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

Xuan Wang
The Graduate Center, The City University of New York
New York, New York, USA
xwang4@gradcenter.cuny.edu

Jiajun Chen
Stony Brook University
Stony Brook, New York, USA
jiajun.chen.2@stonybrook.edu

Hao Tang
The Borough of Manhattan Community College, The City University of New York
New York, New York, USA
htang@bmcc.cuny.edu

Zhigang Zhu
The City College and The Graduate Center, The City University of New York
New York, New York, USA
zzhu@ccny.cuny.edu

ABSTRACT

In this work, a storefront accessibility image dataset is collected from Google street view and is labeled with three main objects for storefront accessibility: doors (for store entrances), doorknobs (for accessing the entrances) and stairs (for leading to the entrances). Then MultiCLU, a new multi-stage context learning and utilization approach, is proposed with the following four stages: Context in Labeling (CIL), Context in Training (CIT), Context in Detection (CID) and Context in Evaluation (CIE). The CIL stage automatically extends the label for each knob to include more local contextual information. In the CIT stage, a deep learning method is used to project the visual information extracted by a Faster R-CNN based object detector to semantic space generated by a Graph Convolutional Network. The CID stage uses the spatial relation reasoning between categories to refine the confidence score. Finally in the CIE stage, a new loose evaluation metric for storefront accessibility, especially for knob category, is proposed to efficiently help BLV users to find estimated knob locations. Our experiment results show that the proposed MultiCLU framework can achieve significantly better performance than the baseline detector using Faster R-CNN, with +13.4% on mAP and +15.8% on recall, respectively. Our new evaluation metric also introduces a new way to evaluate storefront accessibility objects, which could benefit BLV group in real life.

CCS CONCEPTS

• **Computing methodologies** → **Computer vision**; *Deep Learning*.

KEYWORDS

Object Detection, Context Learning, Convolutional Neural Networks, Graph Convolutional Network, Knowledge Graph

ICMR '22, June 27–30, 2022, Newark, NJ, USA

ACM ISBN 978-1-4503-9238-9/22/06…$15.00
https://doi.org/10.1145/3512527.3531361

ACM Reference Format:
Xuan Wang, Jiajun Chen, Hao Tang, and Zhigang Zhu. 2022. MultiCLU: Multi-stage Context Learning and Utilization for Storefront Accessibility Detection and Evaluation. In *Proceedings of the 2022 International Conference on Multimedia Retrieval (ICMR '22), June 27–30, 2022, Newark, NJ, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3512527.3531361

1 INTRODUCTION

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 [8] 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 [41], 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 1). But the object will have higher chance to be recognized as a knob if we consider its context (e.g. the door)

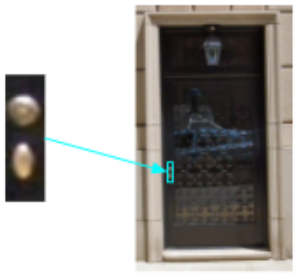

Figure 1: An example of the importance of contextual information for small object - the doorknob.

where it is located. Related works [21, 22] 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, 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 [5]. In order to align the visual context with semantic context in our machine learning model training task, we employed a Graph Convolution Network [19] 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.

How can we measure the accuracy of an object detector? The most common way is to use the intersection over union (IoU). In a object detection task, IoU measures the overlapping area between the predicted bounding box and the ground-truth bounding box of an object. Sometimes the IoU accuracy is not necessarily equivalent to the "accuracy" in real world. For example, when a BLV user tries to open a door, they may prefer to know where the knob's approximate location is (e.g., left middle of the door or right side of the door), rather than to provide the exact location (1.5m height and 20cm from left of the door). For this real world application, we further introduce a relaxed criteria to evaluate a doorknob on the door, in a way that can benefit BLV people in real life.

