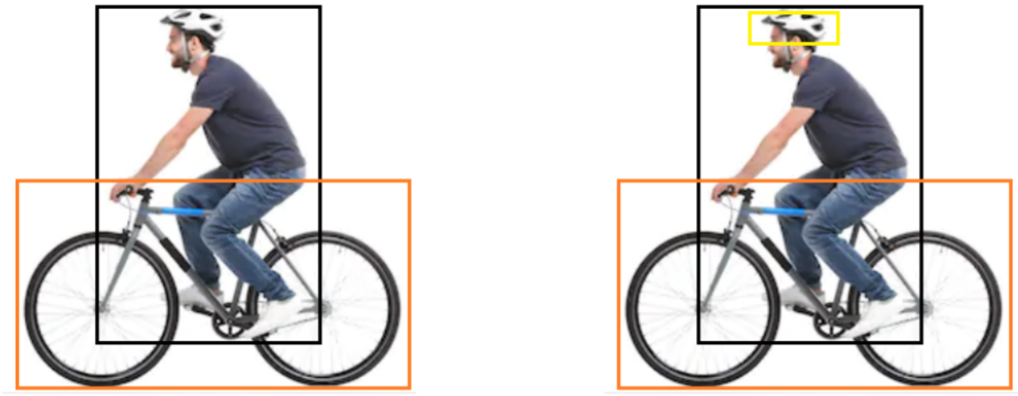


Figure 1.2: Example of information captured by machine. Machine can easily capture single object, but lack of relations between them. Images are from internet source.

cess since, it can be influenced by the presence of other objects as well as by the overall context of the scene. Global context provides a rich source of information that can help to improve the performance of the recognition task.

Human and machine treat context differently. Our brain not only processes the signal that our eyes send, but is also influenced by the rich context from the seeing. Human can localize and recognize the objects or events even in considerable amount of occlusions, illumination changes and various viewpoints, etc., which are still big challenges for computer vision. This gap can be caused by the differences of training and testing data. Machine are trained on images or video of certain objects or events, with certain context, but the models might be used for images or videos in a totally different context. Whereas human vision systems are very experienced with large variances of scenes (with or without objects or events, environment changes, appearance changes, etc.). Nevertheless, machine vision models and algorithms have been explored in decades in understanding context in a systematic way hopefully like humans, in various forms in computer vision tasks.

## 1.2 Contributions of the Thesis

We propose a general context learning and reasoning framework, which could guide the deep learning framework through labeling, training and post processing. Here are the main contributions of the research:

- We present a comprehensive survey of context understanding in computer vision, with a taxonomy to describe context in different types (spatial, temporal and others) and levels (prior knowledge, global, local) (Chapter 2). Furthermore, we review various context based integration in two categories: image-based context integration and video-based context integration.

- A multi-stage context learning and utilization (MultiCLU) framework is designed specifically for storefront accessibility detection (Chapter 3), by employing the specific relationship between storefront accessibility objects (Door, Knob, Stair), in the stages of labeling, training and detection. We further introduce a new evaluation metric in the evaluation stage for the knob category in our task, which could provide a new way to think the evaluation standard in real world applications.

- An AI-enabled Storefront Accessibility Annotation and Localization Platform (Chapter 4) is developed by applying our special MultiCLU framework into our previous developed Doorfront platform [77].With the AI-based pre-labeling, we also introduce an online machine learning mechanism to iteratively train the MultiCLU model, by using newly labeled storefront accessibility objects. By integrating our MultiCLU framework with crowd-sourcing data labeling, our new platform not only significantly improves the efficiency of storefront accessibility data collection, but optimizes user experience.

- The MulitCLU framework for mutli-stage context learning and utilization is generalized to the GMC framework, a general framework for multi-stage context learning, with various base detection architectures, and for various visual detection tasks (Chapter 5). The GMC framework consists of three contextual components: Local Context Representation (LCR), Semantic Context Fusion (SCF) and Spatial Context Reasoning (SCR). These contextual components take advantages of different contextual information, and guide the deep learning detector through labeling, training and post processing. Each component can be applied individually and in combination. The framework is further extended to work for other visual tasks such as semantic segmentation. This shows that the framework can be integrated with various base architectures, and applied to different visual tasks without much changes in coding (Chapter 5, Chapter 6, Chapter 7).

## 1.3 Organization of the Thesis

The thesis is organized as follows. Chapter 2 provides a taxonomy of context in terms of major context types (Spatial, temporal and other), context levels

(Prior knowledge, global, local) and context integration approaches in both image-based tasks and video-based tasks. Chapter 3 describes the proposed MultiCLU framework for storefront accessibility detection task, including data collection, the designs of the three stages of contextual components and a new evaluation criteria specifically for storefront evaluation in detail. We further integrate our special MultiCLU framework into our previous developed Doorfront platform for enabling data labeling more efficiently and for refining the detector for better performance; this will be described in Chapter 4. Chapter 5 discuss how we generalize the specifically designed MultiCLU into GMC, a general multi-stage context learning and utilization framework with three general components: local contextual labeling, contextual graph generation and spatial contextual reasoning in detail. In this chapter, we also shows how we can use the general framework for the storefront detection with the same performance as MultiCLU, and use more advanced base detector for improved performance. Then, we applied our general framework for CityPersons pedestrian detection and show the improved performance of the detection (Chapter 6). Finally, we provide an important extension of the GMC framework for panoptic segmentation tasks (Chapter 7) and conclude the work with potential future directions (Chapter 8).

# Chapter 2

# A Taxonomy of Context in Computer Vision

In this chapter, we present a taxonomy to describe context. We review context based approaches and discuss context in three major types: spatial context, temporal context and other context. We then discuss context in different levels: prior knowledge level, global context level, and local context level, and review how these context has been employed in context based approaches. Furthermore, we review various context based integration in two categories: image-based context integration and video-based context integration. A more detailed analysis on the taxonomy of context, as well as the performance comparison of context integration, can be found in our survey paper published in the Elsevier journal Computer Vision and Image Understanding (CVIU) [138].

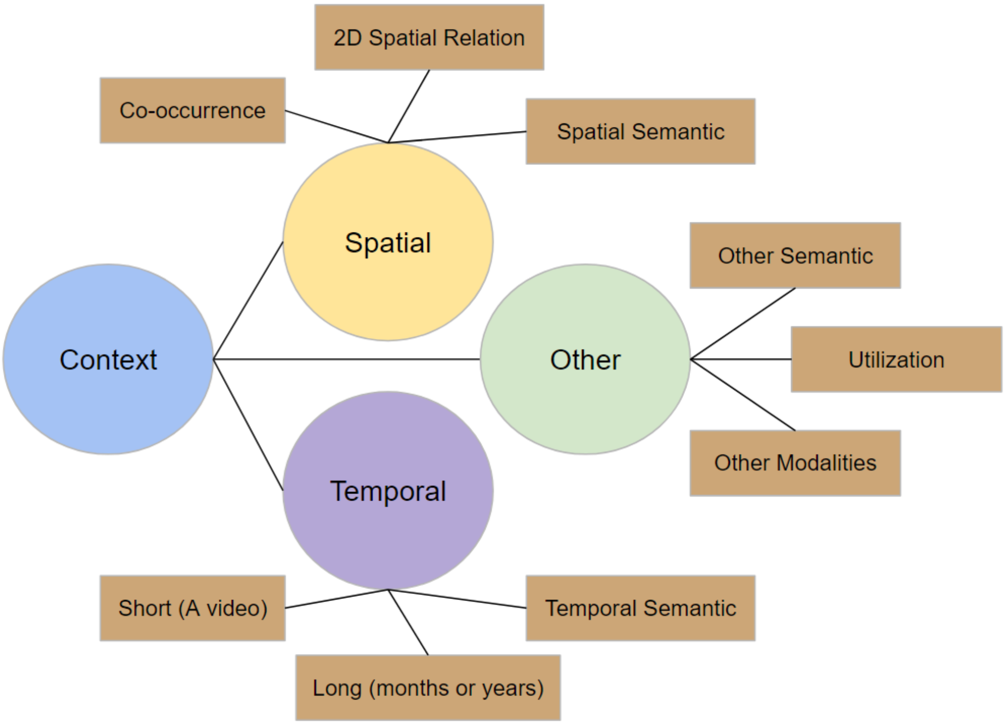


Figure 2.1: Three major context types, their relations and their sub-types.

## 2.1 Major Types of Context

Context information can be from the appearance of the object or the event in consideration, such as shape, color, texture, etc., and can be also from any other information or data not directly related to the appearance of the object or the event, such as environment (inside or outside), location (classroom, restaurant, gym, etc.) and description (drinking coffee, riding bike, etc.), etc. We separate context into three major types: spatial context, temporal context and other context, as shown in Fig. 2.1. Spatial context represents the spatial relation between objects and events, such as co-occurrence, 2D spatial relations and spatial semantic constraints. Temporal context refers to temporally proximal information, either from nearby frames of a video in a short period, or similar scenes captured in months or years, or temporal semantic constraints. Semantic context can indicates an object or an event should be found in some scenes but not others. Semantic context can easily used to describe spatial context or temporal context in a language model and we categorize them under spatial and temporal context types. Other context includes other semantic context that are neither spatial or temporal, and context clues from other modalities such as audio, thermal and weather, etc, and context information stemmed from utilization and purpose. Three types of context are sometimes used in combination in various computer vision tasks.

### 2.1.1 Spatial Context

Spatial context can be defined as the likelihood of finding an object in some positions and not others with respect to other objects in the scene. A car is on the road, not in the sea. If a piece of glass is not on the wall, then it is not a window. An object in an image is supposed to fit into reasonable relationships with other objects in the image. The spatial context can provide information about these spatial knowledge. One of the simplest way to introduce relationships between objects in a scene is co-occurrence. Spatial knowledge such as “A bird is flying in the sky”, can be translated directly into spatial relations between objects in the scene. As common sense, certain objects (e.g. Chopping boards, TVs) should occur more frequently in certain places (e.g., kitchens and living rooms, respectively). Spatial context usually refers to:

1. Environment of the object at present time and location.
2. Related context around target object.
3. Path/direction to destination.
4. Events happen around the object.

How can we arrange the relationship between these context and the target objects? There are two major spatial context representations effectively used in context modeling and contextual reasoning: co-occurrence and 2-D spatial representation. In addition to these two representations, there is also spatial semantic context where semantic constraints can restrict spatial relations between objects.

**Co-occurrence representations.** Co-occurrence is one of the simplest way to introduce relationships between objects in a visual scene. Contextual interactions such as "cars appear on roads" can be translated directly in contextual relations between object labels. It is straightforward to build context matrices to count co-occurrence of labels given a dataset where many objects are labeled. Rabinovich et al. [101] devised interaction potentials for Condition Random Field (CRF) in order to measure contextual agreement between detected objects. It is interesting to notice that the terms "semantic context" and "co-occurence" are sometimes used interchangeably. The statistical model proposed by Carbonetto, Freitas and Barnard [13] also learns co-occurrence between concepts (e.g., image caption words). However in their model, Markov Random Field (MRF) interaction potentials are estimated only between neighboring image segments (e.g., object blobs). Co-occurrence can also be modeled between object parts. Fink and Perona [42] detect faces by using both the individual detections of facial parts (left eye, right eye, mouth, nose, entire face) and their spatial arrangements.

**2D relation representations.** 2D relations can also be used in spatial context modeling can contextual reasoning. The relative directional positions ("above", "below") are frequently used and judged discriminative enough to detect object in conventional dataset like PASCAL [40]. Heitz and Koller [51] also cite some human knowledge e.g., "cars park 20 feet away from buildings" that highlight the limitation of 2D spatial reasoning with a single image in order to describe distance relation since a 3D geometric context would be required to capture that relation. Many recent works [115, 149, 163, 156] uses semantic context to describe the spatial relation between objects and events.

**Semantic spatial context.** Other than co-occurrence and 2D spatial relations, Semantic context can describe these spatial relations in a more general and effective way. Spatial semantic context can be obtained from strongly labeled training data. Several works [115, 149, 163, 156] employed semantic context for scene graph generation tasks. They state that even though a object detector can detect all the objects appears in a scene, it still cannot understand the semantic relationship. Rich semantic context can indicate the specific spatial relations between objects, and result in a deeper understanding of visual scene. Spatial semantic context in non-visual forms can also help in predicting the presence of an object. Rabinovich's work [101] shows that mis-labeled "Lemon" is refined to correct "Tennis" by enforcing semantic contextual constraints (in a scene of tennis match).

### 2.1.2 Temporal Context

In common sense, temporal context can be interpreted as the information in a video, such as nearby frames, previous clips or video captured recently. Many works use the temporal context within a video to improve the performance. However, for some computer vision tasks, such as species classification[85], animal movement[8], temporal context from nearby frames or a recent video is not sufficient, a longer temporal context (over months or years) will be needed to help with these tasks. The longer temporal context sources can provide useful information, such as movement pattern of the species across different time periods, which will better indicate the presence of the objects in the scene. We first categorize temporal context in two categories: temporal context in videos and temporal context across months. Many works are focused on video by using nearby frames as temporal context. There are also temporal semantic context where temporal information is provided mostly in non-visual forms to serve as a temporal cue for the task.

**Short-term temporal context.** Short-term temporal context refers to temporally proximal information, such as nearby frames of a video, images captured right before/after the given image, or video data from similar scenes and time of capture [36]. Temporal cues has been employed widely in video related tasks [141, 132, 133, 151, 156]. Since nearby frames of a video may have a better feature representation of the target, a recent work [141] investigates how to utilize local temporal context to enhance the representations of heavily occluded pedestrians.

**Long-term temporal context.** Long-term temporal context is used as the neighbor frames or a large temporal scale within a video. From a broader perspective of temporal information, temporal context is not limited to nearby frames in a short period of time in video-based tasks, it can also be leveraged in a long period of time such as months or years, which can provide long-term temporal consistency for video-based tasks. These kind of temporal information are used in species recognition tasks [85, 8]. Beery et al. [8] propose Context R-CNN, which leverages temporal context (a month's worth of images from the same camera) for improving object detection regardless of frame rate or sampling irregularity. Aodha et al. [85] introduce a framework which incorporates with the long term temporal context (months to years) to help the model successfully distinguish the species with similar appearance.

**Semantic temporal context.** Semantic context can also be temporal information. These contexts are usually provided by the dataset or embedded in the metadata. The iNatualist dataset [124] consists not only the species' images, but also comes along with descriptions, locations, time and dates, and observer identifications, which are embedded in the metadata. The time and date information can be served as a temporal prior to help identify the species in the image, and also track the movement of the species. Semantic context can also serve as temporal cue to help find activities in video task. Yuan et al. [162] uses semantic context to determine the temporal boundaries in temporal grounding in videos task.

### 2.1.3 Other Context

Here we categorize other contexts into other semantic context, context in other relations and context in other modalities. semantic context can be neither spatial context or temporal context. Other semantic context only describes the dependency of the objects without any spatial information or temporal information. There are also contexts in other relations and other modalities, which can also provide critical cues in computer vision tasks.

**Other semantic context.** As mentioned in spatial context and temporal context, semantic context usually provides constraints of the presence of the objects in a scene. For example, if we know the event a basketball match, we are expecting to see certain objects present in a certain scene: Basketballs and basketball stands in a basketball court. We are expecting snow in winter. These kind of semantic contexts also indicates the spatial information and temporal information. On the other hand, there are also other semantic contexts that are neither spatial or temporal. These kind of semantic context only indicate the presence of the objects, without any spatial information or temporal information. A work [24] in a multi-label image recognition task uses the label dependency to model the semantic relations. Another work [173] in an object detection task uses semantic space projection to model the semantic relation to aid the learning of the visual information of the objects.

**Context in utilization.** Context in other relations such as functionalities, purposes or intention can indicate the occurrence of certain actions or objects. There are also contextual information from (or for) other modalities, such as audio, text, thermal and weather, etc, which can be helpful in computer vision tasks. A recent work [66] introduce the problem of functional correspondence, which is aimed to find the set of correspondence between two objects for a given task. Any two objects that can be used to perform an action are then used to establish a correspondence relationship. Human usually direct their attention and move their body based on their intention. The intention is also informative of the human-object interactions. Another work [148] uses human intention for detecting human-object interactions (HOIs) in social scene images.

**Context in other modalities.** There are also context in other modalities, which can be used in computer vision tasks. When we hear a dog barking, we can estimate how far and which direction the dog is located. Audio has been used in event localization task [120] and floorplan reconstruction task [99]. Besides audio, thermal can also be informative, such as estimating animal populations. A work [106] employs thermal as the context along with imagery to estimate seals in eastern Canada. The proposed methods improves upon shortcomings of computer vision by effectively recognizing seals in aggregations while keeping minimum model setup time.The proposed methods improves upon shortcomings of computer vision by effectively recognizing seals in aggregations while keeping minimum model setup time.

### 2.1.4 Summary of Context Types

In this section, we mainly review three major types of context: spatial context, temporal context and other context. Spatial context can be defined

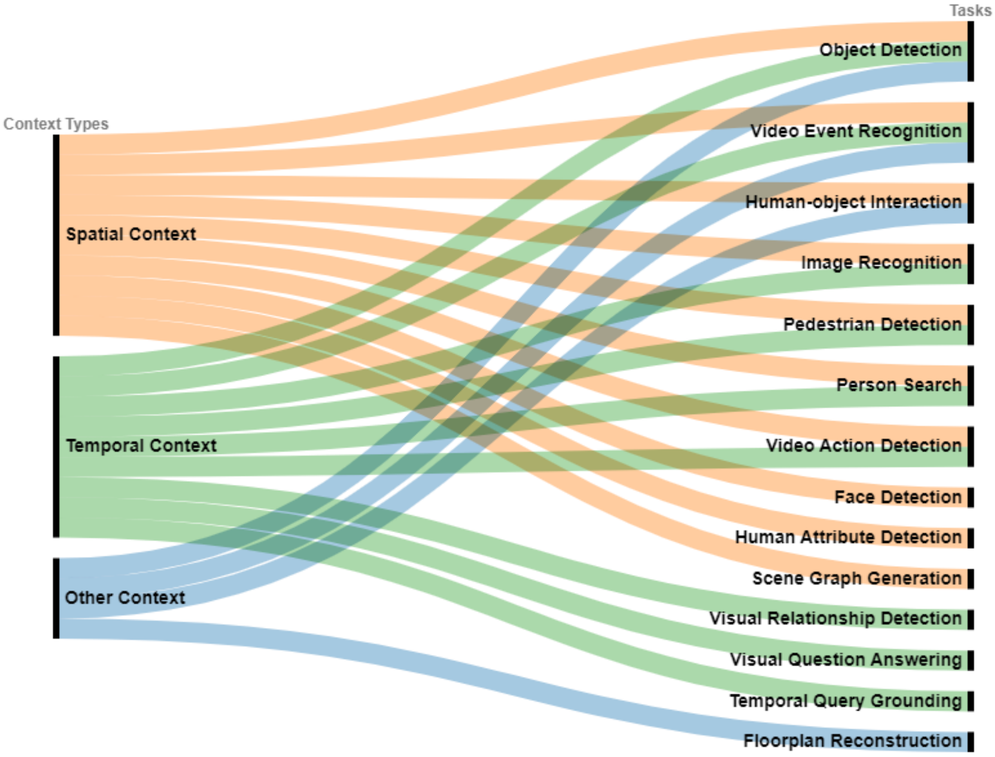


Figure 2.2: Summary of the three major context types and what tasks they are employed in.

as the likelihood of finding an object in some positions and not others with respect to other objects in the scene. We further split the spatial context representations into three categories: co-occurrence, 2D spatial relations, and semantic relations. Temporal context refers to temporally related information and it can be separated into a short term, a long term, or a semantic relation over time. In general, semantic context corresponds to the likelihood of an object to be found in some scenes but not others. Semantic context can be both spatial context and temporal context across time, and can also be neither of them. Context in utilization can reveal the functionalities or purposes of the objects or the actions. Context in other modalities, such as audio and thermal can also be informative for some computer vision tasks. All these types of contexts are employed in various combinations in existing context based approaches. Fig. 2.2 shows the major context types and the related tasks when employing context information.

## 2.2 Levels of Context

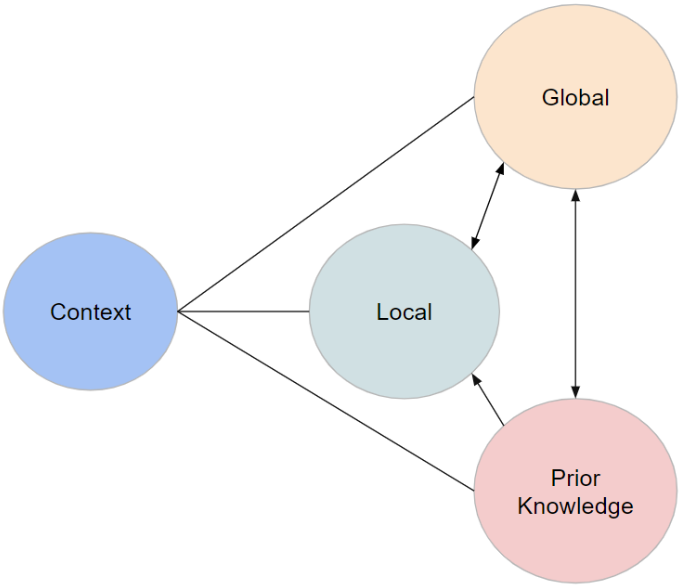


Figure 2.3: Three different context levels and their relations.

Context can also be represented in different levels. We separate context into three levels: prior knowledge level, global level and local level. Prior knowledge refers to the knowledge obtained *before* seeing the scenes or events, such as location, time and weather etc., serves as *a prior* knowledge for computer vision tasks. Global context exploits the visual scene as global information, which could provide context such as spatial layout and semantic relations between objects. Local context contains the intrinsic context (of the object itself) and the extrinsic context (surrounding regions or objects of the target). As shown in Fig. 2.3, global level context can include local level context from the object, which could be further extracted from the local regions. Prior knowledge can serve as *a prior* for a global scene, and global context can also be in the form prior knowledge to indicate the occurrence of the object or the event. Prior knowledge can also provide important information for local context, which can serve as a cue for computer vision tasks such as object recognition and object detection. In this section, we provide details for each context level and how they are employed in different computer vision tasks.

### 2.2.1 Prior Knowledge Level

Context at the prior knowledge level refers to the knowledge obtained before seeing the scenes or events. It reflects the environment such as location and time, that can serve as prior to predict whether certain events would occur or certain object would be detected in the visual scene. For example, if we know there are a hotel building and a bus stop in the scene before we see the scene, we can easily guess the text appeared on a hotel building is probably different from the text appeared on a bus stop advertisement. These context information are treated as high level context information. Furthermore, context between neighboring images can also provide high level information. Both context will provide prior information for the inference of the task. Prior knowledge may not be directly extracted in the image or video in consideration. It may come from previous event for temporal support or metadata, which will serve as prior information to analyze the current scene.

A series of works [133, 132, 131] on video event recognition employs prior level context from two aspects: the context from current scene, and temporal support from previous event. Prior context from current scene reflects the environment such as locations (e.g. a parking lot, a shop entrance) and times (e.g. at noon, in dark) that can serve as prior to dictate whether certain events would occur. Prior context from previous event can provide temporal support for the prediction of the current event. These prior context provides critical cues for event recognition task. The other work [85] uses geographic location information extracted from metadata as spatial information, and it also serves as prior knowledge for the species recognition. These kind of prior knowledge can also provide useful context for recognizing the species appears in certain areas.

### 2.2.2 Global Context Level

Global context exploits scene configuration (image as a whole) as an extra source of global information across categories. The structure of a scene image can be estimated by the mean of global image features, providing a statistical summary of the spatial layout properties. Rabinovich's study [100] shows that by incorporating the statistics of the background, context becomes a global feature of the object category. For example, refrigerators usually appear in a kitchen, thus the usual background of refrigerators is similar. Having learned such a global feature of an object category, one can infer a potential object label: if the background resembles a living room, then the patch of interest may be a TV. The background or scene provides a likelihood of finding an object in the scene (e.g. it is unlikely to find a car in living room). It can also indicate the relative positions at which an object might appear (e.g. car on the road, pedestrians on walkways, etc.)

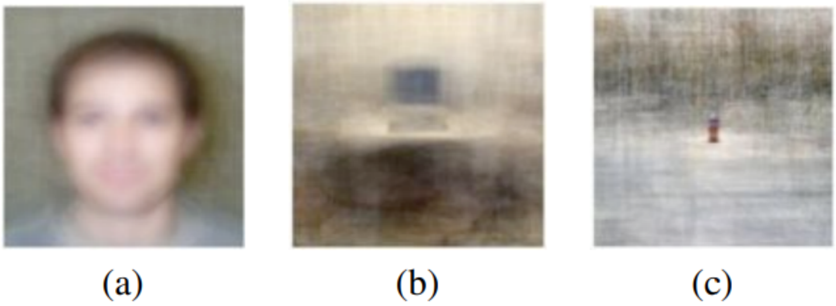


Figure 2.4: The structure of objects and their backgrounds. The figure was presented in [121]. Each image has been created by averaging hundreds of images containing a particular object in the center (a face, a keyboard and a fire hydrant) at a fixed scale and pose. The averages can reveal the regularities existing in the color/brightness patterns across all the images. However, this regularities is only visible for the keyboard in (b).

The work by Oliva and Torralba [121] discusses how context influence on object recognition task. Global context and the objects within it can influence each other. However, some objects (faces, cars, persons, etc.) may have various background scenes based on the locations of their appearance. E.g., a car can be on the road, or in the parking lot. However, some objects (faces, cars, persons, etc.) may have various background scenes based on the locations of their appearance. E.g., a car can be on the road, or in the parking lot. A person can appear indoor or outdoor, day or night. Under these circumstances, the background or the scene cannot indicate the object correctly.

The demonstration of this limitation from [121] is shown in Fig. 2.4(a)(c). Another limitation of global context is that it could be misleading if an object present in irrelevant scenes. Choi et al. [29] present a context model for out-of-context detection, where the object is unusual for a given scene in a image. Another recent work [10] introduce a out-of-context dataset and propose a context reasoning model for out-of-context object recognition.

### 2.2.3 Local Context Level

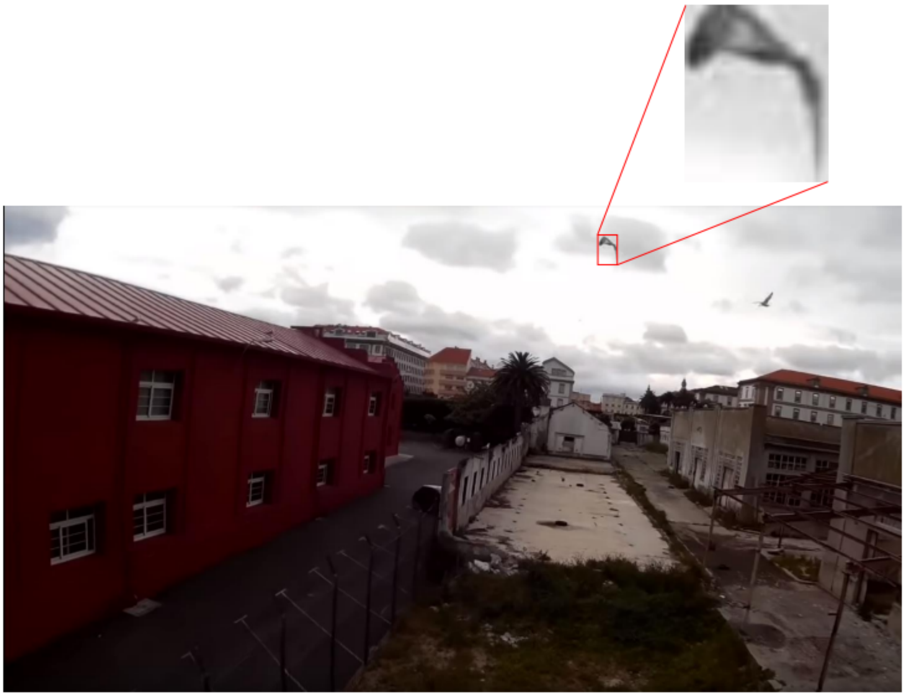


Figure 2.5: Context is necessary to recognize small objects such as birds in this picture. The figure was presented in [74].

Local context level context indicates the context from objects itself and surrounding local regions, such as color, shape, contrast with background, aspect ratio and other objects etc. Local context features can capture different local relations such as pixel, region and object interactions. As aforementioned in global context level, context could misleading if the object present in irrelevant scenes. Therefore, rather than measuring global level features, local context can better impose on potential object presence in the image. For smaller object like a flying bird, the surrounding area can provide important information in the context of sky (example in Fig. 2.5).

Local context can also indicate the location of the objects. This information can be captured using spatial context. Spatial context between objects within surrounding area can help because: (1) most objects are supported by other objects, e.g. car is on the road, pedestrians are supported by sidewalk or ground; (2) objects are not appeared in isolation. Objects that have a common function tend to appear nearby and have a certain spatial relation, e.g. a mouse appears next to a keyboard, a dining chair appears besides the dining table, etc; and (3) The structure of the global scene tend to have a common layout, e.g. a stair should appears under a door, and it should appear at at the lower half of the scene. Sky should appears above buildings, and it should appear at the upper half of the scene. Torralba et al. [121] also shows that the vertical spatial relation indicated by local context is usually more informative than the horizontal spatial relation.

Although global context can help indicate the presence of object and spatial representation between objects, if the number of objects increase in the scene, global context cannot discriminate well between scenes, since many objects may share the same scenes, and scenes may look similar to each other. Local context representation is still object-centered and it requires object recognition as a first step, which is different from global context.

### 2.2.4 Summary of Context Levels

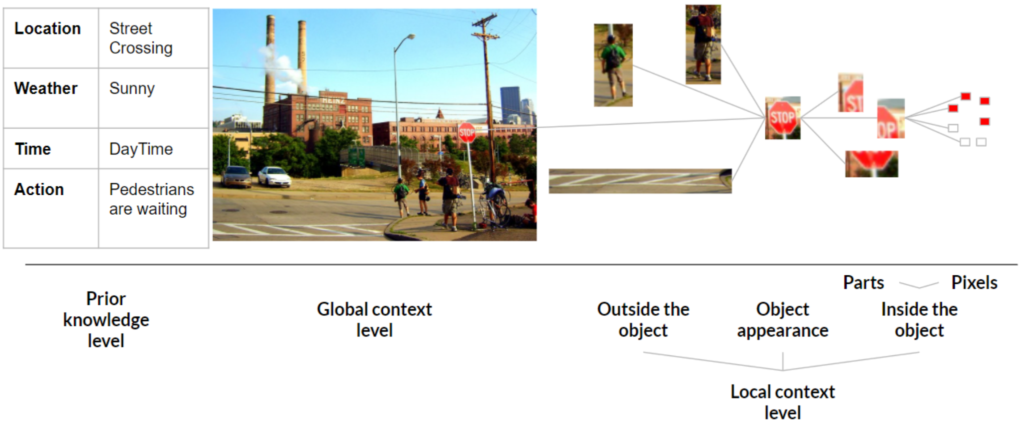


Figure 2.6: Summary of context levels. The image in the middle is from COCO [76] dataset.

Fig. 2.6 illustrates these three levels with a real image example. Prior knowledge level refers to the context information obtained for the whole image globally. It reflects the environment such as location and time, that can serve as prior to predict whether certain events would occur or certain object would be detected. Global context exploits scene configuration (image as a whole) as an extra source of global information across categories. The structure of a scene image can be estimated by the mean of global image features, providing a statistical summary of the spatial layout properties. Local Feature level context indicates the context from objects itself and surrounding local regions, such as color, shape, contrast with background, aspect ratio and other objects etc. All there context levels are employed by various approaches in different computer vision tasks.

## 2.3 Context Integration

In this section, we review how context information has been integrated in various computer vision tasks in two main categories: image-based tasks and video-based tasks. Semantic context and spatial context are heavily used in image-based tasks. Either spatial relation between different objects or different parts within an object is integrated to extract context features. Semantic context, such as location, weather, etc., is served as prior level knowledge. Other semantic context such as object relation description, object appearance, label co-occurrence is served as spatial relation for tasks like image recognition and object detection. A few works use temporal context from long term (months to years) as a historical information for predicting current object appearance. Temporal context is the mainly context sources for video-based tasks. Temporal context not only provide previous clues for current scene, it also carries semantic context and spatial context in both the language form and the visual form. These context can help to solve some challenges in video-based tasks, such as heavy occluded pedestrian detection, video event recognition, temporal query grounding, etc. In this section, we review details for representative context integration tasks in both image-based tasks and video-based tasks.