In summary, we collected SAI - a storefront accessibility image dataset using Google StreetView API and labeled three categories: doors, doorknobs and stairs. We proposed MultiCLU: a multi-stage context learning and utilization framework to detect storefront accessibility objects. Our MultiCLU approach is a unified framework that includes four consecutive stages of context learning and utilization: Context in Labeling (CIL) is mainly applied to small object categories such as doorknobs in the labeling stage, Context in Training (CIT) to model semantic context in the training stage, Context in Detection (CID) to utilize spatial contact in the detection stage, and finally Context in Evaluation(CIE) in redefining the evaluation of detection for practical applications. The main contributions of this paper are:

- A storefront accessibility image dataset is collected at street level scene, which contains three main categories: doors, doorknobs and stairs.
- A unified method is proposed for multi-stage context learning and utilization to employ local visual context, semantic context, spatial context, and application context information into one single framework.
- Our proposed method achieved significantly better performance over a standard end-to-end object detector with both individual and combined context information in the four stages.
- A new evaluation criteria is introduced for a real-world application, such as the storefront accessibility detection task, which could better benefit not only BLV people, but also people with other disabilities.

The paper is organized as follows. Section 2 discusses related work. Section 3 discusses how we collected and processed our dataset. Section 4 proposes our mutli-stage context learning and utilization framework and describes each contextual component in detail. Section 5 presents experimental results on our collected dataset and the ablation study for each component. Section 6 provides a few concluding remarks.

2 RELATED WORK

2.1 Context Understanding

Humans use visual context effortlessly to perceive the real world. An object hanging on the wall is probably a painting, not a car. A doorknob should be within the frame of a door, not on the ground. Contextual information provides critical information to help us visually find and recognize objects faster and more accurately. Not only in human perception, contextual information also plays an important role in many computer vision tasks, such as object detection [10, 12, 40, 49, 50], video event recognition [42, 43], video action detection [47, 51], scene graph generation [45, 48], data augmentation [11], image classification [27], and image inpainting [33]. In these tasks, different forms of contextual information have been employed. The contextual information used in the literature includes: global context [48], local neighborhood context [10, 11, 33], prior semantic knowledge [42, 43], geographic information [27], spatial relation between objects [40, 45, 47, 48] and temporal information [42, 43, 47, 51]. In [11], the author shows that the visual context surrounding objects is crucial to predict the presence of objects. Wang et al. [42, 43] introduces a hierarchical context model to recognize events in videos. Although contextual information has been used in different ways and gains more successes over context-free approaches, context could be misleading if an object present in irrelevant scenes. Choi et al. [7] present a context model for out-of-context detection, where the object is unusual for a given scene in a image. In our work, objects (doors, doorknobs and stairs) in our collected data is highly correlated, our proposed method

makes use of various contextual information by applying a unified multi-stage framework in context learning and utilization from data labeling, model training, to object detection and result evaluation.

2.2 Object Detection in Urban Scenes

Many object detection approaches have been proposed in urban scenes. Several works [10, 36, 49] focus on text detection and recognition in street level imagery and urban signage. [3] proposes an approach to mine existing spatial image databases for discover of zebra crosswalks in urban settings in order to increasing safety for blind travelers. [6] introduces a curb detection paradigm for road and sidewalk detection for mobile robots using stereo vision in the urban residential region. Another work [40] uses pair-wise existence of curb ramps: curb ramps usually appears in pairs in common sense, to find missing curb ramps at city street regions, which could help millions of people with mobility disabilities. Weld et al. [44] provide a method to automatically assess sidewalks accessibility in Google Street View. To our best knowledge, none of these works explicitly explore context information for the given tasks, and few studies have been done to detect storefront accessibility using street level imagery. In this paper, we focus on the use of context information in detecting storefront accessibility in urban scenes.