### 2.3.1 Image-based Context Integration

Spatial context and semantic context are heavily used in image-based context integration, even though some works [85, 165] also use temporal context as a prior to improve the performance. Table 2.1 provides a summary of all the reviewed works in image-based context integration, in terms of tasks, backbone deep NN (DNN) models, employed context types, employed context levels, and mechanisms for using context. We provide details for some representative works in different image-based tasks.

**Face detection.** Yang et al. [153] proposed Faceness-Net for face detection. Faceness-Net uses spatial structure and arrangements of face parts as context cues to detect faces. The Faceness-Net considers using spatial structure and arrangement of face parts as a context cue to detect faces. Each facial parts was scored separately in case of the occlusion and pose variation.

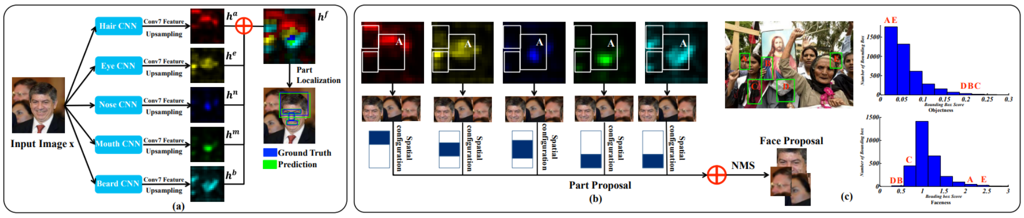


Figure 2.7: The pipeline of Faceness-Net [153]. Faceness-Net uses spatial structure and arrangements of face parts as context cues to detect faces. The figure was presented in [153].

**Human attribute recognition.** Built on the observation that context can unveil more clues to make recognition easier, Li et al. [73] proposed hierarchical context model for human attribute recognition task. Similar to Faceness-Net [153], the hierarchical context model incorporate with both global level context (whole scene) and local level context (human body parts) for final human attribute recognition.

**Image classification.** Several recent works [24, 165, 85] has employed context information as important cue for recognizing the object. Chen et al. [24] uses Graph Convolutional Network to model the co-occurrence rela-

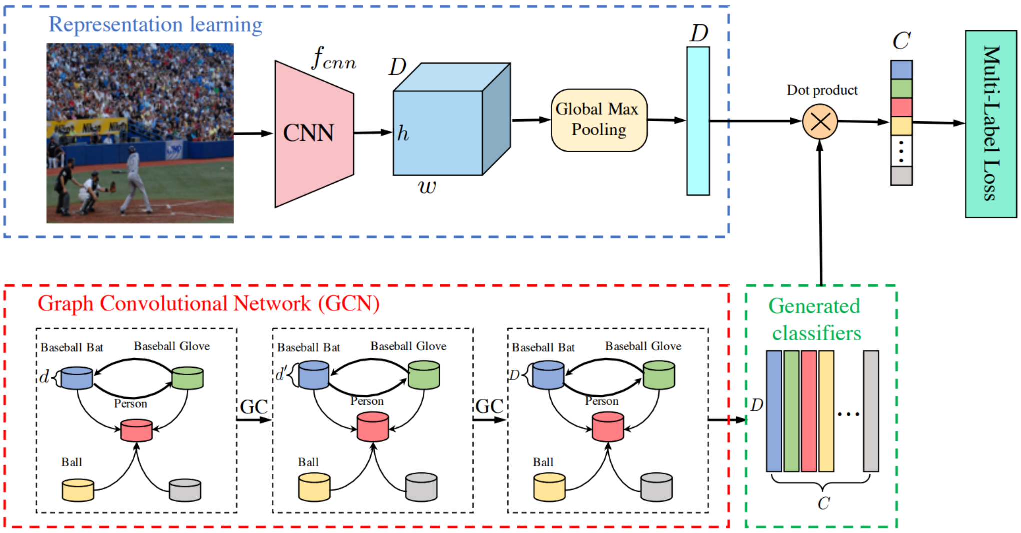


Figure 2.8: The architecture of Multi-label image classification [24]. Graph Convolutional Network is used to model the co-occurrence relation from prior label dependencies. The figure was presented in [24].

Table 2.1: Image-based context integration.

| **Methods** | **Tasks** | **DNN Models** | **Context Types** | **Context Levels** | **Mechanisms** |
|---|---|---|---|---|---|
| Faceness-Net[153] | Face detection | AlexNet | Spatial | Local | Faceness-Net |
| Hierarchical Context Net[73] | Human attribute recognition | VGGNet | Spatial | Global, Local | Hierarchical context net |
| Hierarchical Random Field[158] | Human-object interaction | Custom | Spatial | Local | Graph model |
| ML-GCN[24] | Image recognition | GCN | Spatial, Temporal | Global, Local, Prior knowledge | GCN |
| Spatio-temporal Prior Model[85] | Image recognition | ResNet | Spatial, Temporal | Global, Local, Prior knowledge | Bayesian model |
| CATNet[165] | Image recognition | VGGNet | Spatial, Temporal | Global, Local | Two-stream context net |
| Context Encoder[96] | Image inpainting | AlexNet | Other | Local | Context encoder |
| Co-occurrence Tree Model[29] | Object detection | / | Spatial | Global | Tree-structured model |
| Context Data Augmentation[38] | Object detection | ResNet | Spatial | Local | Context CNN |
| Knowledge-aware Model[41] | Object detection | VGGNet | Semantic | Global, Local, Prior knowledge | Knowledge graphs |
| Internal-External Context Model[68] | Object detection | ResNet | Spatial | Local | Internal-External Network |
| Feature Fusion Attention Model[74] | Object detection | ResNet | Other | Global, Local | Feature fusion SSD |
| Deformable Part-based Model[89] | Object detection | / | Spatial | Global, Local | Markov random field |
| Siamese Context Network[115] | Object detection | Custom | Spatial | Global | Siamese CNN |
| Bayes Probabilistic Model[122] | Object detection | / | Spatial | Global, Local | Bayesian Model |
| Semantic Relation Reasoning Model[173] | Object detection | ResNet | Other | Global | SSD |
| Cascaded Refinement Network[58] | Scene graph generation | GCN | Spatial | Global, Local | GCN |
| Iterative Message Passing[149] | Scene graph generation | VGGNet | Spatial | Global, Local | Conditional random fields |
| Graph R-CNN[152] | Scene graph generation | GCN | Spatial | Global, Local | GCN |
| MOTIFNET[163] | Scene graph generation | ResNet | Spatial | Global | Bayesian model |
| Conditional Random Field (CRF)[90] | Semantic segmentation | / | Other | Global | Conditional random field |
| Context-based SVM[37] | Text detection | Custom | Spatial | Local | SVM |
| Visual-language Re-ranker[105] | Text detection | ResNet/GoogLeNet | Other | Global | Language model |
| PLEX[129] | Text detection | / | Spatial | Local | Trie structure |
| Scene Context-based Model[172] | Text detection | Custom | Other | Global, Local | CNN/SVM |
| Context-dependent Diffusion Network[32] | Visual relationship detection | VGGNet | Spatial | Global | Graph model |
| Dynamic Tree Structure[119] | Visual Q&A | VGGNet | Spatial | Global, Local | Tree-structured model |

tion from prior label dependencies. Since Graph Convolutional Network uses relation descriptor $A$ to propagate information between nodes. The author models the label correlation dependency in the form of conditional probability, and then feed into Graph Convolutional Network. By applying the prior semantic relation using label appearance, the model consistently achieves superior performance over previous competing approaches. The architecture is shown in Fig. 2.8.

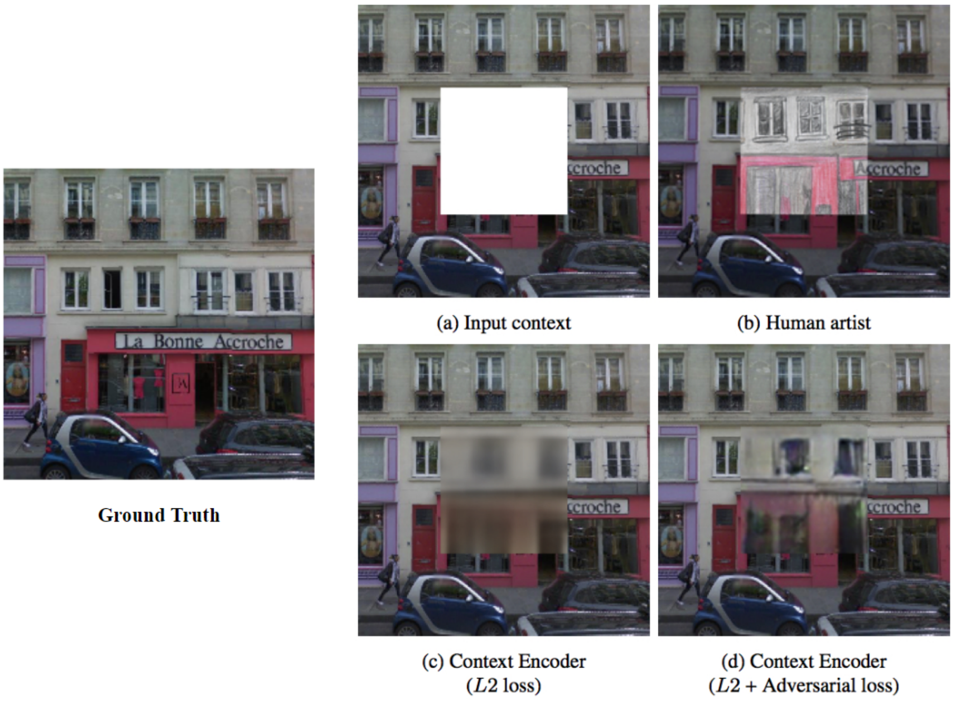


Figure 2.9: Qualitative illustration of the image inpainting task. Given an image with a missing region (a), a human artist has no trouble inpainting it (b). Automatic inpainting using a context encoder is shown in (c) and (d). The mechanism of employing context is to train a Context Encoder to generate the contents of an arbitrary image region conditioned on its surrounding local context. The figure was presented in [96].

**Image inpainting.** Image inpainting is a task of predicting the arbitrary missing region based on the rest of the image. To correctly predict the missing region, the network is required to learn the common knowledge including the color and structure of the common objects. By analogy with auto-encoders, Pathak et al. [96] train a convolutional neural network to generate the contents of an arbitrary image region conditioned on its surrounding context. The context encoder learns a representation that captures not only appearance but also the semantics of visual structures. The overall pipeline is a encoder-decoder architecture. The encoder is a convolutional neural network that predict missing parts of a scene from their surroundings. The decoder then generates pixels of the image using the features learned from encoder. In order to accomplish the task, both encoder and decoder are required to learn the semantic context of images.

**"Out-of-Context" detection.** The context of an image encapsulates rich information about how natural scenes and objects are related to each other. Such contextual information has the potential to enable a coherent understanding of natural scenes and images. However, context models have been evaluated mostly based on the improvement of object recognition performance even though it is only one of many ways to exploit contextual information. Choi et al. [29] present a new scene understanding problem, which is interested in finding scenes and objects that are "out-of-context". Detecting "out-of-context" objects and scenes is challenging because context violations can be detected only if the relationships between objects are carefully and precisely modeled.

**Object recognition.** As shown in Fig. 2.10, the author presented a graph model to combine different context information such as global context, object co-occurrence and spatial relations between objects. The context

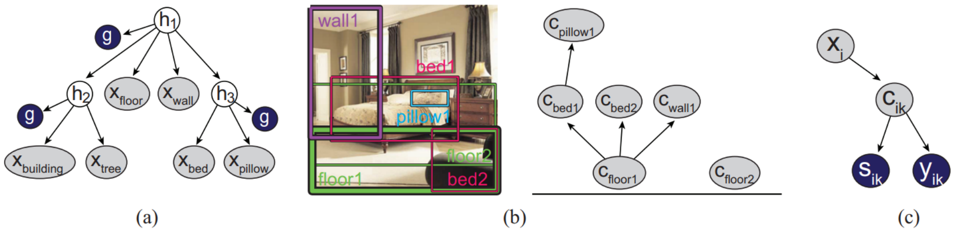


Figure 2.10: An illustrative example of a support context model for 6 object categories. The mechanism of employing context in this work is to use a graph model to combine different context information such as global context, object co-occurrence and spatial relations between objects. The figure was presented in [29].

computes the probability of each object's presence and the likelihood of each detection being correct. The results on SUN 09 [28] dataset demonstrates that context information plays very important role in scene understanding, for both object recognition and out-of-context object detection.

**Data augmentation.** Another work [38] shows that modeling appropriately the visual context surrounding objects is crucial to place them in the right environment. The model estimates the likelihood of a particular category of object to be present inside a box given its neighborhood, and then automatically finds suitable locations on images to place new objects and perform data augmentation. The model (Fig. 2.11) select an image for augmentation and 1) generate 200 candidate boxes that cover the image. Then, 2) for each box, find a neighborhood that contains the box entirely, crop this neighborhood and mask all pixels falling inside the bounding box; this "neighborhood" with masked pixels is then fed to the context neural network module and 3) object instances are matched to boxes that have high confidence scores for the presence of an object category. 4) Select at most two instances that are rescaled and blended into the selected bounding boxes. The resulting image is then used for training the object detector. The author further evaluates their context model for data augmentation the subset of the VOC12 dataset. The experiment demonstrates that context-driven data augmentation has more impact on categories for which visual context is crucial (aeroplane, bird, boat, bus, cat, cow, horse) than some general categories (chair, table, persons, train), since general categories like person, table etc. could appear in various scenes.

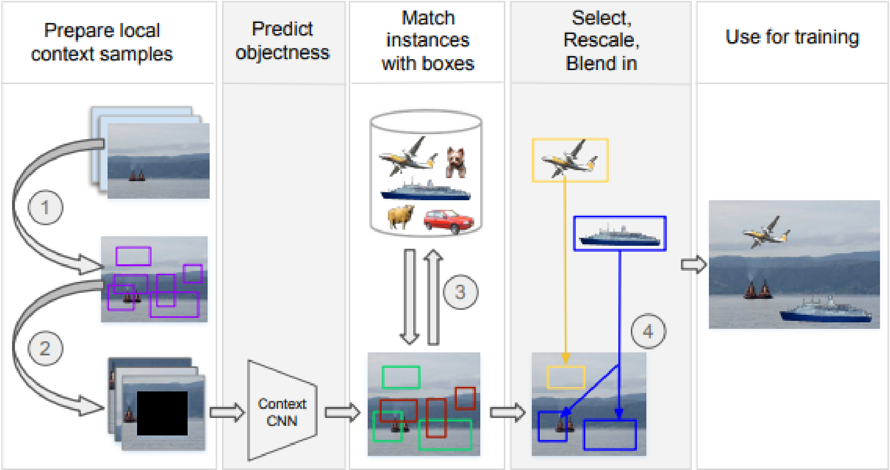

Prepare local context samples
Predict objectness
Match instances with boxes
Select, Rescale, Blend in
Use for training
1
2
3
4
Context CNN

Figure 2.11: Illustration of the data augmentation approach presented in [38]. The work used a context CNN to estimate the likelihood of a particular category of object to be present in certain local context.

Not only local context, prior knowledge also plays an important role for object detection task. Fang et al. [41] propose a novel framework of knowledge-aware object detection, which enables the integration of external knowledge such as knowledge graphs into any object detection algorithm.

Background knowledge can often be organized as a knowledge graph, which is a data structure capable of modeling both real-world concepts and their interactions. The framework considers a knowledge graph for modeling semantic consistency, which can better generalize to a pair of concepts even if they are not connected by any edge. The framework employs the notion of semantic consistency to quantify and generalize knowledge, which improves object detection through a reoptimization process to achieve better consistency with prior knowledge. It also provides context-aware approach for object detection task, which not only considers the visual context, but also considers the prior knowledge context.

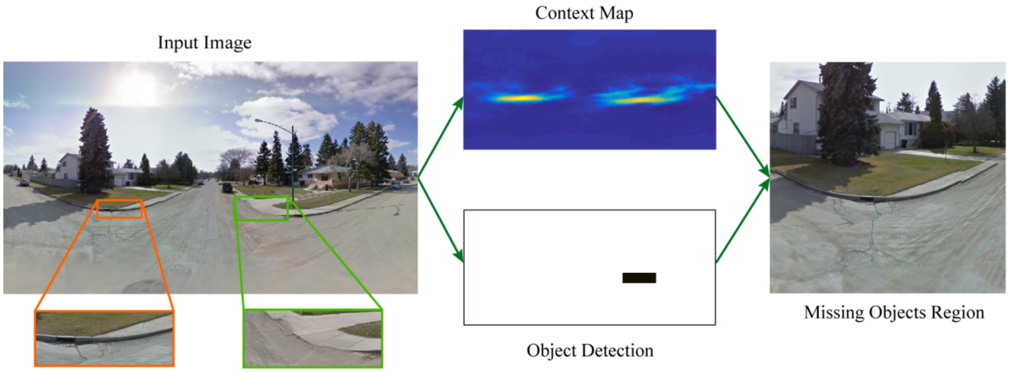


Figure 2.12: The approach to determine where objects are missing by learning a context model so that it can be combined with object detection results. The mechanism of employing context in this work is to train a Siamese Context Network to learn the pair-wise existence of curb ramps. The figure was presented in [115].

**Object detection.** Not only context can be used to detect the object, context can also help to predict where objects should exist, even when no object instances are present. Sun [115] perform a novel vision task: finding where objects are missing in an image. The author proposes a Siamese trained Fully convolutional Context network (SFC) (Fig. 2.12). The network first generate a context heat map using the context network $Q$. This map shows where an object should appear. Then Object detection results are generated by using any object detector. Convert detection boxes into a binary map by assigning 0 to the detected box region, 1 otherwise. This binary map shows where no objects are found. Furthermore, element-wise multiplication is performed between the context heatmap and the binary map. The resulting map shows the regions where an object should occur according to its context but the detector finds nothing. Finally cropping the high scored regions (above a preset threshold) from the image according to the resulting map. These are the regions where objects are missing. With the local context and co-occurrence of the curbs at street crossings, Context map from SFC network and detection results from object detector can be generated in parallel, which provides a more efficient and effective way to combine context information with target objects.

**Few shot detection.** A recent work [173] investigate utilizing the semantic context together with the visual information and introduce explicit relation reasoning into the learning of few-shot object detection. Word embedding is used to represent each class label. Semantic relation consistency is embedded between base class and novel class. If prior knowledge is given that the novel class “bicycle” looks similar to “motorbike”, can have interaction with “person”, and can carry a “bottle”, it would be easier to learn the concept “bicycle” than solely using a few images. Such explicit semantic relation context is even more crucial when visual context is hard to access. This few-shot detector is built on top of Faster R-CNN. A semantic space is built

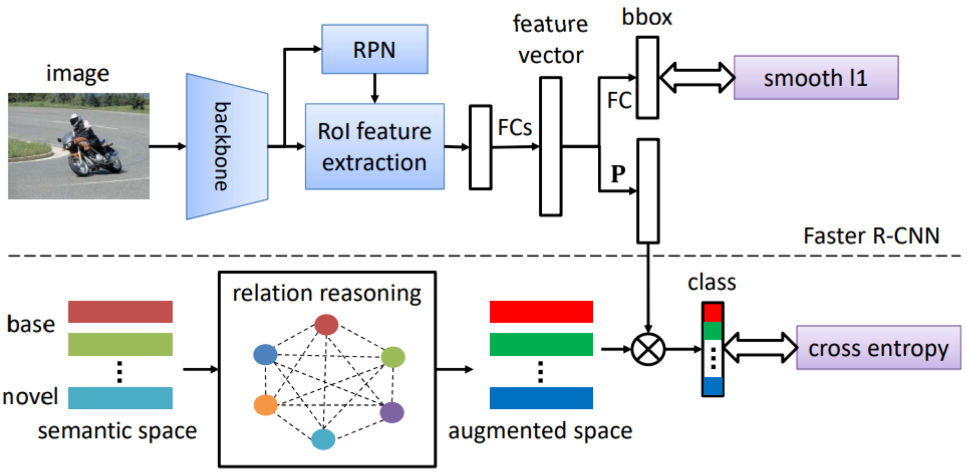


Figure 2.13: Overview of the SRR-FSD. The figure was presented in [173]. The mechanism of employing context in SRR-FSD is to model the semantic relation by building a semantic space, using exclusive semantic context from word embeddings.

from the word embeddings of all corresponding classes in the dataset and is augmented through a relation reasoning module. The overall framework is shown in Fig. 2.13.

### 2.3.2 Video-based Context Integration

In video-based context integration, spatial context and semantic context are carried in the temporal dimension. Video-based tasks heavily use temporal context with spatial relations between target objects or events to make the prediction. In this section, we review recent representative works that employed context information, and provide an overview of all the reviewed video-based context integration. Table 2.2 provides a summary of all the reviewed works in video-based context integration, in terms of tasks, backbone deep NN (DNN) models, employed context types, employed context levels,

Table 2.2: Video-based context integration.

| **Methods** | **Tasks** | **DNN Models** | **Context Types** | **Context Levels** | **Mechanisms** |
|---|---|---|---|---|---|
| Context R-CNN[8] | Object detection | ResNet | Spatial, Temporal | Global, Local | Memory bank |
| Tube Feature Aggregation Network[141] | Pedestrian detection | ResNet | Spatial, Temporal | Local | Feature aggregation |
| Contextual Graph Representation Learning[151] | Person search | ResNet/GCN | Spatial, Temporal | Global, Local | GCN |
| Contextual Boundary-aware Framework[128] | Temporal query grounding | Custom | Semantic, Temporal | Global, Local | Self attention |
| Hierarchical Temporal Network[162] | Temporal query grounding | Custom | Temporal | Global, Local | Semantic conditioned model |
| Spatio-temporal Progressive Learning[156] | Video action detection | VGGNet | Spatial, Temporal | Global, Local | GCN |
| Spatio-temporal Structural Model[174] | Video event recognition | / | Temporal | Local | Structural activity model |
| Hierarchical Context Learning[132, 133] | Video event recognition | Custom | Spatial, Temporal | Prior knowledge, Global, Local | Hierarchical context model |

and mechanisms for using context.

**Pedestrian detection.** Detecting heavily occluded pedestrians is crucial for real-world applications, such as autonomous driving systems. There are two main challenges for this task: (1) Heavily occluded pedestrians are hard to be distinguished from background due to missing/incomplete observations; (2) Detectors seldom have a clue about how to focus on the visible parts of partially occluded pedestrians. Although there are works try to solve the occlusion issue using attention, feature transformation, and part-based detection, they have not leveraged additional context information beyond a single image. A recent work [141] exploited the local context through temporal context of pedestrians in videos by aggregating local context features to enhance pedestrian detectors against occlusions. The model iteratively searches for its relevant local context along temporal order to form a context tube. Furthermore, the model resorts to local spatial-temporal context to match pedestrians with different extents of occlusions using a new temporally discriminative embedding module and a part-based relation module. Overall, the work employs spatial context, temporal context and global level context in combination to overcome the pedestrian occlusion issue, which also outperforms the context-free methods on benchmark datasets.

**Video event recognition.** Video event recognition aims to recognize the spatio-temporal visual patterns of events from videos. Recognizing events in surveillance videos is still challenging due to intra-class variation and low image resolution, etc. Different context information could help on solving these challenges. Context can be regarded as information that is not directly related to event recognition task, but it can be utilized to improve the

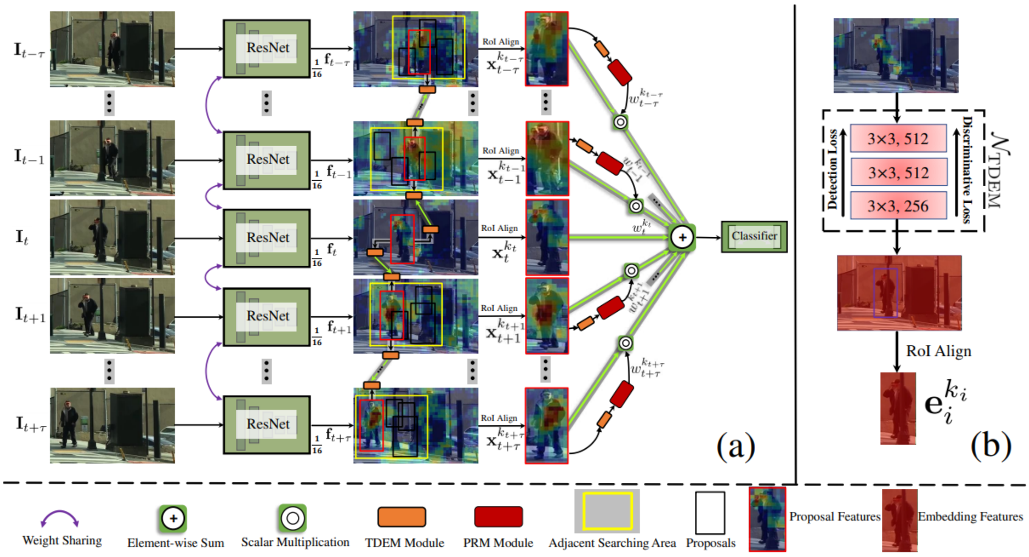


Figure 2.14: Overall framework of the TFAN. The mechanism of employing context is to iteratively search for its relevant local context along temporal order to form a context tube, by training a Feature Aggregation Network. The figure was presented in [141].

traditional data-driven and target-centered event recognition. Wang and Ji published a serial work [132, 133], focus on video event recognition task, by integrating multiple levels of contexts. The mechanism of the serial work is to build a hierarchical context model to incorporate different context (prior knowledge, spatial, temporal, semantic) for a better video event prediction and recognition.

Wang et al. [132, 133] defines three levels of context in video event recognition task: prior level, semantic level, and feature level. The prior level contexts capture the prior information of events, which is the prior knowledge such as location, time and weather, etc. These prior knowledge can indicate the possible scene states in the video. Temporal context is also treated as prior knowledge support in the event, which can provides support for the

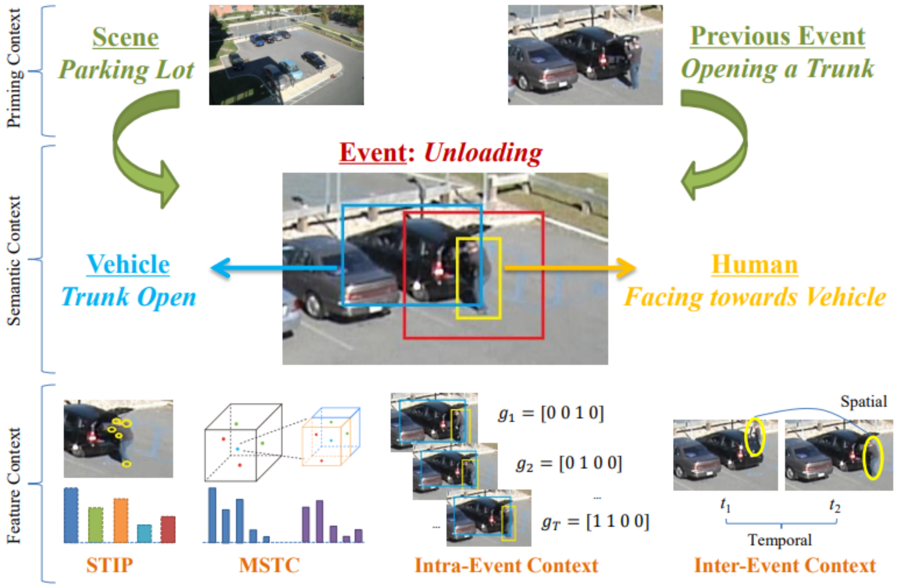


Figure 2.15: An example of incorporating contexts at different levels. A hierarchical context model is built to incorporate different context (prior knowledge, spatial, temporal, semantic) for a better video event prediction and recognition. The figure was presented in [133].

prediction of the current event given previous event. Semantic context can capture the semantic interactions among event entities, such as person get off the car, person open the trunk, etc. Feature level context in this work is defined as local level context (visual appearance) and semantic context (interaction). Temporal context is used to connect feature level context through the event. Wang et al.[132, 133] introduce a hierarchical context model to learn all these feature for a better video event prediction and recognition.

**Temporally grounding language queries.** The task of temporally grounding language queries in videos is to temporally localize the best matched video segment corresponding to a given language (sentence), as shown in Fig. 2.16. It requires both visual understandings and linguistic understand-

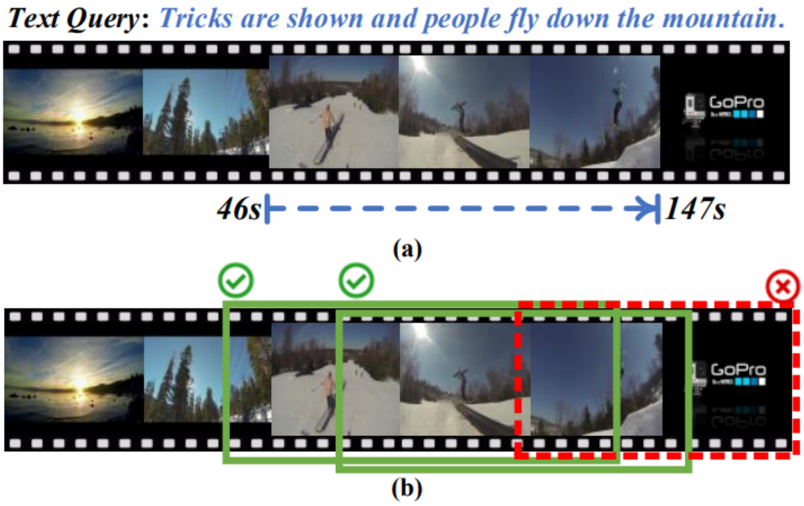

Text Query: Tricks are shown and people fly down the mountain.
46s
147s
(a)
(b)

Figure 2.16: (a) The task of temporally grounding language queries in videos. (b) Positive and negative training segments defined in anchor-based approaches given the sentence query in (a). The figure was presented in [128].

ings. Previous works use predefined sliding window to scan the video, which could ignores the precision of semantic boundaries. By using both semantic context and temporal context, it could provide more accurate boundaries. In order to cooperate both semantic context and temporal context in a video, Wang et al. [128] propose an end-to-end contextual boundary-aware model for temporally grounding language queries task, which aggregates semantic context and temporal context, by modeling the relationship between the current frame and its neighbors. The proposed context module operates on the layer which already integrates query and video information. It thus enables the network to "perceive" the surrounding local context and collect reliable contextual evidences before making predictions at the current step. This is different from previous context modeling, which only considers visual context but ignores the impact of semantic context. The temporal context dependency provides both semantic relation between objects and different local visual context compared to the background. With the aid of its local context, the activity is better localized. Temporal context and semantic context played as crucial cues for better precision in this framework.

### 2.3.3 Summary

In this chapter, we reviewed how context has been understand and integrated in context-based approaches for computer vision tasks. This survey covers recent context integration in both image-based tasks and video-based tasks. In our taxonomy of context, we categorized context in three major types and three context levels, and reviewed basic deep learning architectures and datasets used in context integration. Context information has been integrated and utilized over context-free methods, and it has been achieved great success and surpass the performance of context-free methods, in both image-based tasks and video-based tasks.

However, there are still space for further improvement and better way to incorporate the context in various visual tasks. In the following chapters, we will particularly study models on visual detection tasks and semantic segmentation tasks, and focus on the use of spatial context at different levels. We will start with a specific visual detection application example to design a context learning and utilization framework, and then generalize the framework to be applied to various visual detection tasks, possibly also with different base detection deep models. We will also explore how we can use the context framework for smart data collection and model refinement. Finally, we will generalize the general context framework to other visual tasks, in particularly an advanced semantic segmentation form called panoptic segmentation. We hope that the workflow of generalization can not only be practically used by researchers on similar related tasks, but also offer insights for integrating other forms of context such as temporal context information, and applying for more broader categories of tasks such as video-based tasks, which are not studied in this work.

# Chapter 3

# MultiCLU: Multi-stage Context Learning and Utilization for Storefront Accessibility Detection

In this chapter, we propose MultiCLU, a new multi-stage context learning and utilization approach for storefront accessibility detection task. MultiCLU consists of the following three components: Context in Labeling (CIL), Context in Training (CIT) and Context in Detection (CID). We collected our own Storefront Accessibility Image Dataset (SAI) and implement each context modules in MultiCLU using specific relations between the object categories: doors, knobs and stairs. We further design a Context in Evaluation (CIE) component, a new evaluation criteria specifically for this task, which could benefit the application in real world. Our experiments show that our framework can outperform the standard object detector in a large margin, especially for small objects like knob when apply the new evaluation criteria. In this chapter, we start with problem definition (Section 3.1) and our collected SAI dataset (Section 3.2). We will further provide details for each context component in Section 3.3. In addition, we provide details for how we refine the baseline detector Faster R-CNN (Section 3.4) and design a new evaluation criteria (Section 3.5). Our experimental results are provided in Section 3.6. This work has been published and presented at the ACM International Conference on Multimedia Retrieval (ICMR) in 2022 as a full oral paper [134].