2.3 Accessibility Data Collection and Analysis

Sighted people can navigate to the destination using current navigation platform like Google Map without much difficulties. However, in order to facilitate the mobility and independence of people with disabilities, including blind, low vision and mobility disabilities, accessibility data needs to be collected on street crossings, sidewalks and transportation centers, etc. This kind of data is often required for uses by local government agencies [29, 30], but navigation platforms like Google Map do not include this data because of the lack of wide availability. Another challenge is that it is hard to obtain the complete city-scale data. One of the approaches to collect accessibility data is using crowdsourcing method. Compare to other approaches, such as sending people on site or hiring human collectors with heavy cost of labors, crowdsourcing provides a more efficient and cost-effective data collection approach. Many works [20, 28, 31, 37, 39] have shown the successful usage of crowdsourcing for transit and infrastructure. Other works combine Google Street View and crowdsourcing method for collecting street-level accessibility data of sidewalk issues [16], street crossings [14] and bus stop locations [15]. Our group are developing a crowd-sourcing based storefront image collection app [2, 25] for future use, but at the current stage, we manually collected a dataset called SAI - Storefront Accessibility Image dataset for performing context exploration.

3 SAI DATASET DESCRIPTION

The storefront accessibility image dataset (Fig.2) is collected from Google Street View of New York city using Google Street View API [13]. We then use [4] 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.

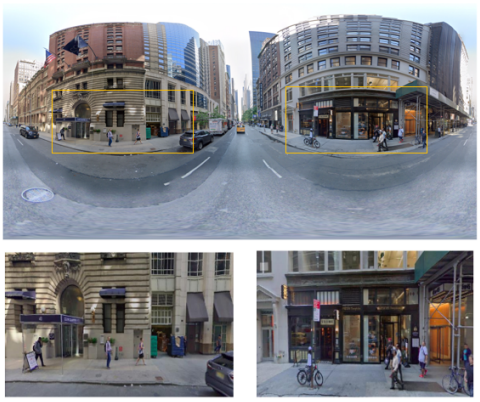

Figure 2: A formed panorama image and the cropped sub-images from Google Street View API of New York City. Top: The panorama image with all tiles. Bottom: cropped images from the middle 5x7 tiles as labeled in the panoramic image.

Figure 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 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[38]. 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 1. Examples of labeled storefront objects in an image is shown in Fig. 3.

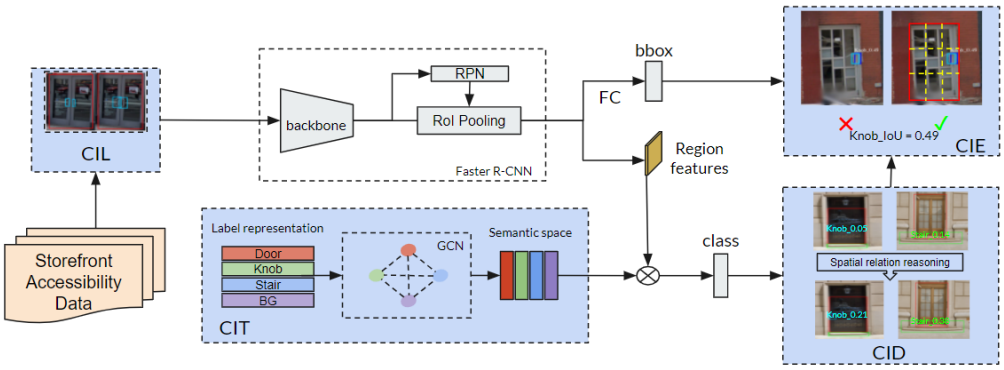

Figure 4: The architecture of our multi-stage context learning and utilization (MultiCLU) framework. Contextual components (in four stages) are shaded in light blue. CIL: Context in Labeling. CIT: Context in Training. CID: Context in Detection. CIE: Context in Evaluation. "⊗": dot product. "FC": Fully-connected layer.

4 PROPOSED METHOD

Our muti-stage context learning and utilization (MultiCLU) framework (Fig. 4) uses Faster R-CNN [35] 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 4.1) 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 4.2), we represent object labels using word embeddings extracted from a pretrained language model [34]. 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) [19] 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 space. Then in the CID stage (Section 4.3), 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 4.4) 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.