## 3.1 Problem Statement

According to the IAPB Vision Atlas [1], there are 1.1 billion people living with vision loss in 2020 globally. Among them, 43 million people are blind, 295 million people have moderate to severe vision impairment, remaining people have mild or near vision impairment. Blind or low vision (BLV) people are facing different daily challenges. One of the obstacles they are facing in their daily life is to access essential activities, such as visiting local stores, visiting museums, and using public transportation facilities, etc. Helping BLV users to identify the accessibility of local stores in street environments can ease their daily burdens and improve their independence.

There are urban various image datasets for different computer vision tasks. Cityscapes [31] is a large-scale street level dataset that is mainly used for semantic urban scene understanding tasks. The Street View Text

Dataset, known as SVT [129], is another open source outdoor street level imagery for text detection and recognition of business signage and business names. However, both datasets don't include annotations for storefront accessibility features. For providing accessibility detection features in a complicated street-level environment, we identify three categories of objects in helping BLV people to identify the storefront accessibility: 1) doors (for store entrances), 2) Knobs (for accessing the entrances) and 3) stairs (for leading to the entrances). We further collected our own storefront accessibility image (SAI) dataset for detection and evaluation in this work.

Current labeling approaches in object detection tasks heavily rely on human labelers to create labels on their datasets. To obtain the consistency of the labels, there are predefined description of the target classes and instructions on how to draw labels on images. Usually tight bounding boxes are fit to the target objects. However, for small objects, these tight bounding boxes may not provide enough information for recognition, even for human observers (e.g., the doorknob in Fig. 3.1). But the object will have higher chance to be recognized as a knob if we consider its context (e.g. the door) where it is located. Related works [74, 68] also show that context information from the surroundings of small objects could provide important cues for successful detection. In this work, in the labeling stage, instead of performing relabeling by humans, we first apply an automatic approach to enhance the tight bounding box for each small object to include some local context information before the training stage.

In addition to the local visual context of an object, semantic context can also provide important information for detecting the object. For example,

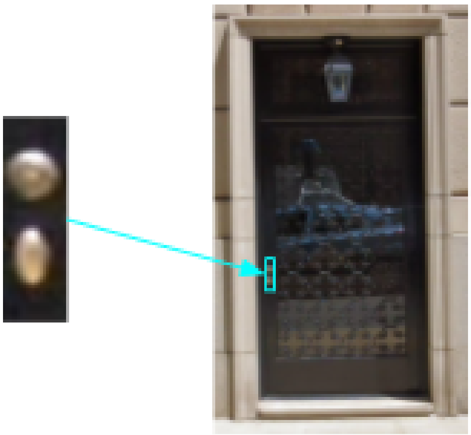


Figure 3.1: An example of the importance of contextual information for small object - the doorknob.

without looking at the image, and if we know that there is a knob in the image, we can easily guess there is a door in the image. To represent this kind of semantic context, word embeddings from Natural Language Processing (NLP) have been used in image classification task [24]. In order to align the visual context with semantic context in our machine learning model training task, we employed a Graph Convolution Network [61] to generate a semantic space and project the regional visual features into the semantic space for classification. Futhermore, objects do not appear in isolation. For our SAI dataset, doors and knobs are highly co-related not only in the semantic context, but also in the spatial context. As common senses, a doorknob must be inside a door frame, and a stair (if exists) should be under the door. We further utilize this type of spatial relation reasoning in the detection stage to refine the object classification before evaluation.

## 3.2 A Storefront Accessibility Image Dataset

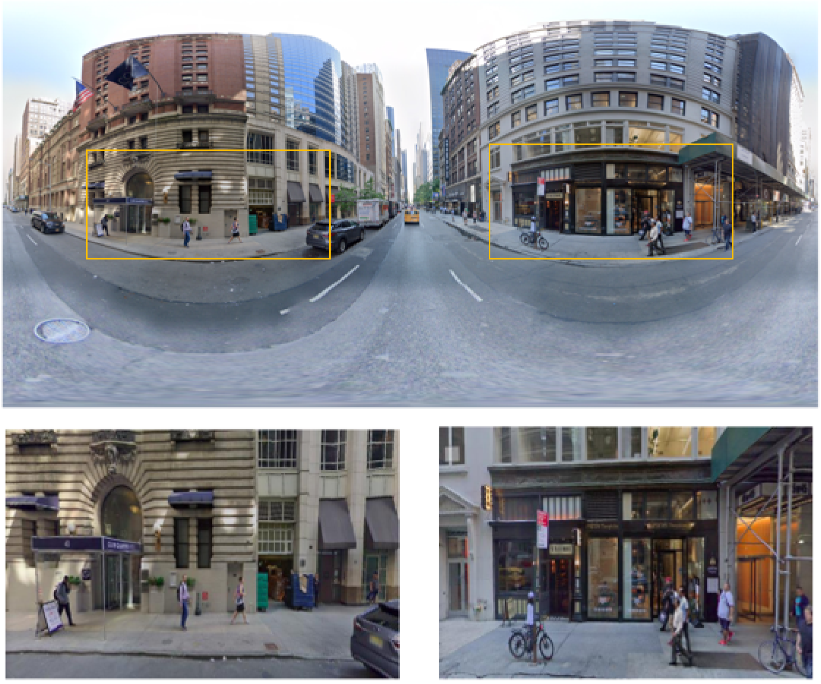


Figure 3.2: A formed panorama image and the cropped sub-images from Google Street View API of New York City. Top: The panorama image with all tiles. Bottom: cropped images from the middle 5x7 tiles as labeled in the panoramic image.

The storefront accessibility image dataset (Fig.3.2) is collected from Google Street View of New York city using Google Street View API [3]. We then use [17] to compose the panorama images. Each panorama image is formed of 16 (vertical) by 32 (horizontal) tiles and often captures building facade on both sides of a street in NYC. Then each formed panorama images are divided into two halves, each covers one side of facade. We cropped 5 (vertical) by 7 (horizontal) tiles in the center of each image in which storefronts are clearly seen and can be labeled easily.

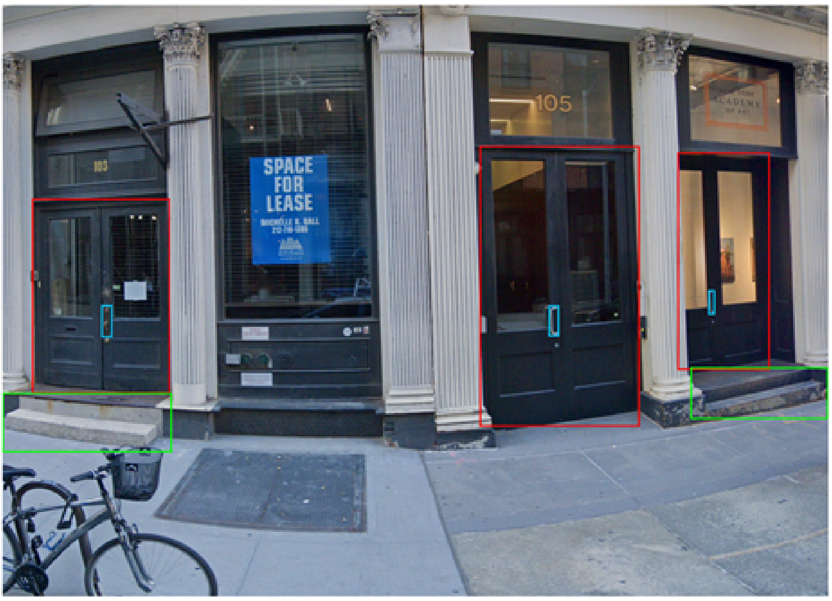


Figure 3.3: An example of labeled objects. Red: Ground truth bounding box of Door. Cyan: Ground truth bounding box of Knob. Green: Ground truth bounding box of stair.

Table 3.1: Statistics of collected storefront accessibility data.

| Dataset | # of Images | Doors | Knobs | Stairs |
|---|---|---|---|---|
| Train | 992 | 1885 | 1614 | 420 |
| Test | 110 | 233 | 126 | 141 |

We collected 1102 images in total and labeled the three main categories for accessibility (Door, Knob, Stair) using Labelbox[107]. Ten (10) percent of collected data were random sampled as the testing set while the remaining 90% of the data were used as the training set. Details of the data are shown in Table 3.1. Examples of labeled storefront objects in an image is shown in Fig. 3.3.

## 3.3 The MultiCLU Framework for Storefront Accessibility

We proposed MultiCLU: a multi-stage context learning and utilization framework to detect storefront accessibility objects. We make use of our general context framework by using specific relationships between three labeled categories: doors, knobs and stairs. Our muti-stage context learning and utilization (MultiCLU) framework (Fig. 3.4) uses Faster R-CNN [102] as the underling detection model (the detector) to extract features and propose candidate bounding boxes for object classes. The proposed MultiCLU framework explores various context information in four processing stages: Context in Labeling (CIL), Context in Training (CIT), Context in Detection (CID) and Context in Evaluation (CIE), in order to improve recognition performance. Local context around small objects, e.g., door knobs, are utilized in the CIL stage (Section 3.3.1) by automatically extending the bounding box of each doorknob (using the knob label) withing a door frame (using the door label where the knob belongs to), before the original images were fed into the detector. In the CIT stage (Section 3.3.2), we represent object labels using word embeddings extracted from a pretrained language model [97]. A contextual co-occurrence graph is built over the prior object appearance knowledge to describe the relation among different categories. A Graph Convolution Network (GCN) [61] is learned over the contextual graph and built a semantic space from word embeddings. Instead of using the original classification head of Faster R-CNN, we output feature vector from each region proposal and then project the region visual features into the semantic

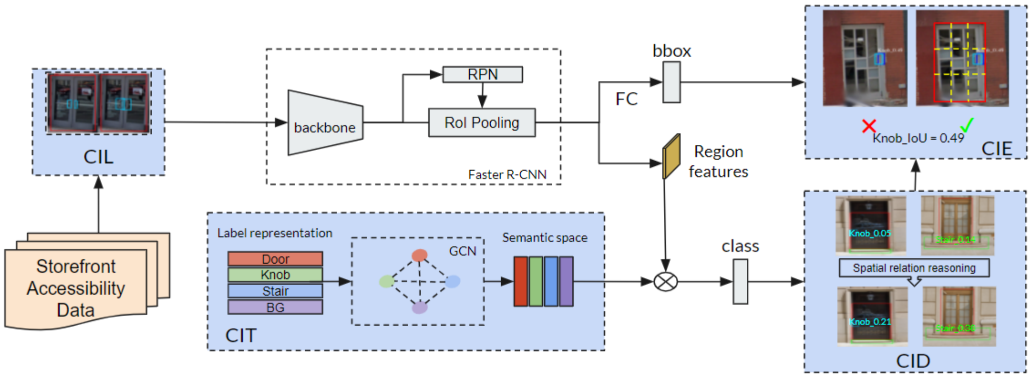


Figure 3.4: The architecture of our multi-stage context learning and utilization (MultiCLU) framework. Contextual components (in four stages) are shaded in light blue. CIL: Context in Labeling. CIT: Context in Training. CID: Context in Detection. CIE: Context in Evaluation. "⊗": dot product. "FC": Fully-connected layer.

space. Then in the CID stage (Section 3.3.3), We refine the confidence scores of detected object candidates using spatial relations among various objects that satisfy certain conditions. Finally we apply a new evaluation criteria for the knob category in the CIE stage (Section 3.5) to produce more applicable recognition results using application-related context information. In the following, we will detail each of the four components of our MultiCLU framework.

### 3.3.1 CIL: Context in Labeling

Starting from our original human annotated labels for knobs, we want to include more contextual information from the surrounding area of each knob, which could have important cues to help the MultiCLU framework to detect and recognize knob precisely. In order to achieve this, we automatically extend the bounding box of a knob within its door frame by using the in-

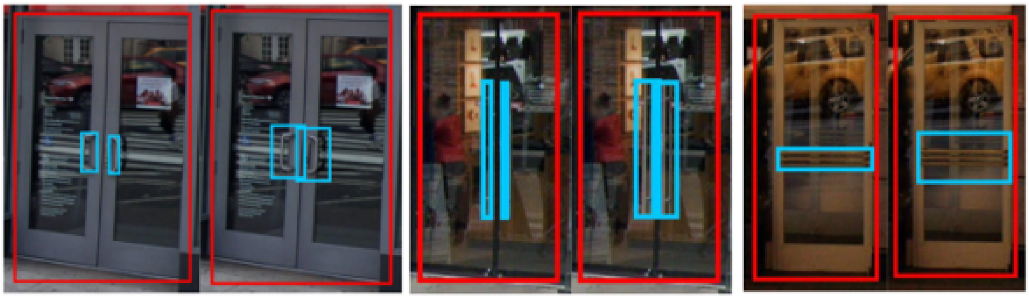


Figure 3.5: Three examples of Context in Labeling for different knob types. Left: extend both width and height. Middle: Only extend width. Right: Only extend height. In each example, the left image shows the original labels and the right image shows the extended labels.

formation of the labeled door the knob belongs to. We use the center of knob bounding box as the center, a certain percent of the door width (now we chose $\alpha = 20\%$) as the threshold for the minimal width of the extended bounding box for the knob, then the width of extended knob bounding box is give as:

$$w'_{knob} = \begin{cases} \alpha w_{door}, & \text{if } w_{knob} < \alpha w_{door} \\ w_{knob}, & \text{otherwise} \end{cases} \tag{3.1}$$

where $w_{knob}$ and $w'_{knob}$ denote the original and the updated widths of the ground truth knob label. Door height usually is longer than door width, so we use a smaller percentage ($beta$=15%) of the door height as the threshold; the new height of knob is calculated as:

$$h'_{knob} = \begin{cases} \beta h_{door}, & \text{if } h_{knob} < \beta h_{door} \\ h_{knob}, & \text{otherwise} \end{cases} \tag{3.2}$$

where $h_{knob}$ and $h'_{knob}$ denote the original and the updated heights of the ground truth knob label. Note that in order to keep the original shape of the knobs which have larger width or height, we only extend either the width or the height of a knob only if the width or the height satisfies the condition in Eq. 3.1 and Eq. 3.2. Restricting the new knob labels within the door frames is applied when extend original knob labels. Three examples are shown in Fig. 3.5. Also note that we keep both the original and the extended bounding boxes for each knob. Therefore each knob has two labels (of the same knob class), in order to improve the robustness of detection.

### 3.3.2 CIT: Context in Training

Kipf and Welling [61] first introduced the Graph Convolutional Network (GCN) to perform semi-supervised classification of nodes in a graph. GCN has also been used to solve computer vision tasks, such as image classification [24], visual relationship detection[32], object detection [150, 57] and scene graph generation [152, 58], etc. As described in [61], A graph $\mathcal{G}$ takes: (1) a feature description of all nodes: $H \in \mathbb{R}^{nxd}$, and (2) a relation descriptor between all nodes: $A \in \mathbb{R}^{nxn}$, as the input to learn a function $f$ over $\mathcal{G}$. Here n is the number of nodes, d is the dimensionality of the node feature. Then the updated node feature $H'$ is:

$$H' = f(H, A) \tag{3.3}$$

After applying a convolution operation, the function in Eq. 3.3 can be written as:

$$f(H, A) = \sigma(AHW) \tag{3.4}$$

where $\sigma$ is a non-linear activation function and $W$ is the weight.

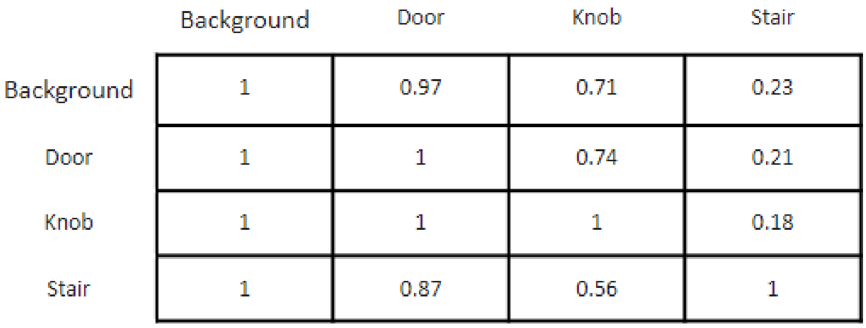
As shown in Fig. 3.4, the GCN network takes feature description of labels $H_{labels} \in \mathbb{R}^{nxd}$, and contextual graph $A \in \mathbb{R}^{nxn}$ as input, where $n$ is the number of labels (number of nodes) and $d$ is the dimensionality of the label word embedding (dimensionality of the node feature). $f_{regions} \in \mathbb{R}^{DxN}$ is the region features of all proposed region extracted from Faster R-CNN, where D is the dimensionality of the region features and N is the number of proposed regions. The output of the GCN network is represented as the label semantic space $H'_{labels} \in \mathbb{R}^{nxD}$. Inspired by [173], we project the region features $f_{regions}$ into semantic space $H'_{labels}$, then the final probability distribution $\mathbf{P}$ for object predictions is calculated as:

$$\mathbf{P} = softmax(H'_{labels} f_{regions}) \tag{3.5}$$

where $\mathbf{P} \in \mathbb{R}^{nxN}$, represents the class probability distribution for each proposed region.

The GCN uses relation descriptor $A$ to propagate information between nodes. For different applications, there are predefined relation descriptor $A$. However, there is no standard definition on generating $A$ for an object detection task. In order to model relationship between categories in our storefront accessibility image dataset, we built the contextual graph following the way described in ML-GCN [24] to define the relation descriptor, by using prior label appearance knowledge acquired from the training set. The co-occurrence between each pair of labels is described by the conditional probability, $P(L_j|L_i)$, which denotes the probability of occurrence of label

$L_j$ when label $L_i$ appears. $P(L_j|L_i)$ is not equal to $P(L_i|L_j)$, e,g., there must be a door if a knob appears, but there might be a knob if a door appears. Thus the contextual graph is asymmetrical. We count the occurrence of label pairs in the training set as prior semantic knowledge and generate the contextual graph built up by $A \in \mathbb{R}^{nxn}$, where n is the number of labels. Background label represents regions that do not belong to any of the categories. Fig. 3.6 shows the relation descriptor matrix generated from the SAI training dataset.



| | Background | Door | Knob | Stair |
|---|---|---|---|---|
| Background | 1 | 0.97 | 0.71 | 0.23 |
| Door | 1 | 1 | 0.74 | 0.21 |
| Knob | 1 | 1 | 1 | 0.18 |
| Stair | 1 | 0.87 | 0.56 | 1 |

Figure 3.6: Relation descriptor matrix generated from the SAI training dataset.

### 3.3.3 CID: Context in Detection

Information such as how objects are related to each other, whether there are spatial relations of objects or co-occurrences of objects in a natural scene, has been encapsulated in spatial context in our work. For our collected SAI dataset, the three category has very strong spatial relations. A knob can only appear inside a door frame. A stair, if exists, must be under a door, etc. Instead of using general topological relation, we leverage our spatial contextual reasoning component to model these relations by not only using prior knowledge from the training set (as in the CIT stage) but also the spatial relations of door vs knob and door vs stair to refine their confidence scores in detection before the predictions are sent to final evaluation. We apply an adaptive Bayesian approach to update confidence scores for recognized objects that satisfy the above spatial relations.

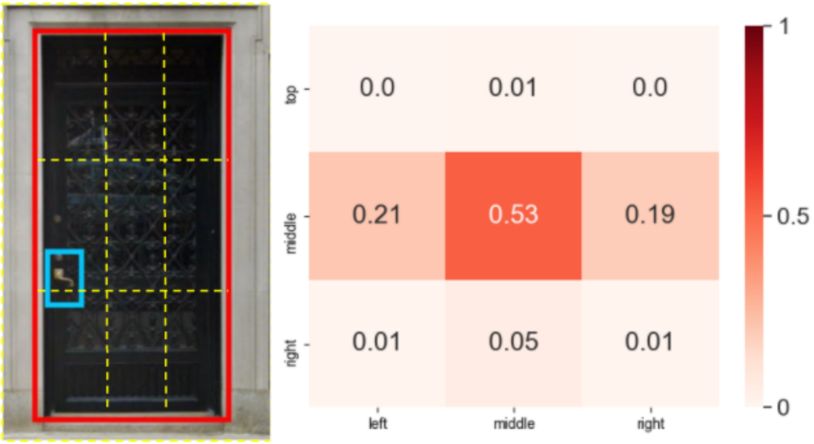


Figure 3.7: Knob probabilistic distribution inside a door frame using the training set. Left: 3x3 regions of a door. Right: 3x3 knob conditional probability distribution array.

To model the spatial relation between a knob and a door, we measure conditional probabilities of the knob distribution inside a door, by dividing the door frame into 3x3 equal-sized regions and count the labeled knobs falling in each region from the training data (Fig. 3.7). During detection, if a knob is predicted inside a predicted door (from the detector), the knob confidence score is updated as $C'_{knob}$:

$$C'_{knob} = \mu_1 C_{knob} + \mu_2 C_{knob|door} C_{door} \tag{3.6}$$

where $C_{knob}$ and $C_{door}$ are the original confidence scores of the predicted knob and door, respectively, and $C_{knob|door}$ is the conditional probability of where the knob is located inside the door frame, which is calculated from training data (Fig. 3.7 right). We take the weighted average of the original prediction score (from the detector) and the "deduction" score (from the Bayesian deduction), where $\mu_1$ and $\mu_2$ are the weights applied to them, respectively.

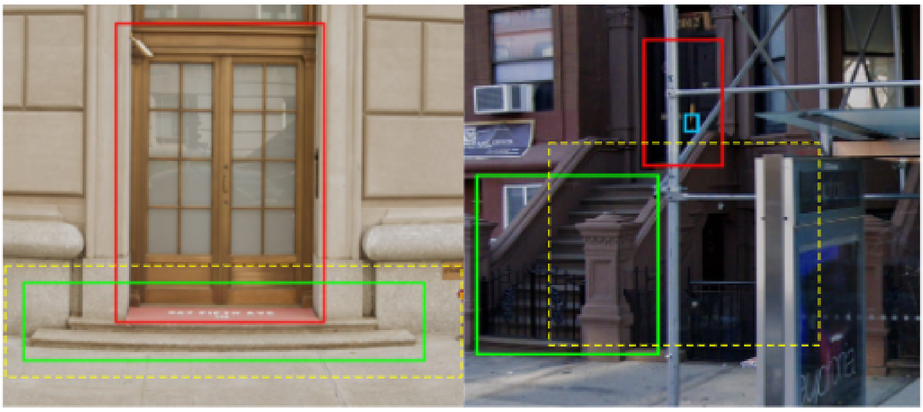


Figure 3.8: Two special cases for stair-door relations. left: door and stair have an overlapped area. Right: The stair is on the left bottom of the door due to camera perspective. Yellow dashed box: Search areas $S$.

A stair usually is located under the door. Because of the various reasons, such as special design layouts, camera perspectives and human labeling inaccuracy, there might be overlaps or spatial dis-alignments between these two categories (see Fig. 3.8). We thus define a search area to find whether there should be a predicted stair under a predicted door. The height of the search area $S$ is defined as:

$$S_{height} = height_{stair} + 0.2height_{door} \tag{3.7}$$

and the width is defined as:

$$S_{width} = width_{stair} + width_{door} \tag{3.8}$$

To check if a predicted stair connects to a predicted door, We search the stair centroid within the search area of the predicted door. If the centroid is located inside the search area, then the predicted stair is confirmed as under predicted door, and then we increase the confidence of the stair recognition to this updated stair confidence score $C'_{stair}$, as:

$$C'_{stair} = \alpha_1 C_{stair} + \alpha_2 C_{stair|door} C_{door} \tag{3.9}$$

where $C_{stair}$ and $C_{door}$ denote the original confidence scores of a predicted stair and a predicted door, respectively, and $C_{stair|door}$ is the conditional probability of a stair under a door, which is measured from the training data. $\alpha_1$ and $\alpha_2$ are the weights applied to the two terms.

Finally, we apply both the detection results of a stair and a knob as conditional terms to update confidence score of a door. The updated door confidence score $C'_{door}$ as:

$$C'_{door} = w_1 C_{door} + w_2 C_{door|knob} C_{knob} + w_3 C_{door|stair} C_{stair} \tag{3.10}$$

where $C_{door|knob}$ and $C_{door|stair}$ denote the conditional probabilities of a door given a knob and a door given a stair, respectively, which can be estimated from the training data. $w_1$, $w_2$ and $w_3$ are the weights applied to the three terms.

## 3.4 Refinement of Faster R-CNN

Currently the Faster R-CNN has had the following key steps to post-process the predictions: (1) Remove predictions with the background label; (2) Remove predictions with low confidence scores under the threshold of 0.05; (3) Remove empty boxes; (4) Apply non-maximum suppression to remove overlapping regions with a threshold of 0.5 (i.e., 50% of overlapping between two regions); and (5) Keep top K scoring predictions with a threshold of K=100 for all the objects. However, in Step (2) of the Faster R-CNN post-processing, certain amount of good predictions will be removed if the threshold of their confidence scores is set at 0.05.

To keep more positive predictions for applying the spatial context in this CID stage, our new post-process steps are modified as:

1. Remove predictions with the background label.
2. Remove empty boxes.
3. Apply non-maximum suppression with an overlapping threshold of 0.5.
4. Apply spatial relation reasoning to all the predictions as long as their original confident scores are greater than zeros.
5. Remove predictions with refined confidence score using the threshold of 0.05.
6. Keep top K scoring predictions with the threshold K=100 for all the objects.

We apply our spatial relation reasoning to the predictions from Faster R-CNN to refine the confidence scores using equations 3.6-3.10 in Step (4) of the CID stage. Then in Step (5), we apply the same score threshold (0.05 as the original Faster R-CNN) to remove low scoring predictions.

Note that if there are overlaps for proposed doors, knobs and stairs, we use the proposal with the max confidence as the base for each, then find all of those that overlap with the best proposal. We further only use the max confidence score of each category of one object to update all of the overlapped regions of another object (e.g., using the max confidence score from overlapped door predictions to update all the overlapped knob predictions) and vice versa. We propose max score door prediction and stair prediction from the overlapped prediction groups. Some doors could have multiple similar knobs labeled around same location, we propose the five highest scoring knobs from the overlapped prediction groups.

## 3.5 CIE: Context in Evaluation

When BLV people arrive a store independently, in order to open the door, they may want to know “the knob is on the left middle of the door” rather than “The knob is located at 1.5m high on the door”. And the estimated location could better benefit the people with disability. The commonly used evaluation metric for object detection task is the IoU evaluation, defined as:

$$IoU = \frac{\text{area}(B_{pred} \cap B_{gt})}{\text{area}(B_{pred} \cup B_{gt})} \tag{3.11}$$

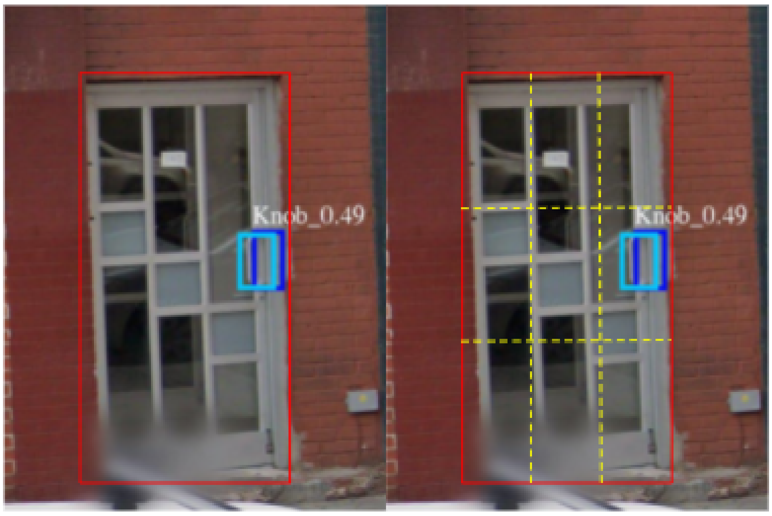


Figure 3.9: Comparison of the commonly used IoU evaluation and our Context in Evaluation (CIE) for knob. The IoU score for predicted knob is 0.49. Left: IoU at 0.5 threshold will treat it as false positive and not detected. Right: CIE uses door distributed regions to evaluate the knob, the predicted knob is accepted as a correct detection.

which measures the overlapping percentage of predicted bounding box $B_{pred}$ and ground-truth bounding box $B_{gt}$ of target object. It is not necessarily equivalent to describe the accuracy in the real world. In order to achieve this, we further define a new criteria in the CIE stage for the detection of knobs considering it is a small objects within the doors, which could help to better estimate the knob location. We segment a door into 3x3 regions as we did in the CID stage, shown also in right figure in Fig. 3.7. If the centroids of ground truth knob bounding box and the actual detection are within the same region, we count the knob as a true positive detection. An example has been shown in Fig. 3.9 when the IOU threshold is set to 0.5 (50% overlap between the prediction and the ground truth).

## 3.6 Experimental Results

In this section, we compare various evaluation results of our model on SAI dataset. First, we compare the mean average precision (mAP) over all categories of our SAI dataset when adding the first three contextual components with all the combinations of local CIL - Context in Labeling, semantic CIT - Context in Training, and spatial CID - Context in Detection. Then we compare the recall(%) and precision(%) per category. Furthermore, we provide results of our Context in Evaluation(CIE) approach for knob category comparing with the standard IoU@0.5 evaluation criteria.

Table 3.2: mAP over 0.5 IoU of all categories on the SAI dataset by applying various combinations of three contextual components (CIL, CIT and CID) to baseline Faster R-CNN.

| **Model** | **CIL** | **CIT** | **CID** | **mAP** | **Recall** |
|---|---|---|---|---|---|
| Faster R-CNN [102] | - | - | - | 53.1 | 69.4 |
| Single Component | √ | - | - | 62.2 | 80.4 |
| | - | √ | - | 55.1 | 74.1 |
| | - | - | √ | 52.8 | 72.2 |
| Two Components | √ | √ | - | 65.8 | 82.0 |
| | - | √ | √ | 56.0 | 76.0 |
| | √ | - | √ | 62.0 | 80.6 |
| All Components (M3) | √ | √ | √ | **66.4** | **85.2** |

If CIL has been applied to the baseline, We measure knob category using both the original labels and the CIL labels. If either label was detected for same knob, we only count as one detection to avoid duplication. We first applied each contextual component to the baseline method. As shown in Table 3.6, only applying one single contextual component among all the four can improve the baseline recall from +3% to 11%. mAP was improved when applying CIT and CIL individually. The recall was improved when applying CID component individually, even though overall mAP decrease slightly ( 0.3%). When applying combination of two contextual components, all combinations outperform the baseline method, in both mAP (+2.9% to +12.7%) and recall (+6.6% to +11.2%). In addition, we found that CIL component has greater impact than the other two components, which implies that the local contextual information used can help detect small objects more accurately in this SAI task. Comparing the results between CID only and CIT plus CID, CID have a positive impact on the CIT component, and further improved both mAP and recall. When apply CID with the CIL component, although both mAP and recall outperform the baseline with large margins (10%+), mAP actually decreases slightly ( 0.3%) comparing to apply CIL only. When applying all the three components to the baseline detector, leading to our MultiCLU approach with three components (M3), the best result is achieved for both mAP and recall, where mAP improves from 53.1% to 66.4% ( +13.3%) and recall improves from 69.4% to 85.2% ( +15.8%).