4.1 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 information 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

Figure 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.

we chose α = 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:

$$w'_{knob} = \begin{cases} \alpha w_{door}, & \text{if } w_{knob} < \alpha w_{door} \\ w_{knob}, & \text{otherwise} \end{cases} \quad (1)$$

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:

$$h'_{knob} = \begin{cases} \beta h_{door}, & \text{if } h_{knob} < \beta h_{door} \\ h_{knob}, & \text{otherwise} \end{cases} \quad (2)$$

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.1 and eq. 2. Restricting the new knob labels within the door frames is applied when extend original knob labels. Three examples are shown in Fig. 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.

4.2 Context in Training

4.2.1 Graph Convolutional Network. Kipf and Welling [19] 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 [5], visual relationship detection[9] and scene graph generation [18, 46], etc. As described in [19], 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:

$$H' = f(H, A) \tag{3}$$

After applying a convolution operation, the function in eq. 3 can be written as:

$$f(H, A) = \sigma(AHW) \tag{4}$$

where σ is a non-linear activation function and W is the weight.

4.2.2 Contextual Graph for GCN. As shown in Fig 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 [50], 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:

$$\mathbf{P} = softmax(H'_{labels} f_{regions}) \tag{5}$$

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 [5] 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. 6 shows the relation descriptor matrix generated from the SAI training dataset.

4.3 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. We 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.

	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 6: Relation descriptor matrix generated from the SAI training dataset.

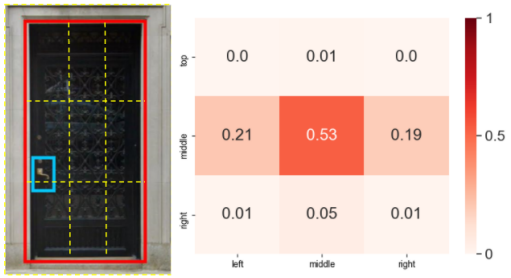

Figure 7: Knob probabilistic distribution inside a door frame using the training set. Left: 3x3 regions of a door. Right: 3x3 knob conditional probability distribution array.

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. 7). During detection, if a knob is predicted inside a predicted door (from the detector), the knob confidence score is updated as C'_{knob}:

$$C'_{knob} = \mu_1 C_{knob} + \mu_2 C_{knob|door} C_{door} \tag{6}$$

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. 7 right). We take the weighted average of the original prediction score (from the detector) and the "deduction" score (from the Bayesian deduction), where μ_1 and μ_2 are the weights applied to them, respectively.

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 disalignments between these two categories (see Fig 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:

$$S_{height} = height_{stair} + 0.2 height_{door} \tag{7}$$

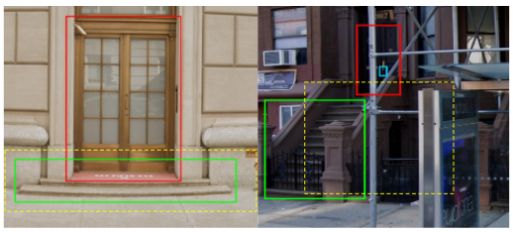

Figure 8: Two special cases for stair-door relations. left: door and stair have an overlapped area. Right: The stair is on the left bottom of the door due to camera perspective. Yellow dashed box: Search areas S.

and the width is defined as:

$$S_{width} = width_{stair} + width_{door} \tag{8}$$

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:

$$C'_{stair} = \alpha_1 C_{stair} + \alpha_2 C_{stair|door} C_{door} \tag{9}$$

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. α_1 and α_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:

$$C'_{door} = w_1 C_{door} + w_2 C_{door|knob} C_{knob} + w_3 C_{door|stair} C_{stair} \tag{10}$$

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.

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 6-10 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.