In order to better understand how effective our contextual components to each category are, we further compare the precision (%) and recall (%) measures per category with various combinations of the four components. First we added a single contextual component to the baseline Faster R-CNN. As shown in Table 3.6, CIL has the best performance on recall for door and stair, and with a great improvement on knob with 23.9% on precision and 30.2% on recall respectively. Our CIT component slight outperforms the baseline on precision for all categories, with  5.6% and  6.3% recall improve-

Table 3.3: Results on recall(%) and precision(%) per category for various combinations of the four contextual components.

| **Model** | **Precision** | | | **Recall** | | |
|---|---|---|---|---|---|---|
| | Door | Knob | Stair | Door | Knob | Stair |
| Faster R-CNN (FR) | 75.6 | 17.7 | 66.0 | 87.5 | 47.6 | 73.1 |
| +CIL | 78.2 | 41.6 | 66.8 | 88.8 | 77.8 | 74.5 |
| +CIT | 77.8 | 19.1 | 68.5 | 89.7 | 53.2 | 79.4 |
| +CID | 74.9 | 16.2 | 67.2 | 88.4 | 53.6 | 74.5 |
| +CIL+CIT | 78.4 | 50.4 | 68.5 | 91.4 | 75.6 | 79.0 |
| +CIT+CID | 78.2 | 20.6 | 69.2 | 90.3 | 56.4 | 81.3 |
| +CIL+CID | **78.8** | 40.4 | 66.8 | 88.8 | 78.5 | 74.5 |
| +CIL+CIT+CID (M3) | 78.0 | **51.2** | **70.0** | **92.3** | **80.4** | **83.0** |
| FR+CIE | 75.6 | **94.2** | 66.0 | 87.5 | 74.6 | 73.1 |
| M3+CIE (**MultiCLU**) | 78.0 | 83.2 | 70.0 | 92.3 | **90.4** | 83.0 |

ment on recall of knob and stair respectively. Although CID decreases the precision a little bit for all categories, the recall improves for all category from 0.9% to 6%. Our proposed method with the first three components (CIL+CIT+CID), which denoted as M3 in Table 3.6, achieves the best result for both precision and recall compare to other combinations. Not only the knob category has great improvement on both precision ( +33.5%) and recall ( +32.8%), both door and stair also achieve 2.4% and 4% improvement on precision, and +4.8%, +9.9% improvement on recall, respectively.

We further compare the result of our new evaluation criteria on knob category between the baseline method and our M3 method. The baseline model can achieve 94.2% on precision and 74.6% on recall on the knob when we apply the new evaluation approach (Section 3.5). Our full model (M3+CIE, which leads to the full MultiCLU model) achieves 90.4% (+15.8%) on recall but 83.2% (-11%) on precision comparing to the baseline with CIE component. This is because all the knobs detected from baseline Faster R-CNN where the IoU with ground truth is lower than 0.5 will be included when CIE is applied, hence the precision is higher and recall is also improved. Our full model achieves higher recall because the model have more detected knobs with contextual components, compared to the baseline model. Based on our experience with BLV people and storefront accessibility labeling with volunteers [2, 77], they prefer higher recall and can tolerate slightly lower precision. Also note that the final MultiCLU model achieves the best performance for all categories in both precision and recall with CIE than without CIE.

## 3.7 Significance Test

To assess the enhancements introduced by our MultiCLU framework in comparison to the baseline Faster R-CNN, we conducted significance tests on both models. Our methodology involved randomly sampling 10 sets of training-testing pairs, running both MultiCLU and Faster R-CNN on each set, and subsequently subjecting the testing results to significance tests for both mean Average Precision (mAP) (refer to Table 3.4) and recall (refer to Table 3.5). This rigorous evaluation approach provides statistical insights into the comparative performance of the two models, contributing to a more comprehensive understanding of the significance of our proposed MultiCLU framework.

Table 3.4: The precision of the significance test. FR: Faster R-CNN; M3: MultiCLU.

| **mAP** | **Set 1** | **Set 2** | **Set 3** | **Set 4** | **Set 5** | **Set 6** | **Set 7** | **Set 8** | **Set 9** | **Set 10** |
|---|---|---|---|---|---|---|---|---|---|---|
| **FR** | 53.1 | 48.4 | 53.9 | 51.8 | 44 | 51.1 | 55.1 | 50.6 | 53.8 | 47.4 |
| **M3** | **66.4** | **60.1** | **64.1** | **66.7** | **58.4** | **65.1** | **63.7** | **62.3** | **62.7** | **58.6** |

Table 3.5: The recall of the significance test. FR: Faster R-CNN; M3: MultiCLU.

| **Recall** | **Set 1** | **Set 2** | **Set 3** | **Set 4** | **Set 5** | **Set 6** | **Set 7** | **Set 8** | **Set 9** | **Set 10** |
|---|---|---|---|---|---|---|---|---|---|---|
| **FR** | 69.4 | 65.9 | 70 | 67.4 | 59.6 | 68.6 | 66.3 | 64.3 | 70.8 | 58.9 |
| **M3** | **85.2** | **79.1** | **85.2** | **80.8** | **76.6** | **82.2** | **85.7** | **82.8** | **81.6** | **77.5** |

Analyzing the mean Average Precision (mAP) (Table 3.4) and recall (Table 3.5) across the 10 sets of training-test data, our MultiCLU model consistently outperforms Faster R-CNN. The statistical evaluation, as presented in Table 3.6, involves assessing the t-value and p-value for both mAP and recall. The obtained p-values, indicating the probability of observing the given results if there were no actual improvement, affirm the statistical significance of our MultiCLU framework's superior performance over the baseline Faster R-CNN. With p-values significantly below the conventional threshold of 0.05, our MultiCLU framework demonstrates robust and substantial improvements, attaining p-values of 1.6e-7 for mAP and 2.3e-8 for recall, underscoring the reliability and significance of the observed performance gains.

Table 3.6: The significant values for MultiCLU vs. Faster R-CNN.

| **Model** | **Evaluation** | **t-value** | **p-value** |
|---|---|---|---|
| ultiCLU vs. Faster R-CNN | mAP | 8.24 | **1.6e-7** |
| | Recall | 9.39 | **2.3e-8** |

## 3.8 Summary

In this chapter, we have proposed MultiCLU: a new multi-stage context learning and utilization approach for storefront accessibility detection, in order to benefit BLV people for their daily life. We collected our own storefront accessibility image dataset SAI with three object categories: door, knob, and stair. We applied our MultiCLU framework over the Faster R-CNN and demonstrated the superior performance of our approach with various combinations of the four contextual components.

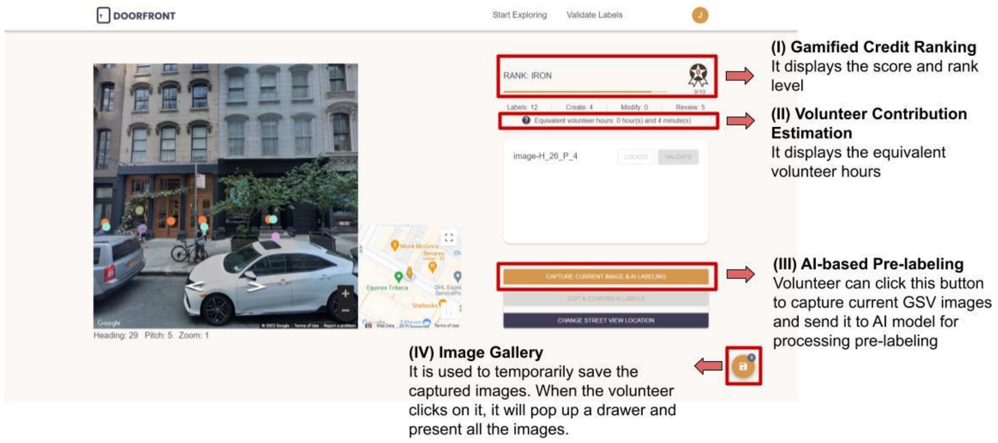
In next chapter, we will show an AI-enabled, web-based storefront accessibility annotation and localization platform, which is integrated with our MultiCLU storefront accessibility detection engine, to collect more data in a more efficient way, and to improve the detection engine in return.

# Chapter 4

# Smart DoorFront: Data Collection and Model Refinement with MultiCLU

In the past, our group has proposed a solution to collect large-scale accessibility data of New York City (NYC) storefronts using a crowdsourcing approach on Google Street View (GSV) panoramas. A web-based crowdsourcing application, DoorFront, has been developed [77], which enables volunteers not only to remotely label storefront accessibility data on GSV images, but also to validate the labeling result to ensure high data quality. In this work, with the MultiCLU storefront accessibility detection engine, we further significantly improve the efficiency of data collection and user engagement in our new AI-enabled Smart DoorFront platform by designing and developing multiple important features, including a gamified credit ranking system, a volunteer contribution estimator, an AI-based pre-labeling function, and an image gallery feature. For achieving these, we integrate a specially designed deep learning model using the MultiCLU framework into the Smart DoorFront. We also introduce an online machine learning mechanism to iteratively train the MultiCLU model, by using newly labeled storefront accessibility objects and their locations in images. In this chapter, we will discuss each features (Section 4.1) and how we integrate our MultiCLU storefront framework in our platform (Section 4.2). Experimental results are provided at the end. More details especially in user studies can be found in our paper in the CSUN Journal on Technology & Persons with Disabilities [135].

## 4.1 DoorFront Platform


DOORFRONT
Start Exploring
Validate Labels
J
RANK: IRON
9/10
Labels: 12
Create: 4
Modify: 0
Review: 5
Equivalent volunteer hours: 0 hour(s) and 4 minute(s)
image-H_26_P_4
LOCKED
VALIDATE
CAPTURE CURRENT IMAGE & AI LABELING
EDIT & CONFIRM AI LABELS
CHANGE STREET VIEW LOCATION
Heading: 29 Pitch: 5 Zoom: 1

**(I) Gamified Credit Ranking**
It displays the score and rank level

**(II) Volunteer Contribution Estimation**
It displays the equivalent volunteer hours

**(III) AI-based Pre-labeling**
Volunteer can click this button to capture current GSV images and send it to AI model for processing pre-labeling

**(IV) Image Gallery**
It is used to temporarily save the captured images. When the volunteer clicks on it, it will pop up a drawer and present all the images.

Figure 4.1: The interface of exploration page for the Doorfront platform.

DoorFront is a web-based application that combines Google Street View and crowdsourcing, with an interactive interface and a user-friendly labeling tool [77]. There are two main pages in the original DoorFront, namely an Exploration page and a Labeling page. Volunteers not only can virtually walk through New York City with embedded interactive Google Street View provided on the Exploration page; but they can also label storefront accessibility data with the functional and user-friendly labeling tool. Even though the feedback from crowd volunteers has demonstrated its usability and high potential for data collection, the process of labeling is still relatively labor intensive. Our studies show that there are two key factors that influence the effectiveness of the data collection: the number of volunteers participating in DooFront and the time they spent in the labeling process. To address these issues, we have made significant improvements to the DoorFront application, leading to Smart DoorFront, which includes four major new features: gamified credit ranking, volunteer contribution estimation, AI-based pre-labeling and image gallery. We will discuss each of them in the following.

**Credit Ranking System**. Inspired by gamified settings, we designed and implemented a gamified ranking system (Fig. 4.1, part I) and a leaderboard for the volunteers. In this project, we define seven different rank levels and their corresponding cumulative credits. The specific names of each level and the details of the accumulated credits for each phase are shown in Table 4.1.

Table 4.1: Rank levels and accumulated credits.

| Level | Iron | Bronze | Silver | Gold | Platinum | Diamond | Challenger |
|---|---|---|---|---|---|---|---|
| Credits | 0 - 9 | 10 - 49 | 50 - 299 | 300 - 999 | 1000 - 1999 | 2000 - 4999 | 5000 - 9999 |

Volunteers will receive credits through three contributed operations: annotating new Google Street View images, correcting other volunteers' annotations, and reviewing other volunteers' annotations. To further increase the entertaining nature of DoorFront, we also develop a treasure hunt feature that allows volunteers to earn extra ten credits whenever they find a treasure; once DoorFront is initialized, all treasures will automatically be hidden in random areas of New York City. Therefore, the locations of the treasures are completely different each time, and volunteers are unable to gain these extra points by memorizing the locations of the treasures.With respect to leaderboard, it will show volunteers according to their levels. In this atmosphere, we believe volunteers will become competitive and spend more time in data collection to advance their level.

**Volunteer Contribution Estimation**. Crowdsourcing brings us flexibility so we can distribute the data collection tasks to volunteers. Volunteers from each borough in New York City can collect storefront accessibility data in their own communities through DoorFront. However, using the original DoorFront platform, we were unable to recruit a large number of volunteers to participate. We needed to address how to attract more volunteers to DoorFront. In the initial version, we decided to award volunteer certificates to volunteers through our collaboration with Lighthouse Guild, a vision and health care organization. They provided volunteer certificates to individuals who have made large contributions of time on the platform. However, the algorithm we used did not work well to calculate the equivalent volunteer time. The core idea was calculating the number of images collected by the volunteers. We assume that, on average, volunteers spend one minute to annotate each image without the assistance of the AI model. The shortcoming of this algorithm is obvious. Since the number of labels in each image is different, our algorithm does not reflect well the effort of volunteers and the time they spend.

Therefore, we decided to design a new volunteer contribution estimator to better calculate volunteer effort. First, we rebuilt the DoorFront's credit management system. With this improvement, we are now able to monitor the number of labels annotated by volunteers, and equivalent-volunteer-time is determined by the number of labels. Second, we implemented a small widget to showcase the volunteer's effort in real-time (Fig. 4.1, part II). In addition, to further encourage volunteers to promote our application, we also provide sharing buttons on different social media applications such as Meta and Twitter, to share their contributions with their friends. With these improvements, we believe that more and more younger volunteers, especially middle or high school students, will be interested in participating in our study.

**AI Pre-labeling**. One of the key issues we needed to address is to reduce the time for annotation by a volunteer. On average, it takes at least one minute to manually annotate a storefront image from scratch using our DoorFront interface. There are three steps in the annotation process: (1) identify a storefront accessibility object; (2) annotate the object with a bounding box; and (3) add a subtype if the object is a door or door handle. Volunteers need to repeat these three steps until they label all the storefront accessibility objects in a scene, hence the labeling task is still time-consuming.

In order to further improve the efficiency of data collection, we enhanced

DoorFront with an AI-based pre-labeling function(Fig. 4.1, part III). With AI support, DoorFront can perform pre-labeling once a volunteer captures a new Google Street View image. This means that they do not need to label the image from scratch and the only thing they need to do is validate the results predicted by our AI model. Compared to the initial workflow, we now have only two steps: (1) Verify and correct the annotations labeled by the AI model; and (2) Add subtypes. Based on the outstanding performance of our model, we can skip the first step in most cases, which dramatically reduces the annotation time.

**Image Gallery** To maximize the utilization of the AI model, we modified the way that we save Google Street View images. In the initial version, DoorFront only allowed volunteers to label one image at a time. Now with the Smart DoorFront, volunteers can capture multiple images while they are virtually walking along with the street. Those images will be temporarily stored in an image gallery (Fig. 4.1, part IV) and then sent to the AI model for pre-labeling processing. Volunteers can then validate all images at once, without frequently switching among different web interfaces, which greatly reduces their labeling time.

Furthermore, we store the untagged images in our remote database. With this information, volunteers will be able to access these images again, regardless of the last time they exited the application. Furthermore, our application will send notifications to remind volunteers that they forgot to annotate these images.

In the next sections, we will describe the enabler of the aforementioned features: the integration of the deep learning model and an iterative learning approach.

## 4.2 The Use of MultiCLU

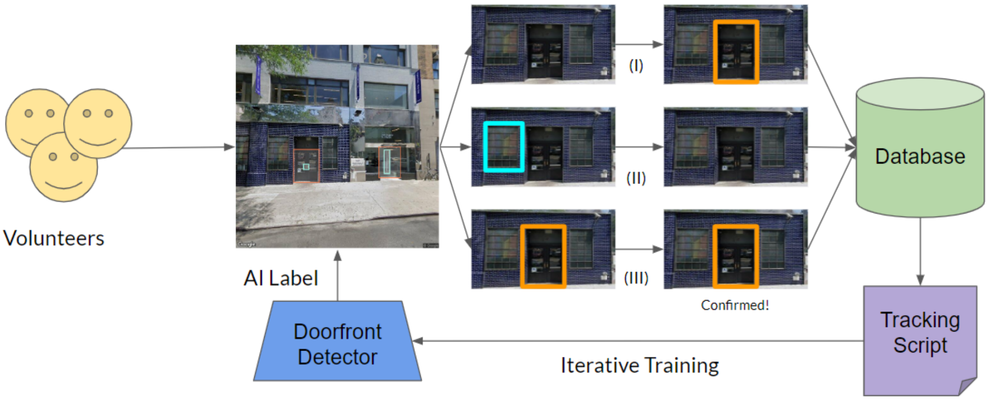


Figure 4.2: Overall pipeline of MultiCLU-based labeling in DoorFront.

In order to improve the efficiency of the storefront accessibility labeling process, we integrate a specially designed deep learning model MultiCLU [134] into DoorFront. Our MultiCLU model is implemented by integrating the state-of-the-art object detector, Faster R-CNN [102], with contextual relationships among storefront accessibility objects, in order to improve the accuracy of image detections (Fig. 4.2, part III). For example, if we know there is a knob in the image, we can easily guess there should be a door in the image. Furthermore, a knob must appear inside a door frame, and if a stair exists, it should be under the door. Our MultiCLU model utilizes these contextual relationships to improve our detection results. The overall pipeline of MultiCLU in DoorFront is shown in Fig. 2. When a volunteer captures an image, MultiCLU will detect storefront accessibility objects within a few seconds before user labeling. Volunteers can then validate or edit the labels which are pre-labeled by MultiCLU. Our platform will record three main types of labels: (1) Added labels from volunteers (Fig. 4.2. part I); (2) Removed labels from volunteers (Fig. 4.2. part II); and (3) Validated AI labels (Fig. 4.2. part III). Our model is further trained on modified labels which are corrected by volunteers, to further improve the performance.

## 4.3 Iterative Training

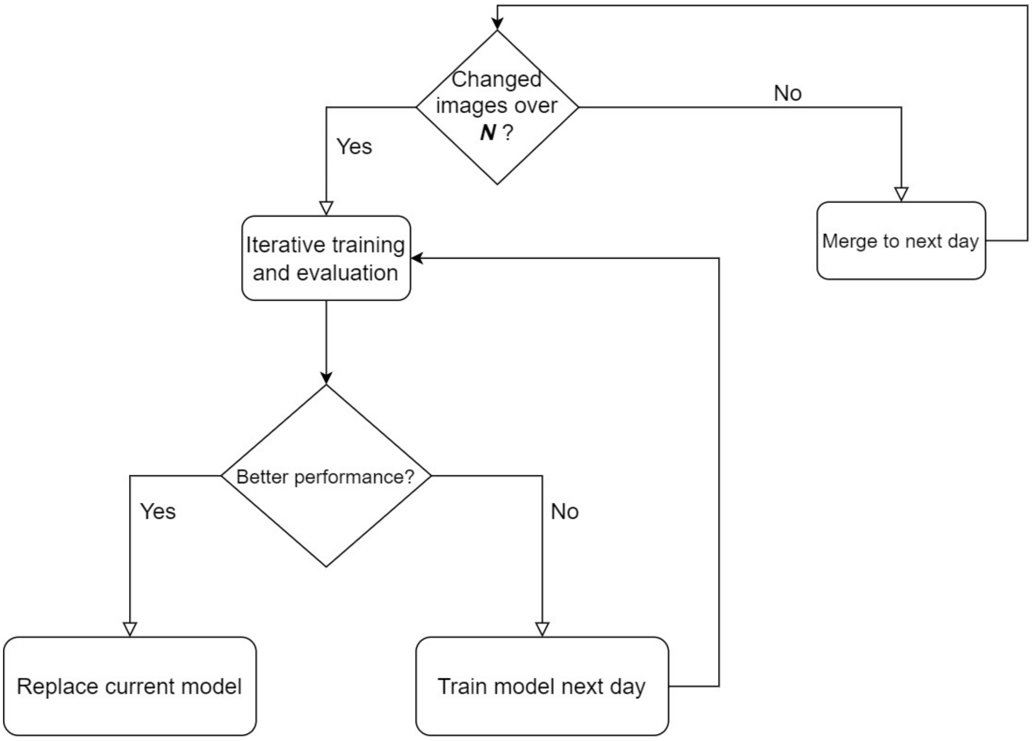


Figure 4.3: The iterative training process for the Doorfront platform.

We also introduced a training automation mechanism to iteratively train our MultiCLU model. The model will start iterative training automatically when a certain amount (N) of new images has been recorded with new labels. As shown in Fig. 4.3, if we achieve better performance after training, we will replace the current model, otherwise we will start another training process the next day and keep the current model. We also use a data aggregation process to improve the robustness of the detection. When the (n+1)_th iteration starts, we accumulate the previous training dataset from n_th training step, where n denotes current training step. And then we use the combined dataset to refine the current model.

## 4.4 Experimental Results

Table 4.2: The training set of collected storefront accessibility dataset over time.

| Category | Initial Training Set | After Day 1 | After Day 2 | After Day 3 |
|---|---|---|---|---|
| Door | 1225 | 1532 | 1719 | 1913 |
| Knob | 863 | 962 | 1060 | 1173 |
| Stair | 270 | 346 | 422 | 475 |

Table 4.3: The testing set of collected storefront accessibility dataset over time.

| Category | Initial Testing Set | After Day 1 | After Day 2 | After Day 3 |
|---|---|---|---|---|
| Door | 1080 | 1132 | 1168 | 1229 |
| Knob | 887 | 905 | 928 | 979 |
| Stair | 197 | 209 | 224 | 240 |

We evaluated our iterative training mechanism using labels collected within three consecutive days. For new labels from each day, we randomly select 80 percent of the dataset as the training set to refine the previous

MultiCLU model, and the remaining 20 percent were added to the current testing set (Table 4.2 and Table 4.3). We accumulated both training and testing data into our previously collected storefront accessibility and localization dataset. We only used the accumulated labels from three consecutive days to refine our model, which could improve the robustness of the detection. We report both precision (Table 4.4) and recall (Table 4.5) of the 3-day iterative training. We observed that both precision and recall for all the categories were improved, with precision increasing from +1.1% to +3.7% and recall from +1.3% to +5.2%, respectively. With a limited number of volunteers before we have formally published the Smart DoorFront app, we only provide results for three consecutive days. Initially, as we add more diverse and informative data to a training set, the model's performance tends to improve. However, there comes a point where the model has already learned the patterns present in the data, and adding more data may not contribute substantially to further improvement, such as overfitting, where the model becomes too specific to the training data and performs poorly on new, unseen data. We will be able to test how much improvement the model can achieve once we have more volunteers to label the data, for example, for months. We can also carefully monitor the model's performance and replace with the best performed model by applying out iterative training mechanism.

## 4.5 Summary

In this chapter, we introduce our new Smart DoorFront platform, which is building on our previous DoorFront Platform [77]. We utilize our MultiCLU

Table 4.4: The precision for the initial model and 3-day iterative trained models.

| Category | Initial Model | Day 1 | Day 2 | Day 3 |
|---|---|---|---|---|
| Door | 78.7 | 79.5 | 83.2 | 83.2 |
| Knob | 79.0 | 78.8 | 81.6 | 82.7 |
| Stair | 81.2 | 81.2 | 81.6 | 82.3 |

Table 4.5: The recall for the initial model and 3-day iterative trained models.

| Category | Initial Model | Day 1 | Day 2 | Day 3 |
|---|---|---|---|---|
| Door | 88.2 | 88.9 | 89.7 | 90.2 |
| Knob | 85.4 | 85.4 | 86.7 | 88.0 |
| Stair | 77.6 | 78.0 | 82.6 | 83.2 |

model for storefront image detection, which is built upon the state-of-the-art object detector and uses of context information among storefront accessibility objects. We also introduce an iterative training mechanism to improve the accuracy and robustness of our deep learning model. Our new platform not only optimizes user experience, but also significantly improves the efficiency of storefront accessibility labeling process with our deep learning model. We will continue gathering feedback from volunteers and develop a mobile app for BLV users to navigate to store entrances, using the collected storefront accessibility and localization data. We will also integrate our deep learning model into a mobile app in the future, to better help BLV users to improve their independence.

Admittedly, however, this specifically designed MultiCLU framework can only be applied to the storefront accessibility detection task. The question is: Can we extend MultiCLU to a general framework that can utilized for various different visual detection tasks with much change of the code? In next chapter, we show how the MultiCLU framework can be generalized not only to various visual detection tasks, but also be integrated with different base detection models.

# Chapter 5

# GMC: A General Multi-stage Context Framework

To generalize our specifically designed MultiCLU for storefront accessibility detection [134] to various other visual detection tasks, we present GMC, a general multi-stage context learning and reasoning framework with various deep learning models, applicable to various visual detection tasks, and therefore offering greater flexibility and adaptability without requiring extensive code changes. As an extended version of our previously published work [136], the GMC framework demonstrates the versatility and adaptability of our context components by successfully applying them to different deep learning models with minimal modification. The pedestrian detection task is greatly enhanced with more categories of contextual objects and includes all the three stages of context reasoning.

The overview of the general framework is shown in Section 5.1. We then describe each contextual component in details (Section 5.2.1 to Section 5.2.3).

Furthermore, we discuss how the general framework work with various deep learning network architectures with minimal modification of the code in Section 5.3. Our experimental results are provided in Section 5.4 for the SAI dataset and then in Chapter 6.3 for the CityPersons dataset and its extension. This part of work has been submitted to the Elsevier journal Computer Vision and Image Understanding (CVIU) [137].

## 5.1 Overview of the GMC Framework

Our proposed GMC framework, as detailed in Fig. 5.1, consists of three key context components: local context representation, semantic context fusion, and spatial context reasoning. These components can be applied individually or in combination with a given visual detection network architecture to enhance object detection performance.

The local context representation component (Section 5.2.1) focuses on capturing local contextual information specific to the objects of interest. By incorporating local context features in the data labeling stage, this component improves the accurate detection of objects, particularly small-scale or occluded ones, by leveraging relevant contextual cues. The semantic context fusion component (Section 5.2.2) integrates semantic information with visual context to capture object relationships. By combining prior knowledge and/or learning from the training dataset in the model training stage, this component enhances the detection network's understanding of the scene and improves its ability to discriminate and classify objects. The spatial context reasoning component (Section 5.2.3) introduces a general topological rela-

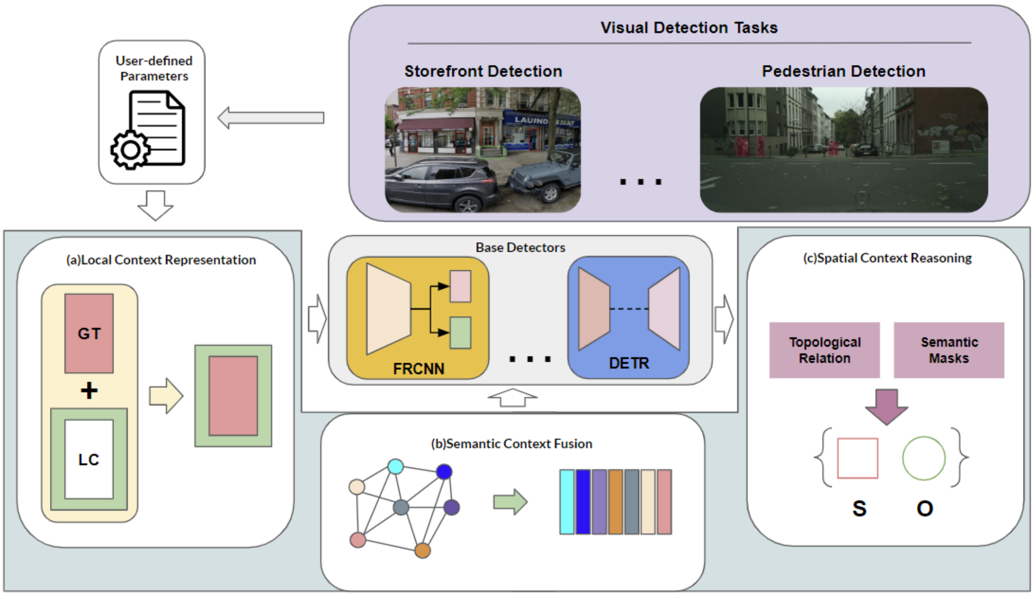


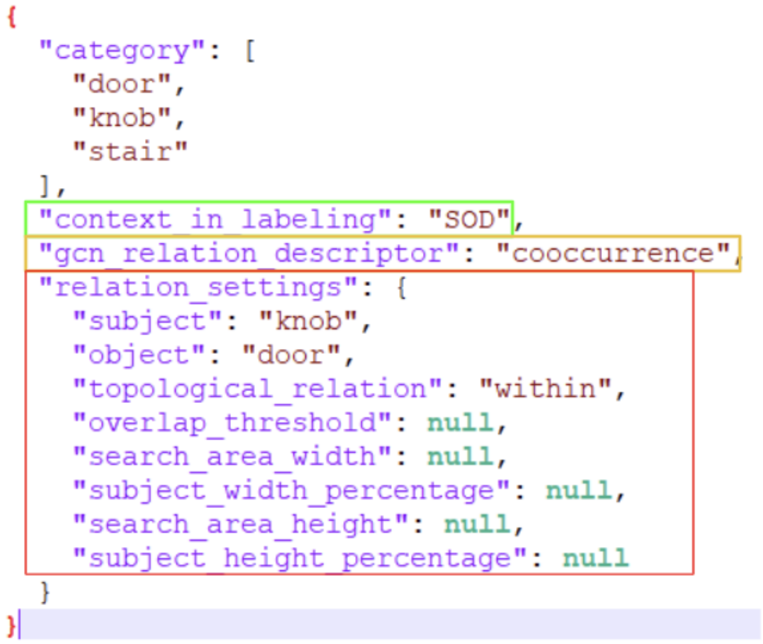
Figure 5.1: Details of our GMC framework, the general framework of multi-stage context learning and utilization for visual detection tasks. We design a user configuration mechanism for automating the process for various detection tasks (e.g., storefront object detection, pedestrian detection), using different base detectors (e.g. a CNN model Faster R-CNN (FRCNN) and a transformer model DETR. Three context learning and utilization components - (a) Local Context Representation, (b) Semantic Context Fusion, and (c) Spatial Context Reasoning, guide the deep learning models during data labeling, model training and post-processing stages. Each component can be applied individually and in combination. *GT*: Ground Truth. *LC*: Local Context. *S*: Subject. *O*: Object.

tion between object categories to optimize detection results. By considering the spatial relationships between objects in the post-processing stage, such as “above”, “under”, or “within”, this component refines detection outputs based on their spatial arrangements. This spatial context reasoning enhances the detection network’s localization accuracy and object classification performance by incorporating topological reasoning into the detection process.

We design a user configuration mechanism for automating the process for various detection tasks (e.g., storefront object detection, pedestrian detection), using different base detectors (e.g. a CNN model Faster R-CNN (FRCNN) and a transformer model DETR. In the following section, we will provide detailed explanations of each component within our proposed general framework. Through some user-defined parameters related to a given visual detection task and the chosen base detector, the GMC framework can be easily configured to form an end-to-end model for the task. The user-defined parameter configuration file is a straightforward JSON document, intentionally structured to cater to both experts and non-experts alike. Fig. 5.2 shows a example. Its simplicity ensures accessibility, allowing users of varying expertise levels to easily navigate and customize the parameters according to their specific needs. This thoughtful design aims to democratize the utilization of the configuration file, making it a user-friendly tool that empowers a diverse user base to tailor the parameters with ease and confidence.

## 5.2 Contextual Components of GMC

In the following, we will provide detailed explanations of each component within our proposed general framework. Through some user-defined parameters related to a given visual detection task and the chosen base detector, the GMC framework can be easily configured to form an end-to-end model for the task.



```
{
  "category": [
    "door",
    "knob",
    "stair"
  ],
  "context_in_labeling": "SOD",
  "gcn_relation_descriptor": "cooccurrence",
  "relation_settings": {
    "subject": "knob",
    "object": "door",
    "topological_relation": "within",
    "overlap_threshold": null,
    "search_area_width": null,
    "subject_width_percentage": null,
    "search_area_height": null,
    "subject_height_percentage": null
  }
}
```

Figure 5.2: A snapshot of a Jason file with user-defined parameters for context learning, including local context representation (LCR), semantic context fusion (SCF) and spatial context reasoning (SCR). Here green is for LCR, yellow is for SCF, and red is for SCR.

### 5.2.1 Local Context Representation

The concept of *local context* for objects, particularly small ones, takes center stage in the Local Contextual Representation (LCR) component. In the realm of computer vision, categorizing an object as "small" isn't always clear-cut. Factors like shooting angles and environmental conditions can render an object that's deemed "small", such as a spoon, appearing quite "large" within an image. Hence, the notion of smallness hinges on an object's size relative to the context of the image, as explained further below. The procedural essence is graphically illustrated in Figure 5.3. A local context calculator is at the heart of this process, guided by user-defined parameters specific to LCR.

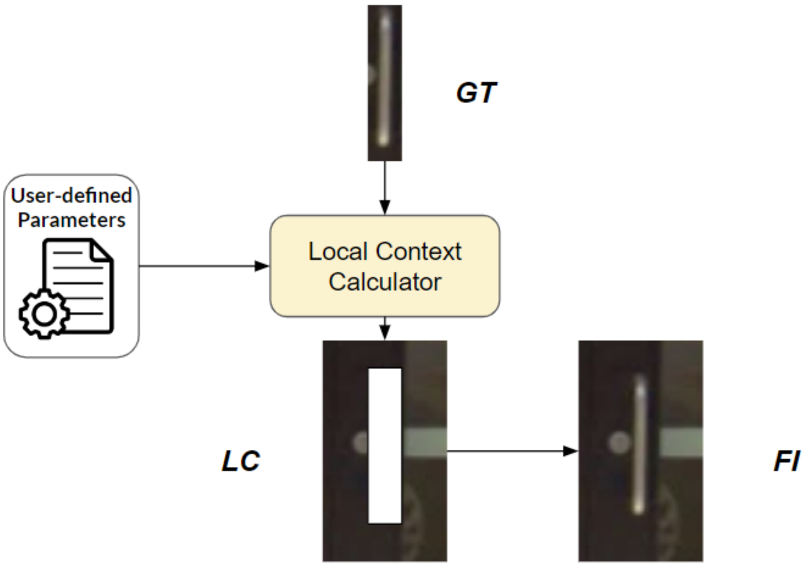


Figure 5.3: An utilized local context representation. The local context calculator is guided by user-defined parameters and enhance the local context around the ground truth label of the object. *GT*: Ground Truth. *LC*: Local Context. *FI*: Final Input.

This calculator works to enrich the local context surrounding the ground truth label of the targeted object. To initialize this local context calculator, we introduce two commonly embraced standards for characterizing small objects.

Within the COCO dataset [76], small objects are defined as those whose dimensions are $32 \times 32$ pixels or smaller, within the confines of an image with a fixed size of $640 \times 480$ pixels. Another definition, as detailed in [19], relates to situations where the overlap area between the ground truth bounding box and the image remains below 0.58%. Given the robustness and widespread adoption of these definitions in the research community, we employ them as reference points for automating the labeling process for small objects. We include the surrounding local context of the bounding box $B$ of an object $O$ in image $I$ if the object satisfies with the COCO standard for a small object as:

$$B'_O = \begin{cases} (1+\alpha)B_O, & \text{if } B_O < 32 \times 32 \\ B_O, & \text{otherwise} \end{cases} \tag{5.1}$$

If the small object satisfies with the second standard - the Small Object Dataset (SOD) Standard [19], we include the local context of the bounding box $B$ of the object $O$ in image $I$ by:

$$B'_O = \begin{cases} (1+\beta)B_O, & \text{if } \frac{B_O}{R_I} < 0.58\% \\ B_O, & \text{otherwise} \end{cases} \tag{5.2}$$

The above equations introduce notations representing the original and updated bounding boxes of the ground truth label for a small object. These notations, $B_O$ and $B'_O$ respectively, are utilized in the context of the user-defined parameters for the Local Context Representation (LCR) component. Firstly, the parameters $\alpha$ and $\beta$ hold significance as extending factors, expressed in terms of a percentage, from the original bounding boxes. These factors are related to two distinct standards: the COCO standard and the SOD standard. The resolution of the input image, denoted as $R_I$, is automatically determined. This automatically calculated resolution serves as a crucial component in the calculation of these factors. Secondly, the framework affords users the liberty to choose between the two contextual labeling standards. Should a given small object meet the criteria of both definitions, the user can opt for the standard that best aligns with their requirements.

Importantly, both the original bounding boxes and the enlarged bounding boxes are retained for all small objects that conform to the user-selected standard. This dual retention strategy serves the dual purpose of integrating local contextual information and enhancing the detection's robustness.

### 5.2.2 Semantic Context Fusion

Semantic information indeed plays a crucial role in visual detection tasks, providing valuable insights to enhance the detection process. To ensure a seamless and automatic Semantic Context Fusion (SCF) into our framework, we have introduced the SCF user-defined parameters, namely, the categories of a given visual detection task and the text embeddings used in the task. For example, for a storefront object detection task, they are door, doorknob, stair. For pedestrian detection, they include pedestrian, vehicle, bicycle (bike), motorcycle, etc. These parameters act as guiding factors for the model to learn and incorporate semantic context using text embeddings. The text embeddings, obtained from pre-trained language models, are utilized to generate semantic spaces that can be effectively fused with the visual information obtained from the detection process. This integration of semantic context with text embeddings allows our framework to automatically leverage valuable semantic information to improve the overall detection performance, while minimizing the need for extensive component modification.

In our framework, the fusion of semantic context is depicted in Figure 5.4. When the framework receives category information from the SCF user configuration, it proceeds to search for word embeddings $H_{labels} \in \mathbb{R}^{n \times d}$ from

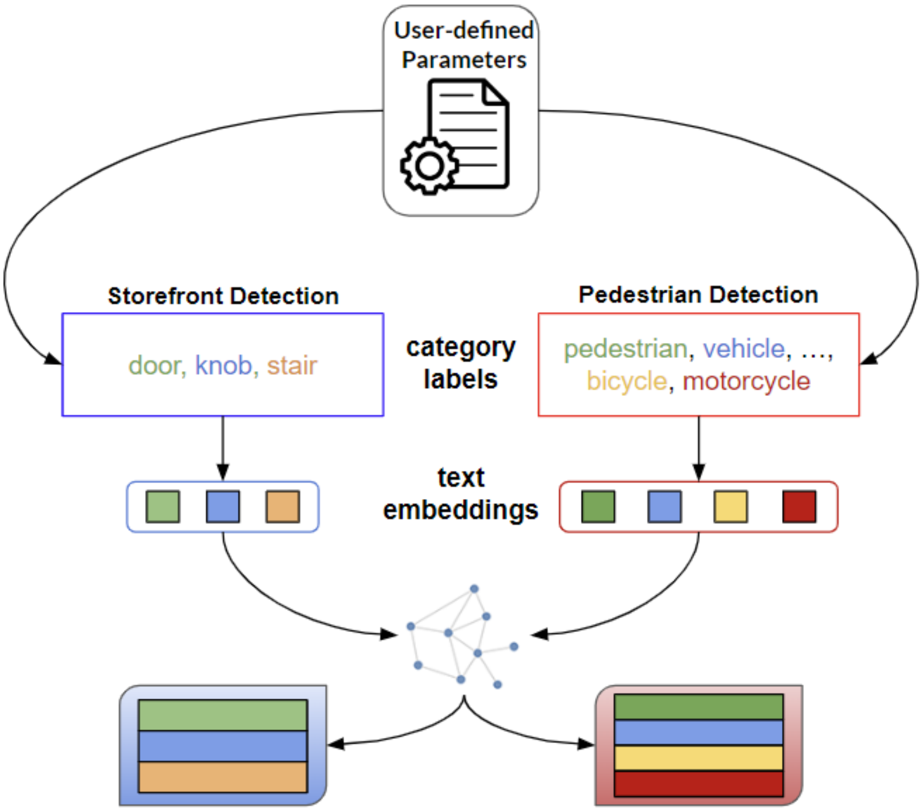


Figure 5.4: The visualization of Semantic Context Fusion. We use category information as the semantic context cues to generate semantic spaces for visual detection tasks.

a pretrained language model (such as GloVe [97]). Here, $n$ represents the number of label categories, and $d$ denotes the dimensionality of the word embeddings. Subsequently, an automatic generation of the contextual graph takes place. The Graph Convolutional Network (GCN) is then employed to learn semantic relations within the contextual graph, effectively constructing a semantic space. This semantic space is obtained by transforming the label feature representation, resulting in $H'_{labels} \in \mathbb{R}^{n \times D}$, where $D$ represents the dimensionality of the region features extracted from the object detector. As illustrated in Figure 5.1, the region features $f_{regions} \in \mathbb{R}^{D \times N}$ are projected into the semantic spaces $H'_{labels}$. Ultimately, the final output is derived from this process:

$$\mathbf{P}_{regions} = softmax(H'_{labels} f_{regions}) \tag{5.3}$$

where $\mathbf{P}_{regions}$ represents the classification probability distribution for each proposed region, and $\mathbf{P}_{regions} \in \mathbb{R}^{n \times N}$. More specific details of integrating with different base detector architectures will be discussed in Section 5.3.

### 5.2.3 Spatial Context Reasoning

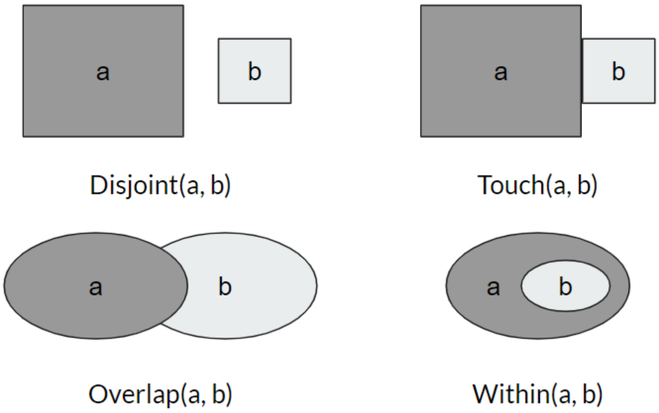


Figure 5.5: The visualization of the commonly used topological relationships from [30] and [39].

In the proposed general Spatial Context Reasoning (SCR) component, we leverage topological relationships to model the spatial relations between different objects. Topological relationships provide a general and abstract representation of the relationships between objects, such as *overlap*, *within*,

Table 5.1: Summary of the provided user-defined parameters for the spatial contextual reasoning component.

| Parameters | Context component | Definition |
|---|---|---|
| [***Subject, Object***] | LCR\SCF\SCR | Subject and object pair |
| ***Labeling_standard*** | LCR | The standard for small object label enlargement |
| ***Enlarge_percentage*** | LCR | The enlarging percentage for small object labels |
| ***Relation_descriptor*** | SCF | The contextual graph generation method |
| ***pred***(optional) | SCR | Directional relationships between subject and object |
| ***t*** | SCR | Topological relationships between subject and object |
| ***Overlap_threshold***(optional) | SCR | The threshold of overlap percentage between subject and object |
| $\boldsymbol{Search}_{\boldsymbol{height}}$(optional) | SCR | The height of search area for object |
| $\boldsymbol{Search}_{\boldsymbol{width}}$(optional) | SCR | The width of search area for object |

*touch*, and so on. These relationships capture the overall spatial configuration and arrangement of objects in a scene, including next two each other, within, and occlusion. The visualization of topological relationships is depicted in Fig. 5.5, illustrating how different objects can be related in terms of their spatial positions and co-occurrence. By incorporating topological reasoning, our framework enables a more comprehensive understanding of the spatial context, enhancing the object detection performance and facilitating richer semantic interpretations of the scene.

We utilize a predicate *pred*, such as *above*, *under*, etc., to describe the directional relation between a subject and object pair $[S, O]$, along with the topological relationship $t$, such as *overlap* and *within.* This general relation $R$ is defined as shown in Equation 5.4:

$$R[S, O] = pred[t(S, O)] \tag{5.4}$$

For instance, in urban settings, a common spatial relationship is that a stair is usually located under a door, even if there might be overlaps or spatial misalignment between them. The general relationship between a pedestrian and sidewalk can be described as

$$R[pedestrian, sidewalk] = under[overlap(pedestrian, sidewalk)] \tag{5.5}$$

It is important to note that the general spatial relation is inversible, meaning that a pedestrian is on the sidewalk, and sidewalk can be considered under a pedestrian. To effectively apply this spatial reasoning, we define a search area around the detected subject, and if an object is detected within this search area and satisfies the condition defined by Equation 5.5. We propose it as a detection and send it for evaluation. In cases where multiple objects are detected within the search area, we propose the object with the highest score as the final prediction.

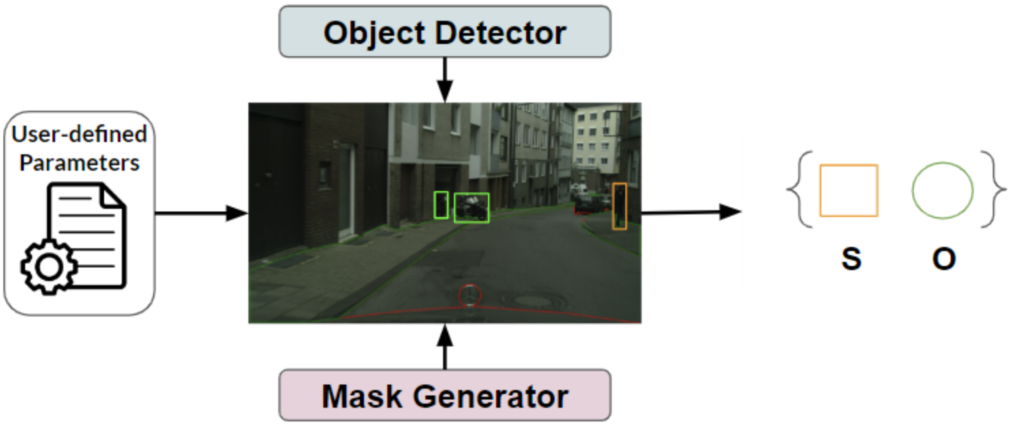


Figure 5.6: The visualization of general Spatial Context Reasoning.

To enhance the applicability of our general framework to diverse visual detection tasks, we have introduced *semantic masks* in our general spatial context reasoning component. This addition allows us to segment large stuff such as sidewalks and roads using a pretrained model, which could significantly improves spatial reasoning in larger scenes. To measure the overlap between subject-object pairs, we use the intersection over subject (IoS) metric to describe the general spatial relation, as defined as:

$$IoS = \frac{(A_s \cap A_o)}{(A_s)} \tag{5.6}$$

where $A_s$ and $A_o$ denote the area of the subject and area of the object. The area can be bounding box or semantic mask based on the specific scenarios. This formulation enables us to capture the relative spatial arrangement of objects in a scene, which is valuable for improving the accuracy of object detection and localization across various visual detection tasks. We also provide users with the flexibility to configure the general spatial relation for the categories in their own dataset, allowing them to adapt the framework according to their specific task requirements. The user-defined parameters for LCR, SCF and SCR components are summarized in Table 5.1.

## 5.3 Working with Various Network Architectures

The GMC framework can work with various deep learning network architectures with minimal modification of the code. In this paper, we give two examples, both which will be used in the tasks of our experiments. We employ two popular object detection models, Faster R-CNN [102] and DETR [14], as the underlying detectors for visual detection tasks, including storefront

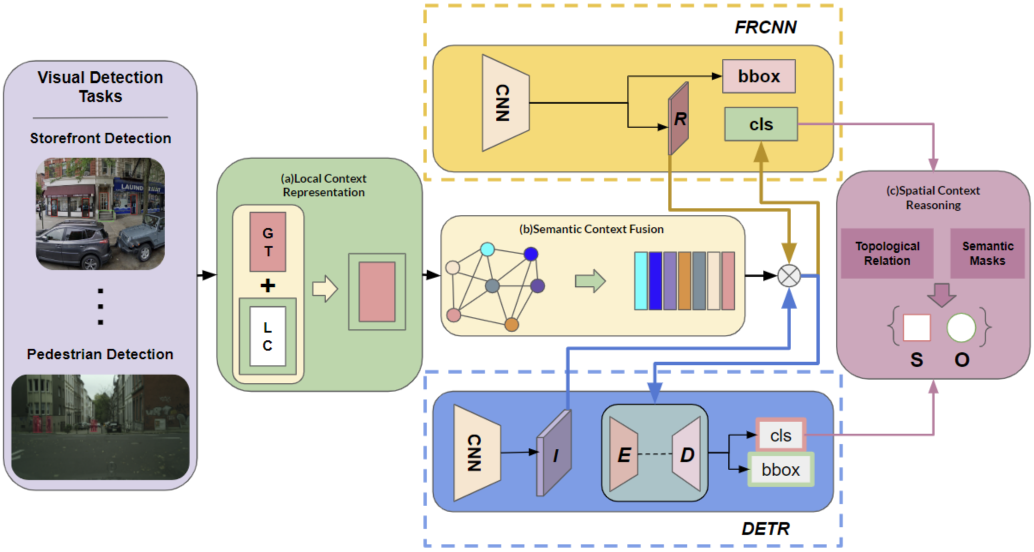


Figure 5.7: Integration of contextual components with different deep learning network architectures: Faster R-CNN (FRCNN) and DETR. $GT$: Ground Truth; $LC$: Local Context; $S$: Subject; $O$: Object; $R$: Region features; $I$: Image features; $E$: Encoder; $D$: Decoder; $bbox$: bounding boxes; $cls$: classification.

accessibility detection (Section 5.4) and pedestrian detection (Chapter 6.3). These two models have demonstrated strong performance in various object detection scenarios. The integration pipeline of the three context components with Faster R-CNN and DETR is shown in Fig.5.7. Other visual detection models can be integrated in a similar way. We will detail how the three context components can be seamlessly integrated with different backbone models, with minimal code modification, and then the experimental settings for the two models.

### 5.3.1 Integration Pipeline

Prior to the input of the visual detection task dataset into the model, we incorporate the Local Context Representation (LCR) component to augment the local context of specific objects. While we begin with two widely adopted definitions of small objects, as detailed in Section 5.2.1, we also empower users to tailor the enhancement of local context according to their preferences by adjusting the enlarge percentage. This integration ensures that the LCR component can seamlessly adapt to diverse models without requiring any modifications to the underlying backbone models. This design approach not only increases the generality of our framework but also facilitates its ease of use and customization across different applications.

Within our Semantic Context Fusion (SCF) component, we harmonize semantic knowledge with visual features prior to the detection process. This integration is illustrated in Fig. 5.7. In the case of Faster R-CNN, we achieve this by mapping the extracted region features (R) from the feature extractor backbone into the semantic space, before subsequently feeding the resulting output into the classification (cls) head. In contrast, for a comparative scenario of DETR in Fig. 5.7, we first project image features (I) into the semantic space and subsequently input the resulting output into a transformer encoder-decoder (E&D) for generating predictions. This design allows users to exercise control over the nature of the pretrained word embeddings in the SCF component, with the default setting being GloVe [97]. The SCF component can be seamlessly integrated into each backbone architecture with minimal adjustments, signifying its adaptability and ease of incorporation into diverse models. This enables the enriched representation of contextual information in conjunction with visual cues, thereby enhancing the overall detection accuracy.

Moreover, the Spatial Context Reasoning (SCR) component can be seamlessly integrated to fine-tune the detected candidates by synergizing topological relationships and semantic masks among identified objects. The SCR component provides a valuable post-processing feature for both Faster R-CNN and DETR models, requiring minimal architectural adjustments. This adaptable SCR component can be easily integrated into the final stage of object classification (cls), offering a streamlined way to enhance object detection performance. Users retain the prerogative to exercise control over the component's parameters within the configuration file, ensuring adaptability and customization to distinct detection scenarios. This feature bolsters the accuracy of detection outcomes by leveraging not only the object-specific information but also the relationships and arrangements among objects within the scene.

### 5.3.2 Experimental Settings

**Faster R-CNN.** In our implementation, we utilize ResNet-50 [50] as the backbone feature extractor along with the Feature Pyramid Network (FPN) [75], which are both pretrained on the COCO dataset. For the semantic context fusion, we employ a 2-layer graph convolutional network (GCN) with LeakyReLU [84] as the activation function. The GCN takes 300-dimensional word embeddings from GloVe [97] as the input label feature vector. During training, we employ Stochastic Gradient Descent (SGD) as the optimizer, with a momentum of 0.95 and a weight decay of 1e-4. The initial learning rate is set to 0.005 and is reduced by a factor of 0.25 every 8 epochs. We train the model for a total of 40 epochs for storefront accessibility detection and 60 epochs for pedestrian detection.

**DETR.** Following the methodology described in [14], we utilize ResNet-50 as the feature extractor and a transformer encoder-decoder for our visual detector. The learning rate for both ResNet-50 and the transformer encoder-decoder is set to 0.005, and a weight decay of 1e-4 is applied. To train the model effectively, we set the maximum number of training epochs to 120 for storefront accessibility detection and 200 for pedestrian detection. During the training process, we log the results every 5 epochs, allowing for detailed monitoring of the model's performance and progress. These settings ensure a comprehensive and robust training process for achieving accurate detection results.

To ensure a fair comparison, we fine-tuned all the pretrained backbone models on SAI, COCO and CityPersons/CityPersons+ Datasets.

## 5.4 Applying GMC to the SAI Dataset: Experimental Results

In order to assess the effectiveness of our proposed general framework, we conducted a thorough comparison with two baseline detectors - Faster R-CNN [102] and DETR[14], and two of our previous context learning approaches

[134, 136], using the SAI dataset. Here we use MultiCLU to represent the specially designed multi-stage context framework with the CNN-based model Faster R-CNN, as reported in [134], GMC-C to represent the GMC framework with the CNN-based model in this paper and also as reported in [134], and GMC-T to represent the GMC framework on the DETR-based model. To gauge the effectiveness of our approach on small objects within the SAI dataset, we adopted the same evaluation metrics outlined in [134]. Here, for the scenarios where the local context representation is employed, we leveraged both the original and expanded labels for small objects adhering to the defined criteria. We apply the same semantic information (Fig. 3.6) in our Semantic Context Fusion component as in MultiCLU. In cases where both labels were detected for the same small object, we considered just one to eliminate any possibility of duplicate detections. The configurations of the general spatial context reasoning (SCR) component for the SAI task are shown in Table 5.2, which are the same as in MultiCLU. The calculation of the overlap threshold (O_T) is integral to our methodology and is derived from the training dataset. The search areas for subsequent operations are then determined based on the statistical insights gained from this calculated overlap threshold. The evaluation primarily focused on two key performance metrics: mean average precision (mAP) and recall. These metrics were measured at a standard Intersection over Union (IoU) threshold of 0.5, which is commonly used in object detection tasks.

Table 5.2: Default user parameter settings for Spatial Context Reasoning in our experiments on the SAI Dataset[134]. O_T: Overlap_threshold.

| [Subject, Object] | Predicate | Topology | O_T | Search_area_height | Search_area_width |
|---|---|---|---|---|---|
| [door, knob] | - | within | - | - | - |
| [door, stair] | under | overlap | 0.2 | $0.2height_{door} + height_{stair}$ | $width_{door} + width_{stair}$ |

### 5.4.1 Performance Comparison on Faster R-CNN

Our comparative analysis revealed significant performance improvements when applying our framework to the CNN-based models (represented in rows 1 to 3 of Table 5.3). Note for the SAI dataset, the GMC-C results have been reported in [136], and the configuration is the same in this paper. Specifically, our GMC-C model outperformed Faster R-CNN, achieving substantial increases in both mAP (+13.6%) and recall (+15.3%). This highlights the effectiveness of our general context framework in enhancing object detection performance, surpassing the baseline detector. Furthermore, our GMC-C model exhibited a slightly higher mAP (+0.3%) compared to the special MultiCLU model, which employed specialized context mechanisms. However, there was a slight decrease in recall (-0.5%).

The comprehensive comparison outcomes demonstrate the compelling performance of our framework when integrated into CNN-based models. By incorporating various context learning and utilization components, our framework successfully enhances both mAP and recall, surpassing the performance of baseline detectors and previous context learning approaches. This reaffirms the potential and value of our general context framework in advancing the field of computer vision and object detection tasks.

Table 5.3: Comparison results on SAI dataset[134] with baseline detectors and previous context learning approaches.

| **Model** | **Precision ↑** | | | **Recall ↑** | | | **mAP ↑** | **Recall ↑** |
|---|---|---|---|---|---|---|---|---|
| | Door | Knob | Stair | Door | Knob | Stair | | |
| Faster R-CNN [102] | 75.6 | 17.7 | 66.0 | 87.5 | 47.6 | 73.1 | 53.1 | 69.4 |
| MultiCLU [134] | 78.0 | 51.2 | 70.0 | 92.3 | 80.4 | 83.0 | 66.4 | 85.2 |
| +LCR | 78.1 | 41.3 | 66.8 | 88.9 | 77.7 | 74.5 | 62.1 | 80.4 |
| +SCF | 78.0 | 19.0 | 68.5 | 90.1 | 53.0 | 79.4 | 55.2 | 74.2 |
| +SCR | 77.8 | 18.6 | 67.2 | 88.8 | 52.4 | 74.5 | 54.5 | 71.9 |
| +LCR+SCF | 78.4 | 50.0 | 69.2 | 90.8 | 75.0 | 79.4 | 65.9 | 81.7 |
| +SCF+SCR | 78.2 | 21.2 | 69.6 | 90.3 | 55.8 | 80.8 | 56.3 | 75.6 |
| +LCR+SCR | 79.2 | 41.2 | 67.8 | 89.2 | 77.8 | 74.5 | 62.7 | 80.5 |
| GMC-C [136] | 78.2 | 52.3 | 69.6 | 92.0 | 79.9 | 82.3 | 66.7 | 84.7 |
| DETR [14] | 75.9 | 23.8 | 69.2 | 91.8 | 58.4 | 77.8 | 56.3 | 76.0 |
| +LCR | 77.0 | 45.6 | 68.5 | 90.5 | 75.4 | 79.4 | 63.7 | 81.7 |
| +SCF | 77.8 | 27.6 | 70.0 | 91.4 | 61.5 | 81.2 | 58.5 | 78.0 |
| +SCR | 77.4 | 25.2 | 69.6 | 90.8 | 60.8 | 79.0 | 57.4 | 76.9 |
| +LCR+SCF | 80.2 | 55.1 | 71.2 | 92.7 | 81.2 | 82.3 | 68.8 | 85.4 |
| +SCF+SCR | 78.2 | 29.8 | 69.2 | 91.4 | 62.3 | 81.5 | 59.1 | 78.4 |
| +LCR+SCR | 78.8 | 50.8 | 69.2 | 92.0 | 77.8 | 80.4 | 66.3 | 83.4 |
| GMC-T | 80.6 | 55.8 | 71.2 | 92.7 | 82.0 | 82.6 | **69.2** | **85.8** |

### 5.4.2 Performance Comparison on DETR

To evaluate the flexibility and general applicability of our proposed framework, we extended its integration to the detection transformer architecture, represented by the DETR model [14]. By incorporating the context learning components into the detection transformer, we conducted a comprehensive analysis of its impact on the detection performance. The evaluation results (rows 4 to 5 in Table 5.3) demonstrated significant improvements of our GMC-T model in both mean average precision (mAP) and recall compared to the baseline transformer model (DETR). Specifically, we observed a noteworthy increase of 12.9% in mAP and 9.8% in recall, highlighting the effectiveness of our context learning components in enhancing detection performance within the transformer framework. These findings further emphasize the adaptability and efficacy of our proposed framework, as it consistently improves detection performance across different model architectures. Note here that the transformer-based model already has context information learnt within the model, this is probably why the improvement (from DETR to GMC-T) is not as high as that on the CNN-based models (from Faster R-CNN to GMC-C). Nevertheless, the GMC-T model, which incorporates our context learning components into the detection transformer, emerged as the top-performing model among the evaluated configurations. This outcome underscores the versatility and effectiveness of our framework in enhancing detection capabilities across diverse model architectures, showcasing its potential for various object detection tasks.

Our proposed framework demonstrates superior performance on the SAI dataset, exhibiting significant improvements over the baseline detectors and delivering competitive results compared to our previous specially-designed context learning model MultiCLU [134]. These findings support the efficacy of our general context framework in improving object detection accuracy and recall rates, meanwhile adapting to different visual detector architectures. By efficiently leveraging contextual information, our framework enhances object detection accuracy and recall rates, demonstrating its flexibility and effectiveness in various detection scenarios.

### 5.4.3 Performance Comparison with Different Context Components

We embarked on a comprehensive performance comparison across various combinations of our three contextual components. The outcomes, presented in Table 5.3, illuminate compelling insights.

First we analyze the performance improvements when using various combinations of contextual components on Faster R-CNN. When each contextual component was applied in isolation, notable enhancements in recall (from 2.8% to 11%) and mAP (from 1.4% to 9%) over the baseline were discernible. Furthermore, it's intriguing to observe that when deploying individual contextual components, the impact of local contextual labeling was more pronounced than that of the other two components.

Upon considering combinations of two contextual components, a noteworthy trend emerged, with each combination outperforming the baseline detector. The improvements ranged from +3.2% to 12.8% for mAP and from 6.2% to 12.3% for recall. Strikingly, when the combinations encompassed the Local Context Representation (LCR) component, they exhibited substantial superiority over other combinations, showcasing considerable gains in both mAP (+6.4% to 9.6%) and recall (+4.9% to 6.1%). This outcome underscores the value of incorporating contextual information around small objects, notably accentuating the detection efficacy of vital elements like doorknobs. Moreover, in relation to the single LCR component, both Semantic Context Fusion (SCF) and Spatial Context Reasoning (SCR) exhibited positive impacts. These components further improved results over a single LCR component, influencing both mAP and recall positively. Intriguingly, when contrasting the application of both SCF and SCR against their individual application, the combined utilization marginally enhanced both mAP and recall compared to using them in isolation.

The apex of our proposed framework's performance emerged with the integration of all three components (GMC-C), attaining a notable 13.6% improvement in mAP and an impressive 15.3% enhancement in recall over the baseline model Faster R-CNN. An interesting observation lies in the fact that our general framework enhances mAP across all categories in contrast to MultiCLU [134], albeit with only minimal reductions in recall. This suggests that the specifically designed MultiCLU might introduce more false positives than accurate predictions, positioning our framework to offer heightened precision at the cost of slightly reduced recall.

One notable distinction between the two base models lies in the impact of the Local Context Representation (LCR) component. Specifically, the improvements achieved by using LCR with DETR are not as substantial as those observed with Faster R-CNN. When solely applying the LCR component to Faster R-CNN, there is a remarkable enhancement in Precision and Recall for the "knob" category, with improvements of 23.6% and 30.1%, respectively. In contrast, when the LCR component is applied to DETR alone, the precision and recall see improvements of 21.8% and 17.0%, respectively, which are comparatively less effective than with Faster R-CNN. Moreover, the mAP and recall for Faster R-CNN see enhancements of 9.0% and 11.0%, whereas DETR experiences improvements of 7.4% and 5.7%, respectively, when the LCR component is added. This discrepancy could be attributed to the inherent self-attention mechanism of the transformer architecture, which inherently incorporates context information of local context especially for small objects, a feature that Faster R-CNN lacks. Nevertheless, the performance improvements achieved through various combinations of contextual components on DETR exhibit similar trends, indicating the consistent and robust functionality of the GMC framework across different backbone models.

## 5.5 Applying GMC to the COCO Dataset: Experimental Results

In order to check the scalability of our proposed general framework, we evaluate our framework on a large detection benchmark COCO dataset. The configurations of the SCR component are shown in Table 5.4. We conducted comparison with two baseline detectors - Faster ([102]) and DETR ([14]). We focus on two performance metrics: average precision (AP) and average precision for small objects ($AP_S$). The comparison results are shown in Table. 5.5.

**Performance comparison on Faster R-CNN[102].** Our comprehensive comparison results underscore the efficacy of our proposed GMC-C model, revealing significant improvements in key metrics. The average precision (AP) metric, a crucial indicator of overall detection performance, exhibited a notable enhancement of +0.7% when employing our framework compared to the baseline Faster R-CNN. Moreover, our model demonstrated

Table 5.4: Default user parameter settings for Spatial Context Reasoning in our experiments on the COCO Dataset[76]. O_T: Overlap_threshold.

| [Subject, Object] | Predicate | Topology | O_T | Search_area_height | Search_area_width |
|---|---|---|---|---|---|
| [person, person] | - | overlap | 0.73 | - | - |
| [person, surfboard] | under | overlap | 0.17 | $0.2height_{person}$ | $width_{surfboard}$ |
| [person, tie] | - | within | - | - | - |
| [person, skateboard] | under | overlap | 0.1 | $0.2height_{person}$ | $width_{skateboard}$ |
| [person, snowboard] | under | overlap | 0.16 | $0.2height_{person}$ | $width_{snowboard}$ |
| [zebra, zebra] | - | overlap | 0.83 | - | - |
| [baseball glove, person] | - | within | - | - | - |
| [potted plant, vase] | under | overlap | 0.45 | - | - |
| [frisbee, dog] | - | overlap | 0.85 | - | - |

a noteworthy advancement in AP for small objects, registering an improvement of +0.5%. This targeted improvement underscore the effectiveness of our proposed framework, particularly in addressing the detection challenges associated with smaller objects within the visual scene. The results substantiate the adaptability and enhanced performance of our GMC-C model, positioning it as a valuable asset in scenarios demanding precise and comprehensive object detection.