4.4 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:

$$IoU = \frac{\text{area}(B_{pred} \cap B_{gt})}{\text{area}(B_{pred} \cup B_{gt})} \tag{11}$$

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 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 9 when the IOU threshold is set to 0.5 (50% overlap between the prediction and the ground truth).

5 EXPERIMENTS

In this section, we first describe our MultiCLU model implementation details. We then compare various evaluation results. 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. Finally, we provide ablation analysis on the overall results and our multiCLU learning components.

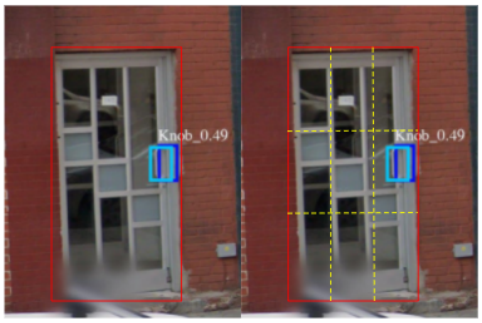

Figure 9: Comparison of the commonly used IoU evaluation and our Context in Evaluation (CIE) for knob. The IoU score for predicted knob is 0.49. Left: IoU at 0.5 threshold will treat it as false positive and not detected. Right: CIE uses door distributed regions to evaluate the knob, the predicted knob is accepted as a correct detection.

5.1 Implementation Details

Our proposed MultiCLU model is based on Faster R-CNN [35]. We adopt ResNet-50 [17] and Feature Pyramid Network [23] as the feature extraction backbone, which is pretrained on the Coco [24] dataset. For contextual graph learning in CIT, our GCN model consists of one layer with the output dimensionality of 1024. We use LeakyReLu [26] as the activation function for GCN model (σ in Section 4.2.1). For word representations, we use 300-dim word vectors (d is 300 in Section 4.2.2) extracted from the language model GloVe [34] pretrained on the Wikipedia dataset. Stochastic Gradient Descent (SGD) is used as the optimizer during training. The momentum and the weight decay to 0.95 and 0.0001, respectively. The initial learning rate is 0.005, which drops by 0.25 for every 8 epochs. The network is trained for 40 epochs in total. For the CID stage, without otherwise stated, we set μ_1, μ_2 to be 0.75, 0.25 in Eq. (6), α_1, α_2 to be 0.75, 0.25 in Eq. (9) and w_1, w_2, w_3 to be 0.7, 0.2, 0.1 in Eq. (10). Our MultiCLU is implemented in Pytorch [32].

5.2 Experimental Results

5.2.1 Comparison with Various Contextual Components. We compare our proposed multiCLU approach with various combinations of the contextual components to the baseline Faster R-CNN [35], and measure the overall recall and mean average precision (mAP) over a 0.5 IoU threshold. 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 2, 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

Table 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 [35]	-	-	-	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**

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%).

5.2.2 Comparison per Category. 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, 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 improvement 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, 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.

5.2.3 Context in Evaluation Comparison. 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 4.4). 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

Table 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

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, 25], 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.

5.3 Ablation Study

We further studied the contribution of each contextual component. As we applied Context in Labeling (CIL 4.1), Context in Training (CIT 4.2), Context in Detection (CID 4.3) to the baseline Faster R-CNN [35] with ResNet-50 [17] and FPN [23] as backbone, with various combinations, we can clearly see the contributions of each and the integration of them in Table 2 and Table 3.

First, CIL was applied before feeding the training images into the network. We used the ground-truth door labels that contain knobs as constraint to updated the knob labels automatically, using surrounding areas as local contextual information for network to learn. Our result shows that when there is enough local visual contextual information for knobs, the network can gain significant improvement on both mAP and recall, with 9.1% mAP and 11% recall gains over the baseline (Table 2), respectively. When we further analyzed the result in Table 3, we found that not only knob category achieved great improvement from using the local contextual information, the detection of door and stair categories also outperforms the baseline in both precision and recall, probably because fewer false positive and negative cases for the knob as other two categories help in improving the performance of these two classes.