The application of the Local Context Representation (LCR) component in isolation on the Faster R-CNN model resulted in a modest improvement, with a 0.2% increase in average precision (AP) and a 0.3% enhancement in AP$_S$ (as detailed in Table 5.5). Remarkably, when the LCR component was synergistically combined with the Semantic Context Fusion (SCF) component, this pairing exhibited the most substantial improvement compared to other combinations. The joint application yielded a 0.5% boost in AP and a 0.4% increase in AP$_S$. It is noteworthy that the individual application of the SCF and Spatial Context Reasoning (SCR) modules had a comparatively minor impact on the COCO dataset. In summary, our holistic framework, encompassing all three components, demonstrated the most remarkable performance improvement across both AP (+0.7%) and AP$_S$ (+0.5%), surpassing the baseline detector and alternative component combinations.

Table 5.5: Comparison results on COCO dataset[76] with baseline detectors. IT: Inference Time (s).

| **Model** | **IT** | **AP ↑** | **AP$_S$ ↑** |
|---|---|---|---|
| Faster R-CNN [102] | 0.028 | 37.4 | 21.2 |
| +LCR | 0.028 | 37.6 | 21.5 |
| +SCF | 0.040 | 37.6 | 21.3 |
| +SCR | 0.030 | 37.5 | 21.2 |
| +LCR+SCF | 0.030 | 37.9 | 21.6 |
| +SCF+SCR | 0.040 | 37.8 | 21.4 |
| +LCR+SCR | 0.028 | 37.7 | 21.6 |
| GMC-C | 0.040 | 38.1 | 21.7 |
| DETR [14] | 0.036 | 42.0 | 21.0 |
| +SCF | 0.042 | 42.3 | 21.4 |
| +SCR | 0.037 | 42.2 | 21.2 |
| +SCF+SCR | 0.042 | 42.7 | 21.5 |

**Performance comparison on DETR [14].** In our evaluation using DETR, the impact of our context components becomes apparent when applied individually. Since we have to fine-tune the large DETR model if we evaluate LCR, we only tested performance improvements for the other two components (SCF and SCR) as the DETR can be frozen when training SCF and no re-training is needed for SCR. The Semantic Context Fusion (SCF) component, when introduced on its own, yields notable enhancements with a relative increase of +0.3% on AP and +0.4% on $AP_S$. This signifies that incorporating semantic relationships between objects contributes positively to the overall detection performance.

Conversely, the Spatial Context Reasoning (SCR) component, when applied independently, demonstrates a more modest impact, with only a +0.2% improvement on both AP and $AP_S$. This result is suggestive of the challenges associated with defining meaningful relations between objects in the COCO dataset, where the provided relations are limited.

Interestingly, the synergy between SCF and SCR components becomes evident when they are combined. Their complementary nature enhances each other's contributions, resulting in a more substantial improvement. The joint application of SCF and SCR leads to a further increase in performance, with a +0.7% improvement on AP and +0.5% on $AP_S$. This collaborative effect underscores the value of integrating both semantic and spatial context reasoning for more effective object detection within the DETR framework.

## 5.6 Summary

In this chapter, we design a general context learning and utilization framework to generalize our specifically designed MultiCLU for storefront accessibility detection to any visual detection tasks. Our proposed framework guided context learning from data labeling, contextual graph during training and general spatial reasoning during post processing. Our results show that our proposed framework can achieve same performance on the SAI dataset as the previous context learning framework, which is specifically designed for storefront accessibility detection. More importantly, we argue that the GMC framework can be applied to various visual detection tasks without the change of code. We further tested the framework on a large detection benchmark - MSCOCO dataset, showing promising results. Furthermore, the contextual components can be applied individually and in combinations, and easily add and remove from the object detector.

More importantly, we also applied our contextual components to the DETR, and the evaluation results show that our contextual components can improve the performance over the transformer-based architecture DETR, which is supposed to have a self-attention mechanism based on the image-part locations. Compared to its implicit uses of local and global context in the images, our contextual components provide explicit local context information, additional semantic context information and more explicit spatial context information, which may be not contained by the self-attention. Hence, our contextual components can improve over the transformer architecture.

Another interesting question is how well our GMC framework will perform on real-time detectors like YOLO and SAM (Segment Anything Model). We anticipate the GMC will be able to boost the performance of them as well; but we will leave this as a future work for researchers to explore.

In next chapter, we will provide the results when we apply the GMC framework to another dataset: the CityPersons dataset for pedestrian detection, with a more general spatial context reasoning component.

# Chapter 6

# Applying GMC to Pedestrian Detection with Enhanced Spatial Context Reasoning

In this chapter, we will discuss how our general multi-stage context learning and reasoning framework is used for pedestrian detection. We first define and discuss the problem of pedestrian detection (6.1) and then introduce the dataset we used (6.2) and a new dataset we and expanded. Furthermore, we provide details on the setup of the GMC framework for pedstrian detection in Section 6.3. The effectiveness of the general Spatial Context Reasoning (SCR) will be discussed in Section 6.4.

## 6.1 Problem Statement

Pedestrian detection in urban scenes presents unique challenges due to factors such as heavy occlusion and small-scale pedestrian images. Several papers have focused on addressing these challenges and improving the performance of pedestrian detection algorithms. For example, Cai et al. [11] proposed a unified framework for pedestrian detection that incorporates contextual information to handle occlusion. Zhang et al. [167] introduced the CityPersons dataset specifically for pedestrian detection in urban environments and proposed a scale-aware network to tackle the problem of detecting small-scale pedestrians.

Other works have explored different approaches to handle occlusion in pedestrian detection. Zhou et al. [171] proposed an attention-based method that focuses on visible parts of partially occluded pedestrians, improving the detection accuracy in challenging scenarios. Wu et al. [142] introduced a part-based detection framework that leverages feature transformation to handle occlusion and improve detection performance.

Despite the progress made by CNN-based pedestrian detectors, there are still limitations in detecting small-scale and heavily occluded pedestrians. These challenges require further exploration and innovation in the design of detection algorithms. For example, the integration of additional context information beyond a single image, such as global scene context and temporal context, could potentially improve the performance of pedestrian detection systems in real-world scenarios. This is beyond the scope of this chapter; some ideas can be found in Chapter 2, and further details can be found in our recent survey paper [138].

Pedestrian detection in urban scenes is a challenging task that has garnered significant attention in the computer vision community. Several papers have focused on addressing the unique challenges associated with detecting pedestrians in such environments. While approaches like Faster R-CNN have become popular for pedestrian detection, they often fall short in effectively handling heavily occluded pedestrians and small-scale pedestrians. Limited progress has been made in leveraging local context information specifically for these scenarios, resulting in sub-optimal detection performance.

To address this gap, our proposed M3C framework first integrates local context for small-scale and occluded pedestrian detection in urban scenes. Our approach also incorporates general topological relations among objects to facilitate spatial reasoning. By considering the relationships (including occlusions) between objects, we can reason about the presence and location of pedestrians, even in challenging situations. Notably, our framework goes beyond improving pedestrian detection alone; it also enhances the detection results for other objects in the scene. By leveraging the synergistic effects of contextual components, our approach aims to achieve superior performance compared to existing methods.

By emphasizing the importance of local context and introducing general topological reasoning, our framework offers a comprehensive solution for pedestrian detection in urban scenes. Note that the general framework is not specially designed for pedestrian detection but the system can be configured to tackle these two challenges in pedestrian detection. Through the incorporation of contextual cues and the utilization of interplay between different components, we can overcome the limitations of traditional approaches and improve detection accuracy. Ultimately, our work contributes to advancing the understanding of urban scenes and objects, opening up new possibilities for real-world applications.

## 6.2 CityPersons and CityPersons+ Pedestrian Datasets

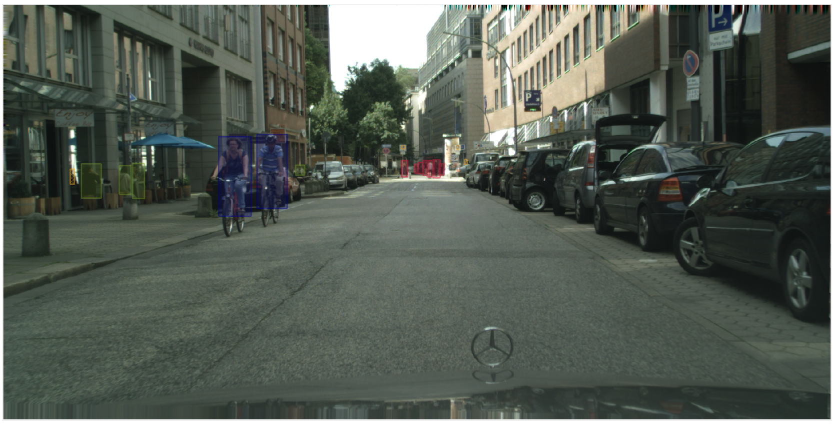


Figure 6.1: The label example from CityPersons Dataset [167]. Red: Pedestrian. Blue: Rider. Yellow: Sitting person.

The CityPersons dataset is derived from the Cityscapes dataset [31], focusing specifically on person annotations. It contains annotations for four categories: *pedestrian, rider, sitting person*, and *person (other)*. Table 6.1 provides an overview of the dataset, including information on the number of images and annotations for each category. Figure 6.1 showcases an example of labeled pedestrians from the dataset, providing a visual representation of the annotated data.

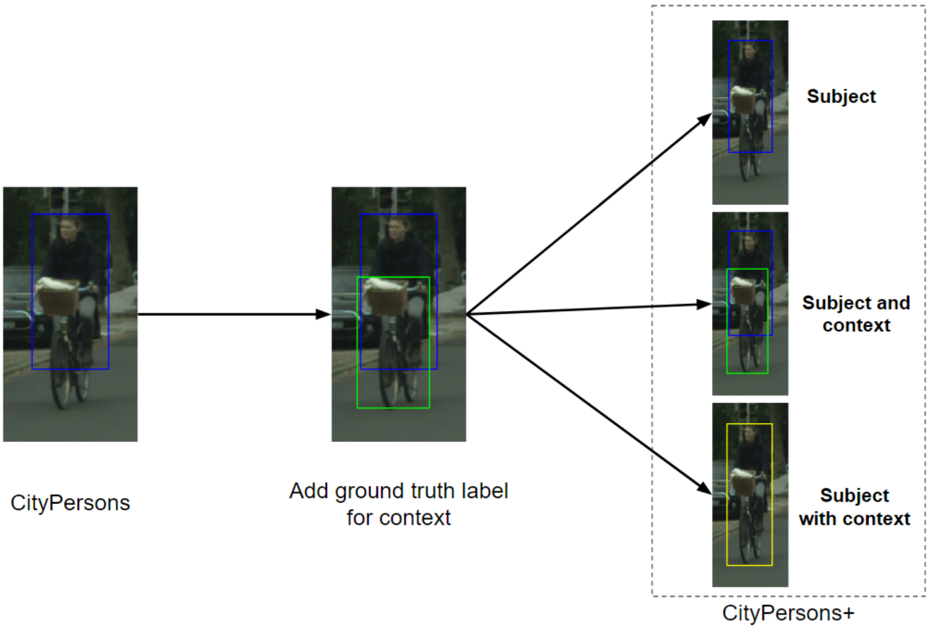


Figure 6.2: The demonstration of riders in CityPersons+ dataset. We extend existing categories in CityPersons dataset, with context information, by adding the ground truth label for context things and combined with the existing subject class label.

Table 6.1: Statistics of CityPersons and CityPersons+ Datasets.

| Dataset | # of Category | # of Training | # of Validation |
|---|---|---|---|
| CityPersons [167] | 4 | 2975 | 500 |
| CityPersons+ | 6 | 2975 | 500 |

To incorporate various context information and leverage the general topological relations between different categories, we introduce the CityPersons+ dataset. This dataset expands upon the CityPersons dataset by incorporating additional object labels from the Cityscapes dataset, including more specific subcategories. Specifically, we categorize pedestrians and riders into four subcategories: *pedestrian on road, pedestrian on sidewalk, rider with motorcycle*, and *rider with bicycle*. Therefore CityPersons+ contains annotations for six categories. The purpose of adding subcategories is to better utilizing context information. Fig. 6.2 shows how we include more context information without changing existing labels. Not only we include the original labels, we also include the context label and the subject-context combined label in the CityPersons+ dataset. We also relate the six categories in CityPersons+ dataset to context information that are beyond these six categories. First, we add the bounding box ground truth labels for *context things*, including motorcycles, bicycles and vehicles, which are related to the existing subject class labels of rider with motorcycle, rider with bicycle, and pedestrian occluded by vehicle, respectively. Second, we include the the semantic segmentation labels of *context stuff*, such as roads and sidewalks, which could provide precise spatial reasoning between different objects, namely, pedestrian on road, and pedestrian on sidewalk, in addition to pedestrian occluded by pedestrian. We also include word embeddings for both context things (motorcycles, bicycles and vehicles) and context stuff (roads and sidewalks) for Semantic Context Fusion (SCF) component. We use the pretrained model weights for Faster R-CNN and DETR to *detect* the context things, and Segformer [144] to *segment* the semantic masks for context stuff, to facilitate general topological reasoning within the Spatial Contextual Reasoning (SCR) component. Table 6.1 provides an overview of the statistics for the CityPersons+ dataset, comparing with CityPersons dataset: we double the class categories for pedestrian and riders (from 2 to 4), add 5 context objects (not shown in the Table), without changing the existing classes (2). For the 4 basic classes in CityPersons and 6 basic classes in CityPersons+, as shown in Table 6.1, the pretrained model weights for Faster R-CNN and DETR are finetuned using the two datasets, respectively, and the proposed GMC models will be evaluated.

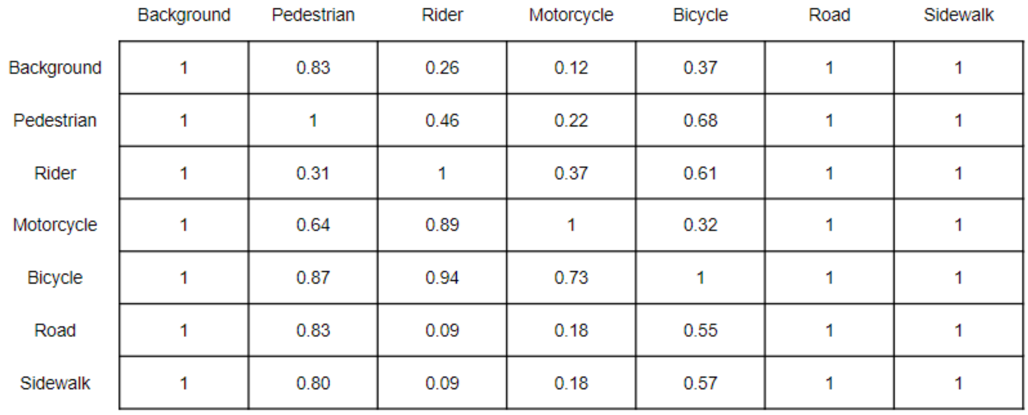
## 6.3 Applying GMC to CityPersons

We conducted further evaluation of our general context learning and reasoning framework on pedestrian detection task using CityPersons dataset, comparing it with the baseline detectors, Faster R-CNN [102] and DETR [14], without any code modifications. Here again, we use GMC-C to represent the general framework of context learning with the CNN-based model, and GMC-T to represent the general framework on DETR-based model, on the original CityPersons dataset (without considering the subcategories or additional context for spatial context reasoning). In summary, in the labeling stage, we employ the small object standard for the CityPersons dataset to enhance the labeling of small objects with local context labeling. The enlarge percentage is set to 10 percent ($\beta$=0.10). We further leverage the fine-grained category rider using GloVe [97] word embeddings in CityPersons dataset to enable the semantic context fusion in the training stage. Then we leverage the spatial context for rider in the postprocessing stage, which are shown in Table 6.2 (First 2 rows for the rider). Note that the GMC-C model in this paper is the same as that in [136].

Further, we use GMC-C+ and GMC-T+ to represent the general framework with further semantic context fusion and spatial context reasoning, using the CityPersons+ dataset with subcategories of pedestrians and riders, as well as information of vehicle, road and sidewalk. The local context representation is the same as for CityPersons. The semantic context fusion uses the object co-occurrence as shown in Fig. 6.3. The configurations of the general spatial context reasoning (SCR) component for the CityPersons+ task are shown in Table 6.2. The computation of the overlap threshold (O_T) is a crucial aspect of our methodology, originating from insights gained during the training dataset analysis. Subsequently, the determination of search areas relies on the statistical characteristics encapsulated by this calculated overlap threshold. Notably, for categories associated with pedestrians, our approach employs only the overlap threshold (O_T) without explicitly defining a search area. This nuanced strategy reflects the adaptability of our model, tailoring its behavior based on the specific requirements and characteristics of different object categories. We compared the evaluation results on the *reasonable* and *heavy* subsets of the data using the standard evaluation metric in pedestrian detection, $MR^{-2}$ (where lower values indicate better performance). Here, the subsets were defined based on the height ($h$) and visible ratio ($v$) of pedestrians: Reasonable subset: $h \in [50, \infty]$, $v \in [0.65, 1]$; Heavy subset: $h \in [50, \infty]$, $v \in [0, 0.65]$.

Table 6.2: Default user parameter settings for Spatial Context Reasoning in our experiments on the CityPersons+ Dataset. O_T: Overlap_threshold.

| [Subject, Object] | Occlusion | Predicate | Topology | O_T | Search_area_height | Search_area_width |
|---|---|---|---|---|---|---|
| [rider, bicycle] | Reasonable | under | overlap | 0.48 | $0.5height_{rider}$ | $width_{bicycle}$ |
| [rider, motorcycle] | Reasonable | under | overlap | 0.5 | $0.5height_{rider}$ | $width_{motocycle}$ |
| [pedestrian, vehicle] | Heavy | under | overlap | 0.68 | - | - |
| [pedestrian, pedestrian] | Heavy | - | overlap | 0.76 | - | - |
| [pedestrian, road] | Reasonable | under | overlap | 0.2 | - | - |
| [pedestrian, sidewalk] | Reasonable | under | overlap | 0.13 | - | - |



| | Background | Pedestrian | Rider | Motorcycle | Bicycle | Road | Sidewalk |
|---|---|---|---|---|---|---|---|
| Background | 1 | 0.83 | 0.26 | 0.12 | 0.37 | 1 | 1 |
| Pedestrian | 1 | 1 | 0.46 | 0.22 | 0.68 | 1 | 1 |
| Rider | 1 | 0.31 | 1 | 0.37 | 0.61 | 1 | 1 |
| Motorcycle | 1 | 0.64 | 0.89 | 1 | 0.32 | 1 | 1 |
| Bicycle | 1 | 0.87 | 0.94 | 0.73 | 1 | 1 | 1 |
| Road | 1 | 0.83 | 0.09 | 0.18 | 0.55 | 1 | 1 |
| Sidewalk | 1 | 0.80 | 0.09 | 0.18 | 0.57 | 1 | 1 |

Figure 6.3: Relation descriptor matrix generated from the CityPersons+ training dataset.

### 6.3.1 Overall Comparison with Baseline Detectors

The comparison results presented in Table 6.3 provide insights into the performance of the GMC framework on different architectures on both the reasonable and heavy subsets. It is observed that DETR and transformer-based GMC model (GMC-T) generally exhibits superior performance on the reasonable subset (+1.6% and +2.9%, respectively, compared to the Faster-RCNN base model), indicating its effectiveness in capturing contextual information and enhancing detection accuracy. However, DETR and GMC-T demonstrates lower performance on the heavy subset (-2.6% and -3.9% respectively, compared to the Faster-RCNN base model), which could be attributed to the absence of design elements such as the feature pyramid network (FPN) [75] employed in the Faster R-CNN framework. In contrast, the CNN-based model GMC-C may not achieve the same level of performance on the reasonable subset as transformer-based model GMC-T, but it often demonstrates better performance on the heavy subset (+1.7% compared to the Faster-

Table 6.3: Comparison results on Citypersons dataset[167] with baseline detectors and previous context learning approaches.

| Model | Reasonable ↓ | Heavy ↓ |
|---|---|---|
| Faster R-CNN[102] | 13.4 | 36.9 |
| +LCR | 12.3 | 35.6 |
| +SCF | 13.3 | 37.1 |
| +SCR | 13.0 | 36.5 |
| +LCR+SCF | 12.2 | 35.2 |
| +SCF+SCR | 13.2 | 36.5 |
| +LCR+SCR | 12.0 | 36.0 |
| GMC-C [136] | 12.0 | **35.2** |
| DETR [14] | 11.8 | 40.8 |
| GMC-T | **10.5** | 39.5 |

RCNN base model). This suggests that the CNN-based model are able to effectively handle challenging scenarios with heavily occluded pedestrians, where precise localization and robust feature extraction are crucial. This evidence supports our rationale of the general context framework in working with various backbone models depending on the task requirements.

### 6.3.2 Performance Comparison with Different Context Components on Faster-RCNN

Upon applying the Local Context Representation (LCR) component alone on Faster R-CNN, there was a noticeable enhancement of 1.1% on the reasonable subset and 1.3% on the heavy subset (as illustrated in Table 6.3). To further amplify our framework's capabilities, we introduced a fine-grained category (rider) into the CityPersons dataset during training to facilitate the Semantic Context Fusion (SCF) and Spatial Context Reasoning (SCR) components. As observed in the results analogous to those from the SAI dataset, configurations with the LCR component consistently yielded superior performance compared to other settings. However, it's worth noting that both SCF and SCR modules had a minor impact on pedestrian detection, possibly attributed to the relatively weak correlation between pedestrians and other urban objects. In summation, our comprehensive framework, encompassing all three components, achieved the most impressive performance across both the reasonable subset (1.4% lower) and the heavy subset (1.7% lower), outperforming the baseline detector and alternative combinations.

### 6.3.3 Comparison with DETR

Upon comparing our newly introduced GMC-T model with the baseline Detection Transformer (DETR) model, our GMC-T model consistently demonstrated superior performance across both the "reasonable" and "heavy" subsets. This was marked by a substantial enhancement in detection performance, exhibiting an impressive 1.3% improvement on both subsets. These results provide compelling evidence for the effectiveness of our context learning and reasoning components in bolstering the detection capabilities of diverse architectural frameworks. Moreover, our framework's adaptability is evident as it showcases its prowess not only in CNN-based models but also in transformer-based models. The ease with which our framework can be integrated and customized underscores its potential to cater to a range of visual detection tasks beyond just pedestrian detection.

Overall, the comparison results highlight the potential and versatility of our proposed context learning and reasoning components in improving object detection performance across different datasets and tasks. The framework offers a flexible and effective solution for incorporating context information and enhancing the detection capabilities of various deep learning models, contributing to advancements in the field of computer vision and object detection.

## 6.4 Effectiveness of the Enhanced General SCR

Table 6.4: Comparison results on the enhanced general spatial context reasoning (SCR) component with baseline detectors and previous designed component.

| Model | Reasonable ↓ | Heavy ↓ |
|---|---|---|
| Faster R-CNN[102] | 13.4 | 36.9 |
| Faster R-CNN + SCR | 12.8 | 36.1 |
| GMC-C [136] &(this paper) | 12.0 | 35.2 |
| GMC-C+ (this paper) | 11.8 | **34.8** |
| DETR [14] | 11.8 | 40.8 |
| DETR + SCR | 11.2 | 39.8 |
| GMC-T (this paper) | 10.5 | 39.5 |
| GMC-T+ (this paper) | **10.2** | 38.6 |

We also conducted an extensive study to evaluate the effectiveness of the enhanced general spatial context reasoning (SCR) component within our framework. In order to achieve a more comprehensive and robust topological reasoning, we leveraged both bounding boxes for objects (such as bicycles, motorcycles, cars, pedestrians) and semantic masks for stuff (such as sidewalks and roads) in CityPersons+ dataset. This allowed us to capture and utilize the spatial relationships between various entities in the scene. To assess the impact of the enhanced general SCR component, we evaluated its performance in two enhanced models - GMC-C+ and GMC-T+, as well as its use on the two baseline object detection models - Faster R-CNN and DETR. Table 6.4 presents the comparative results of these models with and without the SCR component.

### 6.4.1 SCR Performance on Faster R-CNN

When we solely applied the SCR component to the Faster R-CNN model, we observed notable improvements in performance for both the reasonable and heavy subsets, achieving an increase of 0.6% and 0.8%, respectively. However, it is important to note that the Faster R-CNN model, without the inclusion of the local context and semantic context components, did not achieve the same level of performance as the GMC-C model. By replacing the initial spatial context reasoning component with our enhanced SCR component in the GMC-C model, leading to the GMC-C+ model, we observed a slight performance improvement of 0.2% on the reasonable subset and 0.4% on the heavy subset, over the GMC-C model. These results indicate that the integration of the enhanced SCR component can enhance the performance of the GMC-C model to some extent. However, when comparing these results with the performance of the enhanced SCR component alone (i.e., Faster R-CNN + SCR), it is evident that the GMC-C+ model with the combined local context, semantic context, and enhanced SCR component outperformed both subsets, achieving a significant improvement of 1.0% on the reasonable subset and 1.3% on the heavy subset. This demonstrates the synergistic effect of incorporating multiple context sources within the framework. our evaluation confirms that the integration of the enhanced general SCR component can effectively improve the performance of object detection models, particularly when combined with the local context and semantic context components. Overall, GMC-C+ achieves performance improvements of 1.6% on the reasonable and 2.1% on the heavy, compared to the Faster-RCNN base model.

### 6.4.2 SCR performance on DETR

We also study whether our enhanced general SCR component can improve over the DETR model, which already incorporates a self-attention mechanism to leverage context information. Not surprisingly, even with the existing self-attention mechanism, the application of the enhanced SCR component to the DETR model led to performance improvements. Specifically, we observed an increase of 0.6% on the reasonable subset and 1.0% on the heavy subset, indicating that the SCR component can effectively enhance the context utilization capabilities of the DETR model. Furthermore, when we combined the general SCR component with the other two contextual components (local context and semantic context), our GMC-T+ model achieved additional performance improvements over the DETR model and the GMC-T model on both evaluation subsets. The results showed a significant improvement of 1.6% on the reasonable subset and 2.2% on the heavy subset, compared to the DETR base model, and a visible improvement of 0.3% on the reasonable subset and 0.9% on the heavy subset, compared to the GMC-T model. This highlights the complementary nature of the contextual components and their ability to further enhance the detection performance of the DETR model.

### 6.4.3 Further Discussion

Our evaluation on pedestrian detection task confirms that the integration of the more general SCR component can effectively improve the performance of the detection models, particularly when combined with the local context and semantic context components. Our three contextual components, when integrated with the DETR model, demonstrated the best performance on the reasonable subset. On the other hand, the three contextual components combined with the CNN-based model Faster R-CNN exhibited better performance on the heavy subset. These findings indicate that the choice of model architectures, in combination with the specific context components, can have an impact on the overall detection performance, with different configurations achieving better results on different evaluation subsets. This also highlights the importance of leveraging multiple context sources and considering the spatial relationships between objects for achieving more accurate and robust detection.

## 6.5 Summary

In this chapter, we apply our GMC framework for pedestrian detection task. We first define the problem and then introduce the CityPersons and CityPersons+ datasets. Overall, the comparison results highlight the potential and versatility of our proposed context learning and reasoning components in improving object detection performance across different datasets and tasks.

The framework offers a flexible and effective solution for incorporating context information and enhancing the detection capabilities of various deep learning models, contributing to advancements in the field of computer vision and object detection. By working on different datasets, we demonstrate that our general framework can be applied to other detection tasks with minimum code modification. For exploring the full potentials of the GMC framework in data collection and model training, we can further investigate ways in other visual tasks, such as semantic segmentation, using the GMC detection engines. The next chapter will be on extending the GMC framework to panoptic segmentation, an advanced form of semantic segmentation.

# Chapter 7

# Extending the GMC Framework to Panoptic Segmentation

In this chapter, we extend the GMC framwork from visual object detection tasks to semantic segmentation. We particularly look into panoptic segmentation which has emerged as a critical visual task. We show that our GMC framework can be easily adapted to panotic segmentation, with new backbone network architectures. We will particularly use a state-of-the-art transformer-based model called Oneformer [55] as our backbone model, working on the **Cityscapes** dataset [31]. Since the model is big and has been well-trained, we will only test the Semantic Context Fusion (SCF) component and the Spatial context Reasoning (SCR) component. The reason of not testing the Local Context Representation (LCR) component is that we will have to re-trained the backbone model with the LCR component. For completion, we may repeat what we have described in the GMC framework, but the discussions will be in the context of panoptic segmentation.

## 7.1 Introduction

In the dynamic landscape of computer vision research, the segmentation of images has been a focal point, encompassing various tasks such as semantic segmentation and instance segmentation. Semantic segmentation involves the assignment of labels to objects and elements within an image, where entities of the same class are identified by the same label (e.g. using a color for visualization). On the other hand, instance segmentation refines this process by assigning distinct labels/colors to individual entities, effectively excluding background elements, or "stuff", from the segmentation process. The confluence of these two segmentation paradigms has given rise to panoptic segmentation (Fig. 7.1), a task that not only unifies semantic and instance segmentation but also extends their objectives [63]. The term "panoptic" in panoptic segmentation is emblematic of its ambitious aim to provide a comprehensive and all-encompassing view of a visual scene. This is achieved by partitioning images into regions rich with semantic information, allowing for the differentiation between discrete objects and the more amorphous contextual elements, often referred to as "stuff". The nuance brought about by this approach enables a deeper comprehension of a scene, distinguishing between well-defined entities like people or cars and the less delineated environmental elements such as the sky or road. As machines strive to interact intelligently with the visual world, panoptic segmentation has emerged as a critical task. Our work is primarily centered on advancing the field of panoptic segmentation.

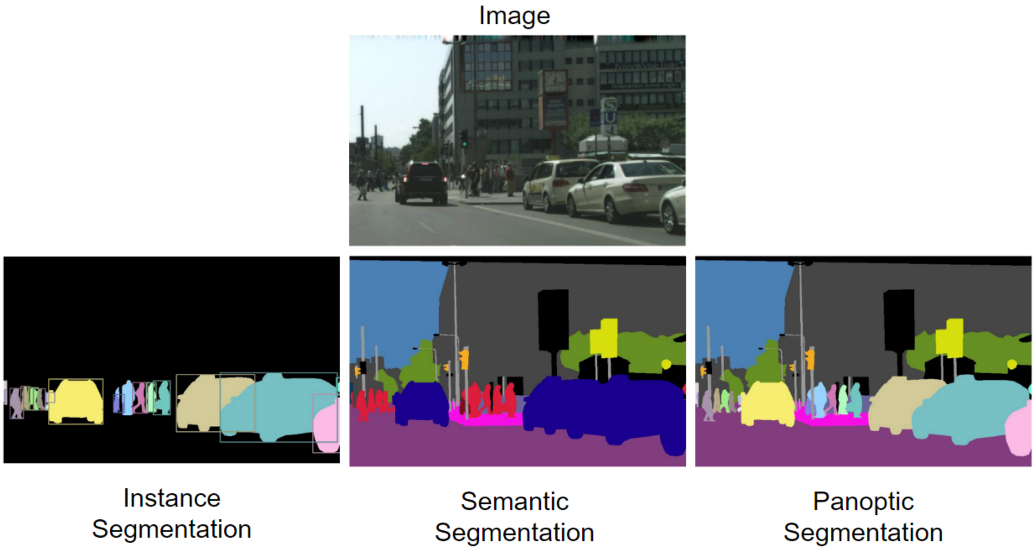


Figure 7.1: Segmentation Results from [63] illustrating instance segmentation, semantic segmentation and panoptic segmentation.

The complexity of panoptic segmentation lies in its dual objective of simultaneously performing semantic and instance segmentation, setting it apart from the conventional approaches of tackling either semantic or instance segmentation in isolation. Earlier methodologies [63] proposed a modular approach, employing distinct modules for each task. For instance segmentation, a Mask-RCNN module was utilized, while a Fully Convolutional Network (FCN)-based module addressed semantic segmentation. Post-processing steps were then employed to fuse these outputs and generate the panoptic segmentation. Recent endeavors [27, 168, 72, 55] have aimed at devising universal architectures capable of handling all segmentation tasks within a singular framework. These architectures, while offering a comprehensive solution, often come with a downside of a substantially huge number of parameters therefore a increased training time. In this research, we build upon a unified architecture with a highly lightweight implementation for semantic and spatial reasoning. By minimizing the number of parameters and reducing training time, our methodology seeks to strike a balance between computational efficiency and achieving the promising performance.

The integration of contextual information has become a cornerstone in various computer vision tasks, where context encompasses any information relevant to the visual attributes of a target, be it an object or an event. This contextual information can take the form of visual or non-visual cues. In the realm of object recognition, singling out a particular object might pose challenges when it exists outside its contextual environment. Here, contextual information becomes invaluable, offering vital clues for accurate target recognition. The significance of context features extends to image segmentation as well, as demonstrated by notable works such as [20, 22, 169]. Despite the recognized importance of context, recent endeavors have often fallen short in harnessing context information effectively. In response to this gap, our work first introduces a context component designed to leverage *semantic* context information for the panoptic segmentation task. By incorporating this semantic context module, we aim to enhance the performance of panoptic segmentation through a more effective integration of semantic contextual cues, contributing to improved scene understanding and segmentation accuracy.

Objects rarely exist in isolation; they share spatial relationships in real-world scenes. While *spatial* context has found applications in various vision tasks like object detection [153, 134, 156] and scene graph generation

[18], its utilization in panoptic segmentation has been relatively limited. Acknowledging the inherent spatial connections between objects, we posit that incorporating spatial context reasoning can significantly enhance the performance of panoptic segmentation. Topological relationships, defined by the arrangement and connectivity of objects, offer a versatile means of modeling interactions between different objects. Notably, this approach is applicable to both bounding boxes and semantic masks, showcasing its flexibility. In our work, we leverage topological relationships as a guiding principle for post-processing predictions. This methodology holds the promise of refining panoptic segmentation results by effectively capturing the nuanced spatial interplay between diverse entities in a scene, without the re-training of the underlying deep learning models.

In general, we propose a lightweight context reasoning framework for panoptic segmentation, by employing semantic context during training and spatial context for post-processing. We summarize the contributions of our work as follows:

- We propose a lightweight context learning and reasoning framework that seamlessly integrates with a baseline model for the problem of panoptic segmentation.
- We employed different context information (semantic context and spatial context) during training and post-processing stage, to leverage the features and final panoptic predictions.
- Our proposed method shows promising results on the Cityscapes dataset, at the same time does not need to fine-tune the baseline model which

usually has a huge number of parameters, thus achieving comparable performance with high computation efficiency.

The rest of the chapter is organized as follows. Section 7.2 discusses related work. Section 7.3 proposes our context learning and reasoning framework for panotic segmentation and describes each component in detail. Section 7.4 presents our experiments, including experimental settings, dataset description, experimental results and the ablation studies of our framework. Section 7.5 provides a few concluding remarks.

## 7.2 Related Work

### 7.2.1 Panoptic Segmentation

Panoptic segmentation, a task at the forefront of computer vision, seeks to holistically interpret visual scenes by addressing both instance and semantic segmentation. In its early stages [63], the approach involved utilizing predictions from separate models dedicated to instance and semantic segmentation. However, this method's inefficiency became apparent due to a lack of parameter sharing between the two models. The evolution of panoptic segmentation methodologies has led to their classification into separate presentations and unified architectures. In separate presentations, instances and semantic classes are segmented by a single model but through different branches. Instances may be segmented using methods based on bounding boxes [23, 70, 157] or box-free techniques [155], while stuff is typically segmented using a fully convolutional branch. Prominent methods using separate representations include AUNet [71], Panoptic FPN [62], and UPSNet [147]. On the other hand, unified approaches segment both things and stuff based on features generated from shared layers [72, 69]. A recent work called Oneformer [55] proposesa universal image segmentation framework that unifies segmentation with a multi-task train-once design. The framework use a task-conditioned joint training strategy that enables training on ground truths of each domain. Despite the advancements made by these unified approaches, a critical consideration remains: the limited integration of contextual and relational information between things and stuff during feature generation and prediction. Addressing this gap is essential for advancing panoptic segmentation methodologies and fully unlocking their potential in understanding and interpreting diverse visual scenes. Our work employs both semantic context and spatial context to enhance the features and final predictions.

### 7.2.2 Context Learning in Semantic Segmentation

In semantic segmentation tasks, context information has been explored through methods like ParseNet [79], Context Encoding Forest [164], and Pyramid Scene Parsing Network [169], which leverage global context information or incorporate context embeddings to improve the segmentation accuracy. Despite the success in incorporating global context information, there are also coherent composition of objects, which haven't been explored much in segmentation tasks. The presence of one object can be compelling evidence of the existence of the other. 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 objects. Furthermore, 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. In order to model the spatial relation between different object instances, topological relationships could be beneficial for a general manner. In this work, we employed semantic context during training, by building a co-occurrence graph from the prior object appearance knowledge. We further use topological relation for a general spatial reasoning to enhance the prediction results during post-processing.

### 7.2.3 Segmentation Architectures

Semantic segmentation is a demanding computer vision task that involves assigning per-pixel labels corresponding to object categories within an image. Achieving accurate predictions in semantic segmentation requires capturing details related to the object category, its precise location, and its shape. Traditional and successful approaches in this domain have been predominantly based on Convolutional Neural Networks (CNNs), with notable examples [83, 20, 21, 25]. Recent advances in semantic segmentation have witnessed a transition toward transformer-based methods, leveraging the success of transformers in language and vision tasks [125, 15]. Transformer-based models [53, 114, 145, 56], have demonstrated remarkable success in semantic segmentation tasks. Notably, MaskFormer [27] has approached semantic segmentation as a mask classification problem, aligning with earlier works that treated semantic segmentation through mask classification [47, 16, 33]. MaskFormer adopts a transformer decoder with object queries, inspired by the effective architecture of the DETR model [15].

Instance segmentation methodologies typically fall into two distinct categories: proposal-based and segmentation-based approaches, each offering unique strategies to address the complexities of the task. In the realm of proposal-based methods, exemplified by the likes of Mask R-CNN [49] and Cascade R-CNN [12], the initial phase involves the detection of a series of bounding boxes. Subsequently, masks are generated for each identified bounding box, providing a fine-grained delineation of instances within the image. On the other hand, segmentation-based methods, such as Spatial Instance [92] and Associate Instance [93], take a different route. Here, semantic segmentation networks come to the forefront, leveraging their ability to predict pixel classes. This process results in a detailed understanding of the semantic content within an image. Post-processing techniques are often employed in this approach to refine and enhance the segmentation results.

Panoptic segmentation, introduced through the lens of Panoptic-FPN [62], emerged as a pioneering concept aiming to integrate instance and semantic segmentation tasks. Panoptic-FPN, among the early architectures in this domain, innovatively incorporated distinct branches for handling instance and semantic segmentation. However, the landscape evolved with remarkable advancements, particularly with the integration of transformer-based architectures. Works [126, 160, 161, 127, 26, 27] have substantially elevated performance standards in the realm of panoptic segmentation . These transformer-based approaches have demonstrated enhanced capabilities in capturing intricate relationships within images, further refining the fusion of instance and semantic segmentation for more accurate and comprehensive scene understanding. Oneformer, a recent unified framework [55] uses a transformer-based text mapper to provide text-based representation for the objects in the image. In our research, we leverage a Graph Convolutional Network (GCN) to model the semantic relationships between objects, as introduced by Kipf and Welling [61]. This choice of utilizing GCN is motivated by its lightweight nature, working with a heavy backbone model that has been pre-trained, making it well-suited for efficiently capturing and representing the intricate semantic connections among objects in our specific context.

## 7.3 Method

Our proposed context reasoning framework is shown in Fig. 7.2. We follow the implementation of Oneformer[55], by replacing the query formulation component with our semantic context reasoning component, which reduces the number of parameters and training time significantly. We use a CNN backbone as the feature extractor. In the Semantic Context Fusion (SCF) component (Section 7.3.1), we use pretrained word embeddings to represent object labels. A contextual co-occurrence matrix is built over the prior object appearance knowledge to describe the relation among different classes. Then we feed the pretrained word embeddings into a Graph Convolutional Network

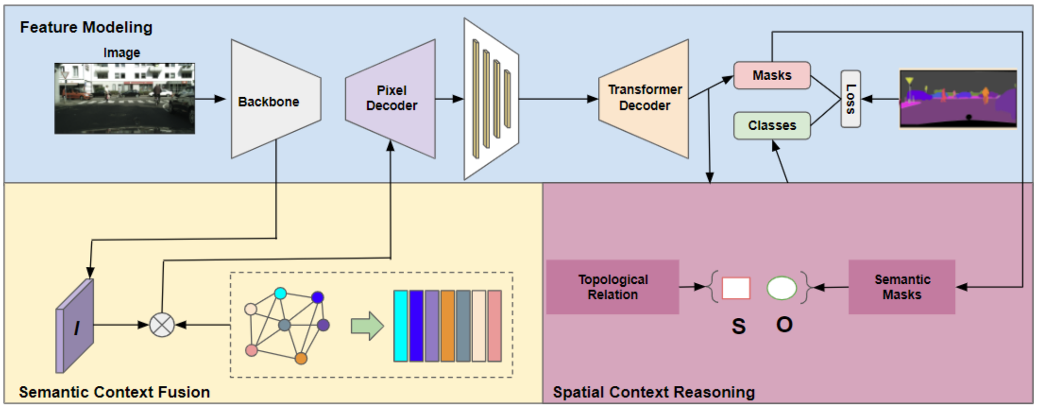


Figure 7.2: **Framework Overview.** (a) We extract features from an input image using a backbone feature extractor. (b) Next, we harmonize semantic knowledge with visual features using Semantic Context Fusion (SCF). We project image features (I) into the semantic space and subsequently input the resulting output into pixel decoder. (c) The Spatial Context Reasoning (SCR) was integrated to fine-tune the predicted candidates by synergizing topological relationships and semantic masks among identified instances.

(GCN) [61] by learning over the co-occurrence matrix. The GCN builds a semantic space and projects the feature into the pixel decoder. We further integrate a Spatial Contextual Reasoning (SCR) component (Section 7.3.2) to optimize the predictions by using the general spatial relations between the thing instances and the stuff masks. In the following, we will detail each component of our proposed lightweight context learning and reasoning framework.

### 7.3.1 Semantic Context Fusion

Semantic information indeed plays a crucial role in visual detection tasks, providing valuable insights to enhance the detection process. For example, for panopic segmentation of Cityscapes, they include pedestrian, vehicle, bicycle (bike), motorcycle, etc. These categories act as guiding factors for the model to learn and incorporate semantic context using text embeddings. The text embeddings, obtained from pre-trained language models, are utilized to generate semantic spaces that can be effectively fused with the visual information obtained from the segmentation process. This integration of semantic context with text embeddings allows our framework to automatically leverage valuable semantic information to improve the overall performance, while minimizing the need for extensive component modification.

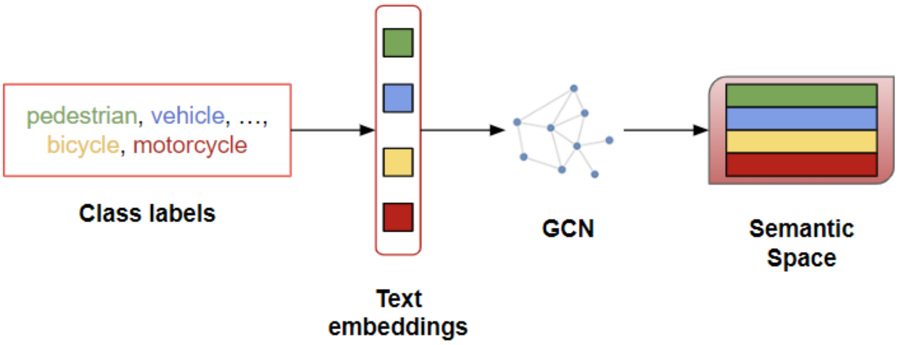


Figure 7.3: The demonstration of Semantic Context Fusion component.

As shown in Fig. 7.3, we use a Graph Convolutional Network (GCN) in our semantic context reasoning component. It takes the labels description $H_{labels} \in \mathbb{R}^{nxd}$ and context co-occurrence matrix $A \in \mathbb{R}^{nxn}$ as input, where $n$ is the number of labels (number of nodes) and $d$ is the dimensionality of the label word embedding (dimensionality of the node feature). The output of the GCN network is represented as the label semantic space $H'_{labels} \in \mathbb{R}^{nxD}$. Then we project the image feature extracted from the backbone feature extractor $f_{image}$ into the semantic space:

$$\mathbf{F} = H'_{labels} f_{image} \tag{7.1}$$

where $F$ denotes the updated image feature, which is fed into the pixel decoder. The ultimate output will be the prediction of the class label using a softmax function, which is described in Section 5.2.2. The detailed implementation is discussed in Section 5.3.

To articulate the co-occurrence relationships between distinct categories, we adopt a label occurrence dependency model based on conditional probability, drawing inspiration from the work by Chen et al. [24]. In our formulation, $P(L_j|L_i)$ signifies the probability of label $L_j$ occurring when label $L_i$ is present. We construct a contextual graph, represented by $A \in \mathbb{R}^{nxn}$, encapsulating the interplay between various categories. This graph is informed by prior knowledge of label occurrences gleaned from the training data, where $n$ denotes the total number of label categories. It's important to note that our model incorporates a background label, effectively accounting for regions that do not align with any specific category within the given context. This approach allows us to comprehensively capture and utilize label co-occurrence information for enhanced contextual understanding.

### 7.3.2 Spatial Context Reasoning

In our proposed Spatial Context Reasoning (SCR) component, we exploit topological relationships as a foundational element for modeling spatial relations among diverse objects. These relationships offer a broad and abstract

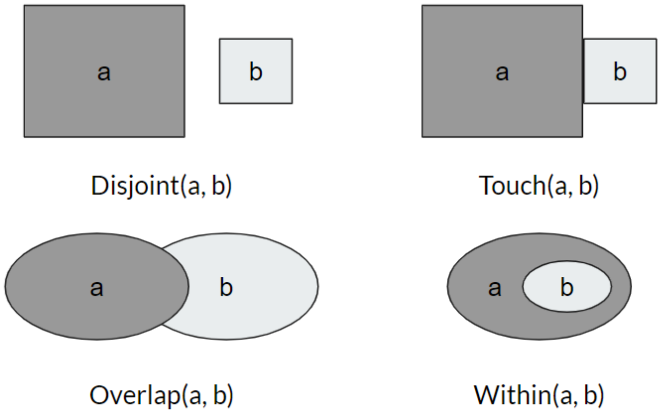


Figure 7.4: The visualization of the commonly used topological relationships from [30] and [39].

representation of instance interactions, encompassing concepts such as *overlap*, *within*, *touch*, and others. They effectively encapsulate the overall spatial configuration and arrangement of objects within a scene, accounting for nuances like adjacency, containment, and occlusion. Figure 7.4 visually illustrates the visualization of these topological relationships, showcasing how different objects can be interrelated in terms of their spatial positions and co-occurrence. By integrating topological reasoning into our framework, we facilitate a more profound understanding of the spatial context, thereby enhancing prediction performance and enabling richer semantic interpretations of the scene.

We utilize a predicate *pred*, such as *above*, *under*, etc., to describe the directional relation between a subject and object pair $[S, O]$, along with the topological relationship $t$, such as *overlap* and *within*. This general relation

$R$ is defined as shown in Equation 7.2:

$$R[S, O] = pred[t(S, O)] \tag{7.2}$$

For instance, in urban settings, a common spatial relationship for rider is that a bicycle is usually located under a person. The general relationship between a pedestrian and sidewalk can be described as $R[pedestrian, sidewalk] = under[overlap(pedestrian, sidewalk)]$. It is important to note that the general spatial relation is inversible, meaning that a pedestrian is on the sidewalk, and sidewalk can be considered under a pedestrian. If an instance is proposed and satisfies the condition defined by Equation 7.2. We propose it as an instance prediction and keep it for evaluation.

We further use the predicted stuff *semantic masks* in our spatial context reasoning component. This addition allows us to segment large stuff such as sidewalks and roads using same model, which could help improve spatial reasoning in larger scenes. To measure the overlap between subject-object pairs, we use the intersection over subject (IoS) metric to describe the spatial relation, as defined as:

$$IoS = \frac{(A_s \cap A_o)}{(A_s)} \tag{7.3}$$

where $A_s$ and $A_o$ denote the area of the subject and area of the object. The area can be bounding box or semantic mask based on the specific scenarios. This formulation enables us to capture the relative spatial arrangement of instances in a scene, which is valuable for improving the localization across predicted instances. Utilizing the Intersection over Scene (IoS) metric enables precise capture of spatial relations for the subject. This approach proves par-

Table 7.1: Spatial Context Reasoning settings for Cityscapes Dataset . O_T: Overlap_threshold.

| [Subject, Object] | Predicate | Topology | O_T |
|---|---|---|---|
| [rider, bicycle] | under | overlap | 0.48 |
| [rider, motorcycle] | under | overlap | 0.5 |
| [pedestrian, vehicle] | under | overlap | 0.68 |
| [pedestrian, pedestrian] | - | overlap | 0.76 |
| [pedestrian, road] | under | overlap | 0.2 |
| [pedestrian, sidewalk] | under | overlap | 0.13 |

ticularly effective in scenarios involving the interaction between objects and stuff, such as pedestrians and sidewalks, where traditional metrics like Intersection over Union (IoU) may yield inaccurate spatial reasoning. Table 7.1 lists the spatial context post-process settings for the Cityscapes Dataset, which will be discussed in Section 7.4.

## 7.4 Experiments

### 7.4.1 Datasets and Evaluation Metrics

**Datasets.** In our experimentation, we utilize the Cityscapes dataset, a benchmark widely acknowledged in the computer vision community. The **Cityscapes** dataset, as detailed in [31], encompasses a total of 19 classes, further categorized into 11 "stuff" classes and 8 "thing" classes. The dataset is partitioned into 2,975 images for training, 500 for validation, and 1,525 for testing. This diverse dataset provides a robust evaluation ground for our methods, allowing us to assess performance across a spectrum of urban scenes and object classes.

**Evaluation Metrics.** We use the standard evaluation metric defined in [63], called Panoptic Quality (PQ). The panoptic quality metric is a comprehensive evaluation measure for panoptic segmentation, assessing the quality of both "stuff" and "thing" predictions in a unified manner. Essentially, it provides a comprehensive measure of the model's ability to understand and segment diverse elements in a scene, offering a combined score that reflects both accuracy in recognizing objects and understanding the overall scene composition. We further report PQ scores per category for the Cityscapes dataset.

### 7.4.2 Experiment Settings

In our task, Oneformer [55] serves as the foundational architecture, by removing the Query Formulation (QF) component, with a focus on training our Graph Convolutional Network (GCN) model. Since Oneformer is a combination of two transformers: one for feature modeling, and one for the query formation, we can simply remove the former, i.e., the QF component and replace it with our GCN model as a lightweight alternative. The pretrained weights of the base model (Oneformer-QF) are frozen, and training exclusively pertains to our GCN model. Given that the Oneformer model has attained state-of-the-art (SOTA) performance, it can be reasonably assumed that it has been optimized; note that we also use the same base model with our GMC framework for a fair comparison. The additional reason for freezing the pre-trained parameters is mainly for training efficiency

Table 7.2: Parameter and inference comparison.

| Method | Parameters | Inference Time |
|---|---|---|
| Oneformer - QF | 202M | 1.076s |
| Oneformer(-QF) + SCF | 205M | 1.092s |
| Oneformer(-QF) + SCF + SCR | 205M | 1.101s |
| Oneformer with QF | 220M | 1.503s |

since the base model is relatively large (with 200M parameters). Our GCN network comprises two layers, employing LeakyReLU [84] as the activation function. For class query representations, we employ 300-dimensional word vectors ($d = 300$ in Section 7.3.1) sourced from GloVe [97], a language model pretrained on the Wikipedia dataset. Stochastic Gradient Descent (SGD) is chosen as the optimizer during training, with momentum and weight decay set to 0.95 and 0.0001, respectively. The initial learning rate is established at 0.005, undergoing a 0.25 decay every 8 epochs. The network undergoes training for a total of 100 epochs. Our topological relationship modeling primarily focuses on "things," and the detailed settings, including the specific classes modeled and their prior knowledge statistics, are outlined in Table 7.1. The add-ons in our framework with both semantic and spatial context components only increase the size of the final model by only about 3M parameters, comparing to the 18M parameters of the QFcomponent in the full Oneformer model (Table 7.2; details will be discussed in the Ablation Studies in Section 7.4.4).

Table 7.3: **Performance Comparison on Cityscapes validation set.** QF: Query Formulation. SCF: Semantic Context Fusion. SCR: Spatial Context Reasoning.

| Method | Backbone | PQ |
|---|---|---|
| Oneformer - QF | ConvNeXt-L [81] | 62.5 |
| Oneformer(-QF) + SCF | ConvNeXt-L [81] | 65.9 |
| Oneformer(-QF) + SCF + SCR | ConvNeXt-L [81] | 66.3 |
| Oneformer with QF | ConvNeXt-L [81] | 68.5 |

### 7.4.3 Main Results

In our comparative analysis on the Cityscapes dataset, we juxtapose our results with those obtained using Oneformer in Table 7.3. Notably, we replace the Query Formulation (QF) component in Oneformer with our Semantic Context Fusion (SCF) component. The baseline Oneformer (without QF) achieves a performance metric of 62.5 on the Cityscapes dataset. Upon integrating our SCF component, our framework exhibits a notable improvement, surpassing the simple Oneformer version by 3.4%. Further enhancement is observed with the inclusion of the Spatial Context Reasoning (SCR) component, contributing an additional 0.4% improvement over the Semantic Context Fusion (SCF) component alone, resulting in an overall 3.8% improment over the baseline Oneformer. It's noteworthy that while our contributions enhance Oneformer's performance, the original Oneformer framework with the QF component maintains its status as the top-performing model in this comparison. But we want to note two things: (1) we refrained from finetuning the model, as the pretrained weights already attained the state-of-the-art (SOTA) performance on the Cityscapes dataset. (2) The two addons (SCF and SCR) are very lightweight (with only 3M additional param-

Table 7.4: Performance comparison on all categories for Cityscapes Dataset.

| Method | road | sidewalk | building | wall | fence | pole | traffic light | traffic sign | veg |
|---|---|---|---|---|---|---|---|---|---|
| Oneformer | 98.7 | 80.8 | 90.6 | 47.5 | 53.7 | 69.5 | 60 | 75.1 | 91 |
| Oneformer - QF | 98.3 | 76.8 | 88.8 | 36.6 | 39.7 | 59 | 51.6 | 65.7 | 90.4 |
| Oneformer(-QF) + SCF | 98.7 | 79.6 | 90.1 | 45.5 | 46.3 | 65.4 | 59.6 | 74.5 | 91.5 |
| Oneformer(-QF) + SCF + SCR | 98.7 | 79.6 | 90.1 | 45.5 | 46.3 | 65.4 | 59.6 | 74.5 | 91.5 |

| Method | terrain | sky | person | rider | car | truck | bus | motorcycle | bicycle |
|---|---|---|---|---|---|---|---|---|---|
| Oneformer | 47.6 | 91.8 | 61.2 | 57.4 | 72.6 | 61.9 | 71.6 | 56.5 | 51.4 |
| Oneformer - QF | 46.8 | 89.6 | 57.7 | 53.5 | 68.9 | 52.6 | 62.2 | 51.2 | 46 |
| Oneformer(-QF) + SCF | 47.9 | 90.3 | 59.2 | 56.6 | 70 | 54.9 | 63.7 | 51.8 | 46.2 |
| Oneformer(-QF) + SCF + SCR | 47.9 | 90.3 | 60.2 | 57 | 72 | 54.9 | 63.7 | 53.4 | 46.6 |

eters), comparing to the 18M additional parameters in the QF components (Table 7.2). By taking this approach, it allows us to assess the effectiveness of our lightweight context implementation. By leveraging the existing high-performing pretrained weights, we aim to gauge how well our added lightweight context components contribute to and enhance the model's capabilities in the specific context of panoptic segmentation on the Cityscapes dataset. This strategy allows us to isolate and evaluate the impact of our context implementation without introducing changes to the pretrained model's already-established excellence.

As shown in Table 7.4, in a detailed examination of panoptic segmentation results across individual categories, our analysis reveals consistent performance improvements with the integration of our Semantic Context Fusion (SCF) component. Notably, our framework outperforms the Oneformer without the Query Formulation (QF) component across all categories. The subsequent addition of the Spatial Context Reasoning (SCR) component leads to further advancements, particularly noteworthy in the categories of person, rider, car, motorcycle, and bicycle, where Spatial Context Reasoning are added, with improvements of +1.5%, +3.1%, +1.1%, +0.6%, and +0.2% observed, respectively.

### 7.4.4 Ablation Studies

We further study the parameter and inference time between oneformer (with and without the QF component) and our context component implementation (Table 7.2). The incorporation of our Semantic Context Component results in a reduction in the total number of parameters compared to the original Oneformer (with QF). This reduction in parameter count signifies a more streamlined and efficient model architecture. Moreover, when assessing the practical implications, particularly in terms of inference time, our model showcases a significant improvement. When tested on a single Nvidia 3080Ti, the inference time is notably reduced from 1.503 seconds (as observed in the original Oneformer with QF) to 1.092 seconds in the original Oneformer with our semantic context fusion (SCF) component instead of QF; adding the spatial context reasoning (SCR) only increases the time by 9 ms . This observation could be attributed to the simplicity of the Graph Convolutional Network (GCN) model compared to the transformer-based query formulation component. This translates to a substantial 27.3% enhancement in computational efficiency, implying that our model processes predictions more swiftly with promising performance. This improvement is particularly relevant in real-time applications or scenarios where rapid inference is crucial.

In the above experiments with our Spatial Context Reasoning (SCR), we mainly model the relationships between pedestrians with other contextual

Table 7.5: Additional spatial context post-processing settings.

| [Subject, Object] | Predicate | Topology | O_T |
|---|---|---|---|
| [Truck, road] | under | overlap | 0.38 |
| [Bus, road] | under | overlap | 0.24 |

classes, showing particular performance improvement of pedestrians and the contextual classes. To assess the adaptability of our SCR component, we conducted additional experiments by modeling two more classes (truck and bus) against their background (i.e., road), using prior knowledge statistics derived from the training data (refer to Table 7.5). As shown in Table 7.6, even with the sole application of the SCR component, our framework demonstrates slight yet notable performance improvements. Specifically, there is a 0.6% enhancement in the truck class and a 0.7% improvement in the bus class. These findings suggest that our SCR component could effectively leverage additional spatial relationships between instances to enhance performance, showcasing its potential to improve results without the need for additional training steps. With the combination of SCF and SCR component, our framework can further improve both class, with + 2.8% for truck, and +1.6% for bus, on top of solo SCR component, respectively. The overall PQ also improved from 66.3% to 66.5%. This highlights the possible flexibility and utility of our SCR component in augmenting the model's predictive capabilities.

Table 7.6: Performance comparison on updated categories for Cityscapes Dataset.

| Method | truck | bus | PQ |
|---|---|---|---|
| Oneformer - QF | 52.6 | 62.2 | 62.5 |
| Oneformer(-QF) + SCR (Updated) | 53.2 | 62.9 | 62.6 |
| Oneformer(-QF) + SCF + SCR (Updated) | 56 | 64.5 | 66.5 |

## 7.5 Summary

In this research, we present a novel lightweight framework for context learning and reasoning in panoptic segmentation. Our approach integrates both semantic and spatial contexts, enhancing the extraction of visual features and refining segmentation quality. While our experiments yield promising results, it's important to note that our current components rely on prior knowledge from the training set and predominantly focus on the ”things” category. To ensure the robustness and applicability of our framework, further validation across diverse benchmarks is imperative. Conducting experiments on various datasets will offer valuable insights into the generalizability and adaptability of our approach in different scenarios. Our overarching objective is to provide a lightweight and versatile framework for the panoptic segmentation task, offering the flexibility to incorporate various forms of context information. Given the limited exploration of context integration in panoptic research, we aspire to pave the way for a new direction in the design of context frameworks. Our work seeks to contribute not only to improved panoptic segmentation performance but also to the broader discourse on effective context utilization in computer vision tasks. Currently we only provide some preliminary results for a panoptic segmentation task, with a single backbone on a single dataset. We hope this work will inspire more studies in integrating our proposed contextual components on latest segmenters, such as Segment Anything (SAM), and to test if and how much the GMC framework can improve their performance.

# Chapter 8

# Conclusions and Future Directions

## 8.1 Summary

In this thesis, we have presented a general context learning and utilization framework with multistage, individual contextual components. Our proposed framework guided context learning from data labeling, contextual graph during training and general spatial reasoning during post processing. Below we summarize our contributions that lead to future research directions:

- We present a comprehensive survey of context understanding in computer vision, with a taxonomy to describe context in different types (spatial, temporal and others) and different levels (prior knowledge, global, local). Furthermore, we review various context based integration in two categories: image-based context integration and video-based context integration. The taxonomy of context not only help us identify the different stages and components in our general framework, but can be used to further investigate other components that could be integrated into the general framework.

- We propose the MultiCLU framework for mutli-stage context learning and utilization for storefront accessibility detection. We design specific context learning mechanisms for storefront accessibility detection, by employing the specific relationship between storefront accessibility objects (Doors, Knobs, stairs, etc.). we further introduce a new evaluation metric for the knob category in our task, which could provide a new way to re-think evaluation standards in real world applications.

- We develop an AI-enabled Storefront Accessibility Annotation and Localization Platform. We apply our special MultiCLU framework into our previous developed Doorfront platform, to enable AI-based prelabeling. We also introduce an online machine learning mechanism to iteratively train the MultiCLU model, by using newly labeled storefront accessibility objects. By integrating our MultiCLU framework, our new platform not only significantly improves the efficiency of storefront accessibility data collection, but also improves user experience. We hope this can shine light to other data collection efforts.

- We design a general multistage context (GMC) framework for various visual detection and segmentation tasks, working with different base architectures. With the current implementation, the GMC framework consists of three contextual components: Local Contextual Representation (LCR), Semantic Context Fusion (SCF), and Spatial Contextual

Reasoning (SCR). These contextual components take advantages of different contextual information, and guide the deep learning detector through labeling, training and post processing. Each component can be applied individually and in combination. The framework is further extended to work for other visual tasks such as semantic segmentation. The framework can be applied to different visual tasks (including detection and segmentation) and work with different deep learning architectures, without much changes in code. We believe the framework can be further extended to include other context components, and to work for other visual tasks.

In summary, we demonstrated that our contextual components can be applied individually and in combinations, and can be easily added and removed from the base architecture. We further integrated our model into our Smart DoorFront platform for labeling automation and validation. 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.

## 8.2 Future Directions

In this section, we will further discuss some potential future directions on how we can make better use of context in computer vision research. In the past, context information has been integrated and utilized over context-free methods, and it has been achieved great success and surpass the performance of context-free methods, in both image-based tasks and video-based tasks. In this thesis, we present a taxonomy of visual context, and based on the analysis, we propose a general multistage context framework that can be applied to both visual detection and visual segmentation. However, there are still space for further improvement within the current GMC framework and unexplored aspects in incorporating other context information and in various other tasks.

Here are some potential future directions on how we could make better use of context in computer vision research.

**Contextual Data Augmentation:** The integration of context including our work has witnessed notable success across various computer vision tasks, including object detection, image recognition, and pedestrian detection. Much of this success has revolved around aggregating context features into context-free methods, showcasing improvements in overall performance. However, a notable gap in the existing research landscape pertains to the limited exploration of leveraging context information for data augmentation. While flipping, rotation, crop, and translation are common data augmentation techniques, they often fall short in addressing the challenges associated with small object detection. Even contemporary context-free methods like Faster R-CNN, YOLOv4, and SSD grapple with the intricacies of small object detection. Notably, a singular work by Dvornik et al. [38] stands out for employing semantic context and local-level context to augment data specifically for small objects. Despite the demonstrated efficacy of context in aiding the detection of small objects, there remains a gap in devising superior methods for augmenting data in this context. Our exploration of local context labeling by enlarging the bounding boxes of labels aims to inspire further research into innovative data augmentation techniques that harness the power of context (including local context) for addressing the challenges associated with small object detection.

**Spatial Context Relation Definition:** Defining spatial relations for diverse object categories within a dataset is a multifaceted challenge. In datasets featuring a wide array of object types, each with its own unique characteristics and spatial dependencies, establishing universally applicable spatial relations automatically becomes intricate. The complexity is compounded by the need for these relations to be not only meaningful but also capable of accommodating the inherent diversity and variability present among different categories. Addressing this challenge requires a nuanced and systematic approach that considers the specific attributes and contextual nuances associated with each category. Furthermore, developing a method that robustly learns and generalizes spatial relations across such diversity necessitates ongoing exploration and research efforts. A comprehensive understanding of the intricate interplay between various object categories and their spatial contexts is crucial for the development of an adaptable and effective spatial relation framework.