During the training stage, we used word embeddings for each category and our contextual graph generated from prior semantic knowledge learnt from the training dataset as the input to Graph Convolutional Network [19], previously used in image recognition task [5], to construct the semantic space. We projected the region proposals extracted from Faster R-CNN [35] to semantic space to obtain the predicted score for each proposed region (Section 4.2.)

As the results shown in Table 2 and Table 3, CIT did improve the baseline, but not as effective as the CIL, which might due to that the semantic context information was not specific to individual objects but rather to object categories statistically. with a small dataset as the SAI, the improvement is not statically significant enough. With CID, we used specific spatial relation reasoning to refine the confidence score for each region proposal which satisfied certain criteria (Section 4.3) and filtered the low confidence predictions using the refined confidence score. The precision slightly decreases as recall improved for all three categories, as shown in Table 3. This is because the way we refined the confidence for overlapping objects, which could introduce more false positives than correct predictions, hence the precision could decrease while recall improved.

Our proposed method with all the first three components (CIL+CIT+CID) exhibited improvement over any other combinations. This shows that the proposed contextual components can benefit each other, hence maxing the performance over the baseline. Furthermore, we introduced an application-oriented evaluation metric for our real-world task - finding the rough location of a doorknob. The new metric used the door regions instead of the standard IoU evaluation (Section 4.4). We hope this proposed evaluation metric can provide a new way to think the object detection evaluation in specific task, which is not only to check the accuracy in a strict condition, but also to take into account what will be useful in real world applications.

6 CONCLUSION

In this work, 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 3 target categories: door, knob, 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. Furthermore, our proposed MultiCLU provide a generic pipeline of contextual learning in deep learning, from data preprocessing, training, post processing to result evaluation, which can be applied to a wide variety of objects in object detection task. The proposed MultiCLU also has the flexibility to use each contextual component individually and with various combinations, and this model would be readily applicable to other tasks. In addition as the fourth contextual component, we introduced a new evaluation metric for the knob category in our task, which could provide a new way to think the evaluation standard in real world applications.

ACKNOWLEDGMENTS

The research is supported by the National Science Foundation (NSF) through Awards #2131186 (CISE-MSI), #1827505 (PFI), and #1737533 (S&CC). The work is also supported by the US Air Force Office of Scientific Research (AFOSR) via Award #FA9550-21-1-0082 and the ODNI Intelligence Community Center for Academic Excellence (IC CAE) at Rutgers University (#HHM402-19-1-0003 and #HHM402-18-1-0007).

REFERENCES

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

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

[3] Dragan Ahmetovic, Roberto Manduchi, James M Coughlan, and Sergio Mascetti. 2015. Zebra Crossing Spotter: Automatic Population of Spatial Databases for Increased Safety of Blind Travelers. In *Proceedings of the International ACM SIGACCESS Conference on Computers and Accessibility*. 251–258.

[4] Marco Cavallo. 2015. 3D City Reconstruction From Google Street View.

[5] Zhao-Min Chen, Xiu-Shen Wei, Peng Wang, and Yanwen Guo. 2019. Multi-label Image Recognition with Graph Convolutional Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5177–5186.

[6] Mingmei Cheng, Yigong Zhang, Yingna Su, José Manuel Álvarez, and Hui Kong. 2018. Curb Detection for Road and Sidewalk Detection. *IEEE Transactions on Vehicular Technology* 67 (2018), 10330–10342.

[7] Myung Jin Choi, Antonio Torralba, and Alan S. Willsky. 2012. Context Models and Out-of-context Objects. *Pattern Recognition Letters* 33 (2012), 853–862.

[8] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. 2016. The Cityscapes Dataset for Semantic Urban Scene Understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

[9] Zhen Cui, Chunyan Xu, Wenming Zheng, and Jian Yang. 2018. Context-dependent Diffusion Network for Visual Relationship Detection. In *Proceedings of the ACM International Conference on Multimedia*. 1475–1482.