**Filling the Gaps in the Context Taxonomy:** In this research endeavor, our focus has primarily delved into the exploration of local, global, and semantic contexts, predominantly within the visual, spatial domain. The developed context taxonomy offers a structured classification comprising three overarching types: spatial, temporal, and other, each of which are further refined into three hierarchical levels: prior, global, and local. A thorough analysis of this taxonomy exposes certain domains that are yet to be comprehensively investigated. For example, we haven't considered the temporal context information in our general context framework. Notably, the exploration of long-term temporal context and temporal semantic context remains relatively limited within the existing body of literature. Furthermore, other dimensions of context beyond the conventional spatial and temporal realms, particularly those encompassing diverse modalities and functionalities, intentions, or purposes, demand a heightened focus for a more holistic understanding. Shifting our attention to the architectural landscape, conventional convolutional neural networks (convNets) have conventionally served as the cornerstone for visual feature extraction. However, this paradigm has often marginalized non-visual features. While notable efforts [24, 173, 156, 120, 99] have sought to model relations between visual and non-visual contexts, a persistent challenge lies in refining representations of visual-other context relations to effectively bridge existing domain gaps.

The transformer architecture has emerged as a focal point in contemporary computer vision research, with its attention mechanism being a key contributor to contextual learning. Despite the inherent contextual capabilities within the transformer architecture, our experiments, particularly with DETR, reveal that explicitly incorporating additional context information is more effective in enhancing overall performance. This observation suggests that, while transformers inherently capture context through attention mechanisms, supplementing them with explicit context information yields more efficient outcomes. Notably, the majority of recent works on transformers in computer vision have primarily concentrated on static or spatial contexts. In contrast, only a limited number of recent studies [80, 5, 91] have delved into the integration of temporal information. In this context, there is a clear call for the development of novel architectures expressly tailored for context learning and integration. These architectures should transcend the limitations of existing models, providing innovative solutions that can propel the field of context-aware computing into new frontiers of understanding and applicability.

**Contextual Evaluation:** In the realm of computer vision, the assessment of model performance often relies on standardized evaluation metrics, such as Intersection over Union (IOU) for object detection. However, it is crucial to recognize that the alignment of these metrics with real-world accuracy is not always straightforward. Our work, particularly in the context of storefront accessibility detection, exemplifies a scenario where the practical implications of model outputs supersede precise localization details. For instance, when a person is searching for a door knob, knowing the estimated location (left or right side of the door) may hold more practical utility than an exact and detailed specification (e.g., 1.5m high on the left). This highlights the necessity for a contextual evaluation framework that aligns with the nuanced requirements of real-world applications. Such an approach is not only pertinent to object detection tasks but also holds the potential to enhance the efficacy of a broader spectrum of computer vision applications.

**Real-time Implementation:** Our context components exhibit a remarkable degree of versatility, making them suitable for both standalone and cloud applications. Their inherent potential extends beyond their role as mere context components to improve detection or segmentation performance, as we envision them functioning as a dynamic API tailored for real-time applications, such as location and navigation services for individuals who have blindness or low vision. The beauty of this lies in the adaptability of our components, which opens avenues for their integration into a lightweight deep learning model for real-time implementations. This integration, in turn, could empower the creation of mobile applications, ready to tackle real-world scenarios with agility and precision. The flexibility and efficiency of our components position them as not just tools, but as transformative elements with the capacity to elevate the user experience in a myriad of contexts.

## 8.3 Publications During the Thesis Work

1. Wang, X., Tang, H. and Zhu, Z. GMC: A General Framework of Multistage Context Learning and Utilization for Visual Detection Tasks. Computer Vision and Image Understanding. Volume 241, 2024, 103944, ISSN 1077-3142, DOI: 10.1016/j.cviu.2024.103944.

2. Wang, X, and Zhu, Z. Context understanding in computer vision: A survey. Computer Vision and Image Understanding. Volume 229, March 2023, 103646, ISSN 1077-3142, DOI:10.1016/j.cviu.2023.103646.

3. Wang, X., Tang, H. and Zhu, Z. A General Context Learning and Reasoning Framework for Object Detection in Urban Scenes. In Proceedings of the 18th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2023) - Volume 5: VISAPP; ISBN 978-989-758-634-7, SciTePress, pages 91-102. DOI: 10.5220/0011637600003417.

4. Wang, X, Chen, J., Tang, H. and Zhu, Z. MultiCLU: Multi-stage Context Learning and Utilization for Storefront Accessibility Detection and Evaluation. ACM International Conference on Multimedia Retrieval, Pages 304–312, Newark, NJ, USA, June 27-30, 2022.

5. Wang, X, Liu, J., Tang, H., Zhu, Z. and Seiple, W. An AI-enabled Annotation Platform for Storefront Accessibility and Localization. Journal on Technology and Persons with Disabilities, 2023, Volume 11.

6. *Olmschenk, G., *Wang, X., Tang, H. and Zhu, Z. Impact of Labeling

Schemes on Dense Crowd Counting Using Convolutional Neural Networks with Multiscale Upsampling. International Journal of Pattern Recognition and Artificial Intelligence, Vol. 35, No. 13, September 15, 2021. (* Corresponding authors with equal contributions.)

7. Tang, H., Wang, X., Olmschenk, G. Feeley, C., Zhu, Z. Assistive Navigation and Interaction with Mobile & VR Apps for People with ASD. The 35th CSUN Assistive Technology Conference, March 9-13, 2020.

# Bibliography

[1] Global estimates of vision loss. https://www.iapb.org/learn/vision-atlas/magnitude-and-projections/global, 2021.

[2] Collect accessibility data. https://doorfront.org, 2022.

[3] Google street view api. https://developers.google.com/maps/documentation/streetview/ove 2022.

[4] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2425–2433, 2015.

[5] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF International Conference on Computer vision*, pages 6836–6846, 2021.

[6] Moshe Bar and Elissa Aminoff. Cortical analysis of visual context. *Neuron*, 38(2):347–358, 2003.

[7] Sara Beery, Grant Van Horn, and Pietro Perona. Recognition in terra incognita. In *Proceedings of the European Conference on Computer Vision*, pages 456–473, 2018.

[8] Sara Beery, Guanhang Wu, Vivek Rathod, Ronny Votel, and Jonathan Huang. Context r-cnn: Long term temporal context for per-camera object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13075–13085, 2020.

[9] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.

[10] Philipp Bomatter, Mengmi Zhang, Dimitar Karev, Spandan Madan, Claire Tseng, and Gabriel Kreiman. When pigs fly: Contextual reasoning in synthetic and natural scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 255–264, 2021.

[11] Zhaowei Cai, Quanfu Fan, Rogerio S Feris, and Nuno Vasconcelos. A unified multi-scale deep convolutional neural network for fast object detection. In *Proceedings of the European Conference on Computer Vision*, pages 354–370, 2016.

[12] Zhaowei Cai and Nuno Vasconcelos. Cascade R-CNN: Delving into high quality object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[13] Peter Carbonetto, Nando de Freitas, and Kobus Barnard. A statistical model for general contextual object recognition. In *Proceedings of the European Conference on Computer Vision*, pages 350–362, 2004.

[14] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Proceedings of the European Conference on Computer Vision*, pages 213–229, 2020.

[15] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Proceedings of the European Conference on Computer Vision*, 2020.

[16] Joao Carreira, Rui Caseiro, Jorge Batista, and Cristian Sminchisescu. Semantic segmentation with second-order pooling. In *Proceedings of the European Conference on Computer Vision*, 2012.

[17] Marco Cavallo. 3d city reconstruction from google street view. https://api.semanticscholar.org/CorpusID:26811584, 2015.

[18] David Abou Chacra and John Zelek. 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, 2022.

[19] Chenyi Chen, Ming-Yu Liu, Oncel Tuzel, and Jianxiong Xiao. R-cnn for small object detection. In Shang-Hong Lai, Vincent Lepetit,

Ko Nishino, and Yoichi Sato, editors, *Proceedings of the Asian Conference on Computer Vision*, pages 214–230, Cham, 2017. Springer International Publishing.

[20] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. In *Proceedings of the International Conference on Learning Representations*, 2015.

[21] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[22] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv*, 2017.

[23] Yifeng Chen, Guangchen Lin, Songyuan Li, Omar El Farouk Bourahla, Yiming Wu, Fangfang Wang, Junyi Feng, Mingliang Xu, and Xi Li. Banet: Bidirectional aggregation network with occlusion handling for panoptic segmentation. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3792–3801, 2020.

[24] Zhao-Min Chen, Xiu-Shen Wei, Peng Wang, and Yanwen Guo. Multi-label image recognition with graph convolutional networks. In *Proceed-*

*ings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5177–5186, 2019.

[25] Bowen Cheng, Liang-Chieh Chen, Yunchao Wei, Yukun Zhu, Zilong Huang, Jinjun Xiong, Thomas S Huang, Wen-Mei Hwu, and Honghui Shi. Spgnet: Semantic prediction guidance for scene parsing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[26] Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[27] Bowen Cheng, Alexander G. Schwing, and Alexander Kirillov. Perpixel classification is not all you need for semantic segmentation. In *Proceedings of the Conference on Neural Information Processing Systems*, 2021.

[28] Myung Jin Choi, Antonio Torralba, and Alan S Willsky. A tree-based context model for object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(2):240–252, 2011.

[29] Myung Jin Choi, Antonio Torralba, and Alan S Willsky. Context models and out-of-context objects. *Pattern Recognition Letters*, 33(7):853–862, 2012.

[30] Eliseo Clementini, Paolino Di Felice, and Peter van Oosterom. A small set of formal topological relationships suitable for end-user interaction.

In *Proceedings of the International Symposium on Spatial Databases*, pages 277–295, 1993.

[31] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016.

[32] Zhen Cui, Chunyan Xu, Wenming Zheng, and Jian Yang. Context-dependent diffusion network for visual relationship detection. In *Proceedings of the ACM International Conference on Multimedia*, pages 1475–1482, 2018.

[33] Jifeng Dai, Kaiming He, and Jian Sun. Convolutional feature masking for joint object and stuff segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2015.

[34] T. E. de Campos, B. R. Babu, and M. Varma. Character recognition in natural images. In *Proceedings of the International Conference on Computer Vision Theory and Applications*, 2009.

[35] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[36] Santosh K Divvala, Derek Hoiem, James H Hays, Alexei A Efros, and Martial Hebert. An empirical study of context in object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1271–1278, 2009.

[37] Yuning Du, Genquan Duan, and Haizhou Ai. Context-based text detection in natural scenes. In *Proceedings of the IEEE International Conference on Image Processing*, pages 1857–1860, 2012.

[38] Nikita Dvornik, Julien Mairal, and Cordelia Schmid. Modeling visual context is key to augmenting object detection datasets. In *Proceedings of the European Conference on Computer Vision*, pages 364–380, 2018.

[39] Max J Egenhofer and Robert D Franzosa. Point-set topological spatial relations. *International Journal of Geographical Information System*, 5(2):161–174, 1991.

[40] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *Proceedings of the IEEE/CVF International Journal of Computer Vision*, 88(2):303–338, 2010.

[41] Yuan Fang, Kingsley Kuan, Jie Lin, Cheston Tan, and Vijay Chandrasekhar. Object detection meets knowledge graphs. In *Proceedings of the International Joint Conferences on Artificial Intelligence*, 2017.

[42] Michael Fink and Pietro Perona. Mutual boosting for contextual inference. *Advances in Neural Information Processing Systems*, 16, 2003.

[43] Carolina Galleguillos and Serge Belongie. Context based object categorization: A critical survey. *Computer Vision and Image Understanding*, 114(6):712–722, 2010.

[44] Joshua OS Goh, Soon Chun Siong, Denise Park, Angela Gutchess, Andy Hebrank, and Michael WL Chee. Cortical areas involved in object, background, and object-background processing revealed with functional magnetic resonance adaptation. *Journal of Neuroscience*, 24(45):10223–10228, 2004.

[45] Chunhui Gu, Chen Sun, David A Ross, Carl Vondrick, Caroline Pantofaru, Yeqing Li, Sudheendra Vijayanarasimhan, George Toderici, Susanna Ricco, Rahul Sukthankar, et al. Ava: A video dataset of spatio-temporally localized atomic visual actions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6047–6056, 2018.

[46] Kotaro Hara, Jin Sun, Robert Moore, David Jacobs, and Jon Froehlich. Tohme: detecting curb ramps in google street view using crowdsourcing, computer vision, and machine learning. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*, pages 189–204, 2014.

[47] Bharath Hariharan, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. Simultaneous detection and segmentation. In *Proceedings of the European Conference on Computer Vision*, 2014.

[48] Dafang He, Xiao Yang, Wenyi Huang, Zihan Zhou, Daniel Kifer, and C Lee Giles. Aggregating local context for accurate scene text detection. In *Asian Conference on Computer Vision*, pages 280–296, 2016.

[49] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2017.

[50] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[51] Geremy Heitz and Daphne Koller. Learning spatial context: Using stuff to find things. In *Proceedings of the European Conference on Computer Vision*, pages 30–43, 2008.

[52] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[53] Zilong Huang, Xinggang Wang, Yunchao Wei, Lichao Huang, Humphrey Shi, Wenyu Liu, and Thomas S. Huang. Ccnet: Criss-cross attention for semantic segmentation. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[54] Soonmin Hwang, Jaesik Park, Namil Kim, Yukyung Choi, and In So Kweon. Multispectral pedestrian detection: Benchmark dataset and

baselines. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2015.

[55] Jitesh Jain, Jiacheng Li, Man Chun Chiu, Ali Hassani, Nikita Orlov, and H. Shi. Oneformer: One transformer to rule universal image segmentation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2989–2998, 2022.

[56] Jitesh Jain, Anukriti Singh, Nikita Orlov, Zilong Huang, Jiachen Li, Steven Walton, and Humphrey Shi. Semask: Semantically masking transformer backbones for effective semantic segmentation. *arXiv*, 2021.

[57] Chenhan Jiang, Hang Xu, Xiaodan Liang, and Liang Lin. Hybrid knowledge routed modules for large-scale object detection. *Advances in Neural Information Processing Systems*, 31, 2018.

[58] Justin Johnson, Agrim Gupta, and Li Fei-Fei. Image generation from scene graphs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1219–1228, 2018.

[59] Justin Johnson, Ranjay Krishna, Michael Stark, Li-Jia Li, David Shamma, Michael Bernstein, and Li Fei-Fei. Image retrieval using scene graphs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3668–3678, 2015.

[60] Dong-Jin Kim, Xiao Sun, Jinsoo Choi, Stephen Lin, and In So Kweon. Detecting human-object interactions with action co-occurrence priors.

In *Proceedings of the European Conference on Computer Vision*, pages 718–736, 2020.

[61] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[62] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[63] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[64] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123(1):32–73, 2017.

[65] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25, 2012.

[66] Zihang Lai, Senthil Purushwalkam, and Abhinav Gupta. The functional correspondence problem. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15772–15781, 2021.

[67] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[68] Jiaxu Leng, Yihui Ren, Wen Jiang, Xiaoding Sun, and Ye Wang. Realize your surroundings: Exploiting context information for small object detection. *Neurocomputing*, 433:287–299, 2021.

[69] Qizhu Li, Anurag Arnab, and Philip H. S. Torr. Weakly- and semi-supervised panoptic segmentation. In *Proceedings of the European Conference on Computer Vision*, 2018.

[70] Qizhu Li, Xiaojuan Qi, and Philip H. S. Torr. Unifying training and inference for panoptic segmentation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13317–13325, 2020.

[71] Yanwei Li, Xinze Chen, Zheng Zhu, Lingxi Xie, Guan Huang, Dalong Du, and Xingang Wang. Attention-guided unified network for panoptic segmentation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7019–7028, 2018.

[72] Yanwei Li, Hengshuang Zhao, Xiaojuan Qi, Yukang Chen, Lu Qi, Liwei Wang, Zeming Li, Jian Sun, and Jiaya Jia. Fully convolutional networks for panoptic segmentation with point-based supervision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45:4552–4568, 2021.

[73] Yining Li, Chen Huang, Chen Change Loy, and Xiaoou Tang. Human attribute recognition by deep hierarchical contexts. In *Proceedings of the European Conference on Computer Vision*, pages 684–700, 2016.

[74] Jeong-Seon Lim, Marcella Astrid, Hyun-Jin Yoon, and Seung-Ik Lee. Small object detection using context and attention. In *Proceedings of the International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, pages 181–186, 2021.

[75] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2117–2125, 2017.

[76] 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 *Proceedings of the European Conference on Computer Vision*, pages 740–755, 2014.

[77] Jiawei Liu, Hao Tang, William Seiple, and Zhigang Zhu. Annotating storefront accessibility data using crowdsourcing. *Journal on Technology and Persons with Disabilities*, 10:154–170, 2022.

[78] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *Proceedings of the European Conference on Computer Vision*, pages 21–37, 2016.

[79] Wei Liu, Andrew Rabinovich, and Alexander C. Berg. Parsenet: Looking wider to see better. *ArXiv*, abs/1506.04579, 2015.

[80] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3202–3211, 2022.

[81] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[82] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2015.

[83] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2015.

[84] Andrew L Maas, Awni Y Hannun, Andrew Y Ng, et al. Rectifier nonlinearities improve neural network acoustic models. In *Proceedings of the International Conference on Machine Learning*, 2013.

[85] Oisin Mac Aodha, Elijah Cole, and Pietro Perona. Presence-only geographical priors for fine-grained image classification. In *Proceedings of*

*the IEEE/CVF International Conference on Computer Vision*, pages 9596–9606, 2019.

[86] Oge Marques, Elan Barenholtz, and Vincent Charvillat. Context modeling in computer vision: techniques, implications, and applications. *Multimedia Tools and Applications*, 51(1):303–339, 2011.

[87] Gilberto Marzano, Joanna Lizut, and Luis Ochoa Siguencia. Crowdsourcing solutions for supporting urban mobility. *Procedia Computer Science*, 149:542–547, 2019.

[88] Markus Mathias, Rodrigo Benenson, Marco Pedersoli, and Luc Van Gool. Face detection without bells and whistles. In *Proceedings of the European Conference on Computer Vision*, pages 720–735, 2014.

[89] Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan Yuille. The role of context for object detection and semantic segmentation in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 891–898, 2014.

[90] Roozbeh Mottaghi, Sanja Fidler, Jian Yao, Raquel Urtasun, and Devi Parikh. Analyzing semantic segmentation using hybrid human-machine crfs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3143–3150, 2013.

[91] Daniel Neimark, Omri Bar, Maya Zohar, and Dotan Asselmann. Video transformer network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3163–3172, 2021.

[92] Davy Neven, Bert De Brabandere, Marc Proesmans, and Luc Van Gool. Instance segmentation by jointly optimizing spatial embeddings and clustering bandwidth. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[93] Alejandro Newell, Zhiao Huang, and Jia Deng. Associative embedding: End-to-end learning for joint detection and grouping. In *Proceedings of the Conference on Neural Information Processing Systems*, 2017.

[94] Sangmin Oh, Anthony Hoogs, Amitha Perera, Naresh Cuntoor, Chia-Chih Chen, Jong Taek Lee, Saurajit Mukherjee, JK Aggarwal, Hyungtae Lee, Larry Davis, et al. A large-scale benchmark dataset for event recognition in surveillance video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3153–3160, 2011.

[95] Stephen Palmer. The effects of contextual scenes on the identification of objects. *Memory and Cognition*, 3:519–526, 1975.

[96] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2536–2544, 2016.

[97] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1532–1543, 2014.

[98] Roland Perko and Aleš Leonardis. A framework for visual-context-aware object detection in still images. *Computer Vision and Image Understanding*, 114(6):700–711, 2010.

[99] Senthil Purushwalkam, Sebastia Vicenc Amengual Gari, Vamsi Krishna Ithapu, Carl Schissler, Philip Robinson, Abhinav Gupta, and Kristen Grauman. Audio-visual floorplan reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1183–1192, 2021.

[100] Andrew Rabinovich and Serge Belongie. Scenes vs. objects: a comparative study of two approaches to context based recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 92–99, 2009.

[101] Andrew Rabinovich, Andrea Vedaldi, Carolina Galleguillos, Eric Wiewiora, and Serge Belongie. Objects in context. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1–8, 2007.

[102] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems*, 28:91–99, 2015.

[103] Bryan C Russell, Antonio Torralba, Kevin P Murphy, and William T Freeman. Labelme: a database and web-based tool for image annotation. *International Journal of Computer Vision*, 77(1-3):157–173, 2008.

[104] M. S. Ryoo and J. K. Aggarwal. UT-Interaction Dataset, ICPR contest on Semantic Description of Human Activities (SDHA). http://cvrc.ece.utexas.edu/SDHA2010/Human_Interaction.html, 2010.

[105] Ahmed Sabir, Francesc Moreno-Noguer, and Lluís Padró. Enhancing text spotting with a language model and visual context information. *ACM Special Interest Group on Information Retrieval Forum*, 2018.

[106] AC Seymour, J Dale, M Hammill, PN Halpin, and DW Johnston. Automated detection and enumeration of marine wildlife using unmanned aircraft systems (uas) and thermal imagery. *Scientific reports*, 7(1):1–10, 2017.

[107] M Sharma, D Rasmuson, B Rieger, D Kjelkerud, et al. Labelbox: The best way to create and manage training data. *https://www.labelbox.com*, 2019.

[108] Ya-Fang Shih, Yang-Ming Yeh, Yen-Yu Lin, Ming-Fang Weng, Yi-Chang Lu, and Yung-Yu Chuang. Deep co-occurrence feature learning for visual object recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4123–4132, 2017.

[109] Jamie Shotton, John Winn, Carsten Rother, and Antonio Criminisi. Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. *International Journal of Computer Vision*, 81(1):2–23, 2009.

[110] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[111] Amit Singhal, Jiebo Luo, and Weiyu Zhu. Probabilistic spatial context models for scene content understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, volume 1, pages I–I, 2003.

[112] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

[113] Thomas M Strat and Martin A Fischler. Context-based vision: recognizing objects using information from both 2 d and 3 d imagery. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(10):1050–1065, 1991.

[114] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

[115] Jin Sun and David W Jacobs. 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, 2017.

[116] Yi Sun, Xiaogang Wang, and Xiaoou Tang. Deep learning face representation from predicting 10,000 classes. In *Proceedings of the*

*IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1891–1898, 2014.

[117] Alexandra Swanson, Margaret Kosmala, Chris Lintott, Robert Simpson, Arfon Smith, and Craig Packer. Snapshot serengeti, high-frequency annotated camera trap images of 40 mammalian species in an african savanna. *Scientific Data*, 2(1):1–14, 2015.

[118] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.

[119] Kaihua Tang, Hanwang Zhang, Baoyuan Wu, Wenhan Luo, and Wei Liu. Learning to compose dynamic tree structures for visual contexts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6619–6628, 2019.

[120] Yapeng Tian, Jing Shi, Bochen Li, Zhiyao Duan, and Chenliang Xu. Audio-visual event localization in unconstrained videos. In *Proceedings of the European Conference on Computer Vision*, pages 247–263, 2018.

[121] Antonio Torralba. Contextual priming for object detection. *International Journal of Computer Vision*, 53(2):169–191, 2003.

[122] Antonio Torralba, Kevin P Murphy, and William T Freeman. Using the forest to see the trees: exploiting context for visual object detection and localization. *Communications of the ACM*, 53(3):107–114, 2010.

[123] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497, 2015.

[124] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8769–8778, 2018.

[125] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the Conference on Neural Information Processing Systems*, 2017.

[126] Huiyu Wang, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. MaX-DeepLab: End-to-end panoptic segmentation with mask transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[127] Huiyu Wang, Yukun Zhu, Bradley Green, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. Axial-DeepLab: Stand-alone axial-attention for panoptic segmentation. In *Proceedings of the European Conference on Computer Vision*, 2020.

[128] Jingwen Wang, Lin Ma, and Wenhao Jiang. Temporally grounding language queries in videos by contextual boundary-aware prediction.

In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 12168–12175, 2020.

[129] Kai Wang, Boris Babenko, and Serge Belongie. End-to-end scene text recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1457–1464, 2011.

[130] Xiaogang Wang, Gianfranco Doretto, Thomas Sebastian, Jens Rittscher, and Peter Tu. Shape and appearance context modeling. In *Proceedings of the International Conference on Computer Vision*, pages 1–8, 2007.

[131] Xiaoyang Wang and Qiang Ji. Incorporating contextual knowledge to dynamic bayesian networks for event recognition. In *Proceedings of the International Conference on Pattern Recognition*, pages 3378–3381, 2012.

[132] Xiaoyang Wang and Qiang Ji. Video event recognition with deep hierarchical context model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4418–4427, 2015.

[133] Xiaoyang Wang and Qiang Ji. Hierarchical context modeling for video event recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(9):1770–1782, 2016.

[134] Xuan Wang, Jiajun Chen, Hao Tang, and Zhigang Zhu. Multiclu: Multi-stage context learning and utilization for storefront accessibility detection and evaluation. In *Proceedings of the International Conference on Multimedia Retrieval*, page 304–312, 2022.

[135] Xuan Wang, Jiawei Liu, Hao Tang, Zhigang Zhu, and William Seiple. An ai-enabled annotation platform for storefront accessibility and localization. *Journal on Technology and Persons with Disabilities*, 2023.

[136] Xuan Wang, Hao Tang, and Zhigang Zhu. A general context learning and reasoning framework for object detection in urban scenes. In *Proceedings of the International Conference on Computer Vision Theory and Applications*, 2023.

[137] Xuan Wang, Hao Tang, and Zhigang Zhu. Gmc: A general framework of multi-stage context learning and utilization for visual detection tasks. *Computer Vision and Image Understanding*, 2023.

[138] Xuan Wang and Zhigang Zhu. Context understanding in computer vision: A survey. *Computer Vision and Image Understanding*, 2023.

[139] Galen Weld, Esther Jang, Anthony Li, Aileen Zeng, Kurtis Heimerl, and Jon E Froehlich. Deep learning for automatically detecting sidewalk accessibility problems using streetscape imagery. In *Proceedings of the International ACM SIGACCESS Conference on Computers and Accessibility*, pages 196–209, 2019.

[140] Lior Wolf and Stanley Bileschi. A critical view of context. *International Journal of Computer Vision*, 69(2):251–261, 2006.

[141] Jialian Wu, Chunluan Zhou, Ming Yang, Qian Zhang, Yuan Li, and Junsong Yuan. Temporal-context enhanced detection of heavily occluded pedestrians. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[142] Jialian Wu, Chunluan Zhou, Qian Zhang, Ming Yang, and Junsong Yuan. Self-mimic learning for small-scale pedestrian detection. In *Proceedings of the ACM International Conference on Multimedia*, page 2012–2020, 2020.

[143] Tong Xiao, Shuang Li, Bochao Wang, Liang Lin, and Xiaogang Wang. Joint detection and identification feature learning for person search. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3415–3424, 2017.

[144] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in Neural Information Processing Systems*, 34:12077–12090, 2021.

[145] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M. Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. In *Proceedings of the Conference on Neural Information Processing Systems*, 2021.

[146] Yuanjun Xiong, Kai Zhu, Dahua Lin, and Xiaoou Tang. Recognize complex events from static images by fusing deep channels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1600–1609, 2015.

[147] Yuwen Xiong, Renjie Liao, Hengshuang Zhao, Rui Hu, Min Bai, Ersin Yumer, and Raquel Urtasun. Upsnet: A unified panoptic segmentation

network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[148] Bingjie Xu, Junnan Li, Yongkang Wong, Qi Zhao, and Mohan S Kankanhalli. Interact as you intend: Intention-driven human-object interaction detection. *IEEE Transactions on Multimedia*, 22(6):1423–1432, 2019.

[149] Danfei Xu, Yuke Zhu, Christopher B Choy, and Li Fei-Fei. Scene graph generation by iterative message passing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5410–5419, 2017.

[150] Hang Xu, ChenHan Jiang, Xiaodan Liang, Liang Lin, and Zhenguo Li. Reasoning-rcnn: Unifying adaptive global reasoning into large-scale object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6419–6428, 2019.

[151] Yichao Yan, Qiang Zhang, Bingbing Ni, Wendong Zhang, Minghao Xu, and Xiaokang Yang. Learning context graph for person search. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2158–2167, 2019.

[152] Jianwei Yang, Jiasen Lu, Stefan Lee, Dhruv Batra, and Devi Parikh. Graph r-cnn for scene graph generation. In *Proceedings of the European Conference on Computer Vision*, pages 670–685, 2018.

[153] Shuo Yang, Ping Luo, Chen-Change Loy, and Xiaoou Tang. From facial parts responses to face detection: A deep learning approach. In

*Proceedings of the IEEE International Conference on Computer Vision*, pages 3676–3684, 2015.

[154] Shuo Yang, Ping Luo, Chen-Change Loy, and Xiaoou Tang. Wider face: A face detection benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5525–5533, 2016.

[155] Tien-Ju Yang, Maxwell D. Collins, Yukun Zhu, Jyh-Jing Hwang, Ting Liu, Xiao Zhang, Vivienne Sze, George Papandreou, and Liang-Chieh Chen. Deeperlab: Single-shot image parser. *ArXiv*, abs/1902.05093, 2019.

[156] Xitong Yang, Xiaodong Yang, Ming-Yu Liu, Fanyi Xiao, Larry S Davis, and Jan Kautz. 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, 2019.

[157] Yibo Yang, Hongyang Li, Xia Li, Qijie Zhao, Jianlong Wu, and Zhouchen Lin. Sognet: Scene overlap graph network for panoptic segmentation. *ArXiv*, abs/1911.07527, 2019.

[158] Bangpeng Yao and Li Fei-Fei. Modeling mutual context of object and human pose in human-object interaction activities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17–24, 2010.

[159] Deli Yu, Xuan Li, Chengquan Zhang, Tao Liu, Junyu Han, Jingtuo Liu, and Errui Ding. Towards accurate scene text recognition with semantic

reasoning networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12113–12122, 2020.

[160] Qihang Yu, Huiyu Wang, Dahun Kim, Siyuan Qiao, Maxwell Collins, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. Cmt-deeplab: Clustering mask transformers for panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2560–2570, 2022.

[161] Qihang Yu, Huiyu Wang, Siyuan Qiao, Maxwell Collins, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. k-means mask transformer. In *Proceedings of the European Conference on Computer Vision*, 2022.

[162] Yitian Yuan, Lin Ma, Jingwen Wang, Wei Liu, and Wenwu Zhu. Semantic conditioned dynamic modulation for temporal sentence grounding in videos. *Advances in Neural Information Processing Systems*, 32, 2019.

[163] Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. Neural motifs: Scene graph parsing with global context. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5831–5840, 2018.

[164] Hang Zhang, Kristin J. Dana, Jianping Shi, Zhongyue Zhang, Xiaogang Wang, Ambrish Tyagi, and Amit Agrawal. Context encoding for semantic segmentation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7151–7160, 2018.

[165] Mengmi Zhang, Claire Tseng, and Gabriel Kreiman. Putting visual object recognition in context. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12985–12994, 2020.

[166] Shanghang Zhang, Guanhang Wu, Joao P Costeira, and Jose MF Moura. Understanding traffic density from large-scale web camera data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5898–5907, 2017.

[167] Shanshan Zhang, Rodrigo Benenson, and Bernt Schiele. Citypersons: A diverse dataset for pedestrian detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3213–3221, 2017.

[168] Wenwei Zhang, Jiangmiao Pang, Kai Chen, and Chen Change Loy. K-net: Towards unified image segmentation. *ArXiv*, abs/2106.14855, 2021.

[169] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6230–6239, 2016.

[170] Liang Zheng, Hengheng Zhang, Shaoyan Sun, Manmohan Chandraker, and Qi Tian. Person re-identification in the wild. *arXiv preprint arXiv:1604.02531*, 2016.

[171] Chunluan Zhou and Junsong Yuan. Bi-box regression for pedestrian detection and occlusion estimation. In *Proceedings of the European Conference on Computer Vision*, 2018.

[172] Anna Zhu, Renwu Gao, and Seiichi Uchida. Could scene context be beneficial for scene text detection? *Pattern Recognition*, 58:204–215, 2016.

[173] Chenchen Zhu, Fangyi Chen, Uzair Ahmed, Zhiqiang Shen, and Marios Savvides. 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, 2021.

[174] Yingying Zhu, Nandita M Nayak, and Amit K Roy-Chowdhury. Context-aware modeling and recognition of activities in video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2491–2498, 2013.