[10] Yuning Du, Genquan Duan, and Haizhou Ai. 2012. Context-based Text Detection in Natural Scenes. In *Proceedings of the IEEE International Conference on Image Processing*. IEEE, 1857–1860.

[11] Nikita Dvornik, Julien Mairal, and Cordelia Schmid. 2018. Modeling Visual Context is Key to Augmenting Object Detection Datasets. In *Proceedings of the European Conference on Computer Vision*. 364–380.

[12] Yuan Fang, Kingsley Kuan, Jie Lin, Cheston Tan, and Vijay Chandrasekhar. 2017. Object Detection Meets Knowledge Graphs. In *Proceedings of the International Joint Conferences on Artificial Intelligence*.

[13] Google. 2022. Google street View API. https://developers.google.com/maps/documentation/streetview/overview

[14] Richard Guy and Khai Truong. 2012. CrossingGuard: Exploring Information Content in Navigation Aids for Visually Impaired Pedestrians. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 405–414. https://doi.org/10.1145/2207676.2207733

[15] Kotaro Hara, Shiri Azenkot, Megan Campbell, Cynthia L. Bennett, Vicki Le, Sean Pannella, Robert Moore, Kelly Minckler, Rochelle H. Ng, and Jon E. Froehlich. 2015. Improving Public Transit Accessibility for Blind Riders by Crowdsourcing Bus Stop Landmark Locations with Google Street View: An Extended Analysis. *ACM Transactions on Accessible Computing* 6, 2 (2015). https://doi.org/10.1145/2717513

[16] Kotaro Hara, Vicki Le, and Jon Froehlich. 2013. Combining Crowdsourcing and Google Street View to Identify Street-Level Accessibility Problems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 631–640. https://doi.org/10.1145/2470654.2470744

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 770–778.

[18] Justin Johnson, Agrim Gupta, and Li Fei-Fei. 2018. Image Generation from Scene Graphs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1219–1228.

[19] Thomas N Kipf and Max Welling. 2016. Semi-Supervised Classification with Graph Convolutional Networks. *arXiv preprint arXiv:1609.02907* (2016).

[20] Daniel Krajzewicz, Jakob Erdmann, Michael Behrisch, and Laura Bieker-Walz. 2012. Recent Development and Applications of SUMO - Simulation of Urban MObility. *Int.J. Advances in Systems and Measurements* 3-4 (2012).

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

[22] Jeong-Seon Lim, Marcella Astrid, Hyun-Jin Yoon, and Seung-Ik Lee. 2021. Small Object Detection using Context and Attention. In *Proceedings of the International Conference on Artificial Intelligence in Information and Communication*. 181–186. https://doi.org/10.1109/ICAIIC51459.2021.9415217

[23] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. 2017. Feature Pyramid Networks for Object Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2117–2125.

[24] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft COCO: Common Objects in Context. In *Proceedings of the European Conference on Computer Vision*. 740–755.

[25] Jiawei Liu, Hao Tang, William Seiple, and Zhigang Zhu. 2022. Annotating Storefront Accessibility Data Using Crowdsourcing. *Journal on Technology and Persons with Disabilities* 10 (May 2022), 154–170.

[26] Andrew L. Maas, Awni Y. Hannun, and Andrew Y. Ng. 2013. Rectifier Nonlinearities Improve Neural Network Acoustic Models. In *Proceedings of the ICML Workshop on Deep Learning for Audio, Speech and Language Processing*.

[27] Oisin Mac Aodha, Elijah Cole, and Pietro Perona. 2019. Presence-only Geographical Priors for Fine-grained Image Classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 9596–9606.

[28] Gilberto Marzano, Joanna Lizut, and Luis Ochoa Siguencia. 2019. Crowdsourcing Solutions for Supporting Urban Mobility. *Procedia Computer Science* 149 (01 2019), 542–547. https://doi.org/10.1016/j.procs.2019.01.174

[29] DoITT MODA. 2017. NYC Open Data. https://opendata.cityofnewyork.us/

[30] NYDOT. 2022. New York City Department of Transportation. https://www1.nyc.gov/html/dot/html/home/home.shtml

[31] Claudio E. Palazzi and Armir Bujari. 2016. Fostering Accessible Urban Mobility through Smart Mobile Applications. In *Proceedings of the IEEE Annual Consumer Communications Networking Conference*. 1141–1145. https://doi.org/10.1109/CCNC.2016.7444950

[32] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. *Advances in Neural Information Processing Systems* 32 (2019), 8026–8037.

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

[34] Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global Vectors for Word Representation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. 1532–1543.

[35] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. 2015. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39 (2015), 1137–1149.

[36] Ahmed Sabir, Francesc Moreno-Noguer, and Lluís Padró. 2018. Enhancing Text Spotting with a Language Model and Visual Context Information. In *Proceedings of the Computer and Communications Industry Association*.

[37] José Salcedo and J. Octavio Gutierrez-Garcia. 2015. Crowdsourcing Information for Knowledge-based Design of Routes for Unscheduled Public Transport Trips. *Journal of Knowledge Management* 19 (05 2015). https://doi.org/10.1108/JKM-02-2015-0053

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

[39] Dongyoun Shin. 2016. Urban Sensing by Crowdsourcing: Analysing Urban Trip behaviour in Zurich. *Int.J. Urban and Regional Research* 40 (2016), 1044–1060.

[40] Jin Sun and David W Jacobs. 2017. Seeing What is Not There: Learning Context to Determine Where Objects are Missing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5716–5724.

[41] Kai Wang, Boris Babenko, and Serge Belongie. 2011. End-to-end Scene Text Recognition. In *Proceedings of the International Conference on Computer Vision*. IEEE, 1457–1464.

[42] Xiaoyang Wang and Qiang Ji. 2015. Video Event Recognition with Deep Hierarchical Context Model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4418–4427.

[43] Xiaoyang Wang and Qiang Ji. 2017. Hierarchical Context Modeling for Video Event Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39 9 (2017), 1770–1782.

[44] Galen Weld, Esther Jang, Anthony Li, Aileen Zeng, Kurtis Heimerl, and Jon E Froehlich. 2019. Deep Learning for Automatically Detecting Sidewalk Accessibility Problems Using Streetscape Imagery. In *Proceedings of the International ACM SIGACCESS Conference on Computers and Accessibility*. 196–209.

[45] Danfei Xu, Yuke Zhu, Christopher B Choy, and Li Fei-Fei. 2017. Scene Graph Generation by Iterative Message Passing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5410–5419.

[46] Jianwei Yang, Jiasen Lu, Stefan Lee, Dhruv Batra, and Devi Parikh. 2018. Graph R-CNN for Scene Graph Generation. In *Proceedings of the European Conference on Computer Vision*. 670–685.

[47] Xitong Yang, Xiaodong Yang, Ming-Yu Liu, Fanyi Xiao, Larry S Davis, and Jan Kautz. 2019. STEP: Spatio-Temporal Progressive Learning for Video Action Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 264–272.

[48] Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. 2018. Neural Motifs: Scene Graph Parsing with Global Context. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5831–5840.

[49] Anna Zhu, Renwu Gao, and Seiichi Uchida. 2016. Could Scene Context be Beneficial for Scene Text Detection? *Pattern Recognition* 58 (2016), 204–215.

[50] Chenchen Zhu, Fangyi Chen, Uzair Ahmed, Zhiqiang Shen, and Marios Savvides. 2021. Semantic Relation Reasoning for Shot-stable Dew-shot Object Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8782–8791.

[51] Yingying Zhu, Nandita M Nayak, and Amit K Roy-Chowdhury. 2013. Context-Aware Modeling and Recognition of Activities in Video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2491–2498.