GAR: Graph Assisted Reasoning for Object Detection

Zheng Li
Arizona State University
zhengl11@asu.edu

Xiaocong Du
Arizona State University
xiaocong@asu.edu

Yu Cao
Arizona State University
ycao@asu.edu

Abstract

It is well believed that object-object relations and object-scene relations inherently improve the accuracy of object detection. However, the way to efficiently model relations remains a problem. Graph Convolutional Network (GCN), an effective method to handle structured data with relations, inspires us to leverage graphs in modeling relations for objection detection tasks. In this work, we propose a novel approach, Graph Assisted Reasoning (GAR), to utilize a heterogeneous graph in modeling object-object relations and object-scene relations. GAR fuses the features from neighboring object nodes as well as scene nodes and produces better recognition than that produced from individual object nodes. Moreover, compared to previous approaches using Recurrent Neural Network (RNN), the light-weight and low-coupling architecture of GAR further facilitates its integration into the object detection module. Comprehensive experiments on PASCAL VOC and MS COCO datasets demonstrate the efficacy of GAR.

1. Introduction

Recently, significant development of object detection has been witnessed due to the advance in deep Convolutional Neural Networks (CNNs) [25, 39, 18]. The current object detection methods [10, 15, 17, 37] mostly follow the philosophy of anchor or region proposal introduced by R-CNN [16]. In these approaches, object classification and bounding box (bbox) regression are performed either on explicitly selected proposals that are generated from predefined anchors [10, 37], or directly on anchors (or prior boxes) [29, 36, 27]. Besides, anchor-free methods are also a set of emerging solutions that achieve admirable performance on the detection of multi-scale and heavily occluded objects [22, 24].

Contextual information including scene context and object relationships plays a critical role in humans' capability of recognizing objects, revealed by psychological investigations [3, 34]. Take Figure 1a as an example. There are two cows on the grassland. Implied by the blue sky and shadows, the scene is outdoor rather than in a room with green mat. The white objects on the grass can be recognized as cows, or closely, goats. However, the white cloud cannot be wrongly detected as cows or sheeps. The most obvious reason is that it is in sky. Figure 1b is another good example. The man-made object is probably a ship instead of car or train since it is on the water. Studies in the computer vision community are also conducted to boost performance of object detection by utilizing contextual information. For instance, previous studies [8, 12, 15, 32, 33] suggest that one can leverage the modeling of implicit context or explicit relation in recognition algorithms. Specifically, what categories of objects may appear in a specific scene, or what category of objects may appear simultaneously with another category of objects. However, most of the methods were proposed before the popularity of deep learning and have not been well explored in modern CNNs on object detection tasks. One of the challenges in relation modeling is the computation complexity due to the significant variations in the quantity and category of objects across different images. Another challenge is to efficiently encode and process the object relation into a CNN-based object detector.

(a) (b)

Figure 1. (a) Two cows on the grassland. (b) A ship at the seaside.

In the recent past, Graph Convolutional Network (GCN) has been successfully applied to node classification on graph-structured data [23], such as the citation network, text classification [41] and some other Natural Language Processing (NLP) tasks [31, 4]. By aggregating information from neighboring nodes, GCN produces better inference than merely taking the features from an individual node [23]. This intrinsic property of GCN provides it unique advantages in handling entities with relations.

Motivated by the property of GCN, we propose a novel approach, namely Graph Assisted Reasoning (GAR), to improve the efficacy of object detection. In the graph of GAR, the *object nodes* are regional features generated by Regional Proposal Network (RPN) and Region of Interest (ROI) pooling, and the *scene nodes* are learned embeddings from the entire image features. The *edge* between two object nodes is created with *object-object co-occurrence*, while the *edge* between an object node and a scene node is built with *object-scene co-occurrence*. Then GAR generates the node scores, which act as regularizing items for the basic one-layer object classifier, suppressing abnormal object candidates and amplifying probable ones, thus leading to more reliable object detection.

To summarize, our contributions are as follows:

(1) We propose a novel graph-assisted reasoning approach, GAR, that leverages GCN for object detection.

(2) To the best of our knowledge, this is the first study modeling object proposals and scene features as *nodes* in an heterogeneous graph, object-object relation and object-scene relation as *edges*, converting object detection from a perception problem to a reasoning problem.

(3) The proposed GAR is an extendable scheme that encodes relations into an adjacent matrix for object detection. Besides the co-occurrence relation, other relations such as spatial relations and higher level semantic relations can also be incorporated into the GAR architecture.

2. Related work

2.1. Contextual Methods for Object Detection

Prior to the emergence of deep learning, various approaches have explored adding contextual information to improve object detection [2, 19, 33, 40, 12]. In [15], the detected objects are re-scored by considering object relationships such as co-occurrence, which implies how likely two categories of objects can exist in the same image. On the contrary, the presence of objects in irrelevant scenes is penalized in [40]. These methods achieved moderate success in pre-deep learning era but have not been well established for deep CNNs. One of the possible reasons is that deep CNNs generally convey implicitly and hidden contextual information which is hard to use directly. Another reason is that to accommodate the contextual information within CNNs is a complicated and nontrivial work.

Recently, some approaches [5, 38, 42] based on deep CNNs have made attempts to incorporate contextual information into object detection. The work ION [5] integrates contextual information outside the ROI using a spatial RNN. GBD-Net [42] proposes a gated bi-directional CNN to pass messages between the features of different support regions around objects. Shrivastava *et al*. [38] use segmentation to provide top-down context to guide region proposal generation and object detection.

Despite the aforementioned approaches that essentially exploit local context near objects and the whole image context, Chen *et al*. [9] propose a sequential reasoning architecture that mainly utilizes object-object relationship to detect objects in an image sequentially. Similarly, Hu *et al*. [21] introduce attention modules to model object-object relations. Combining both object-object and object-scene relations, SIN [30] uses Gated Recurrent Unit (GRU) for message passing. Different from the existing methods, the proposed GAR adopts a light-weight GCN with an explicit and accurate scene detection module. This property further improves the efficiency of GAR in object detection with both object-object relation and object-scene relation considered.

2.2. Graph on Neural Networks

The topic of Graph Neural Networks (GNN) has received growing attention recently [7, 6]. Kipf *et al*. [23] presents a simplified yet well-behaved GNN model, i.e., GCN, which achieves state-of-the-art classification results on several benchmark graph datasets. GCN is then explored in several NLP tasks such as semantic role labeling [31] and machine translation [4] to encode the syntactic structure of sentences.

In [11], a document or a sentence is treated as a graph of word nodes, and GCN-Text [41] regards both the documents and words as nodes and constructs the corpus graph. In our work, proposal features and learned scene embeddings form the nodes of the heterogeneous graph, and the co-occurrence that is appropriately processed embodies the edges of the graph in GAR.

3. Approach

Our approach is designed upon the heterogeneous graph composed by object-object subgraph and object-scene subgraph. In this section, we will firstly derive the edges, i.e., relations of contextual information. Then we will show the relation graphs computing flow within the entire object detection network, as shown in Figure 2.

3.1. Relation Modeling

Adopting the same spirit as GCN-Text [41], we use co-occurrence to encode the relations among objects and scenes.

There are four relations involved in GAR: (i) object-object, estimating the probability of two different categories of objects appear in the same image; (ii) object-indoor/outdoor, measuring the frequency of all types of objects appear in the indoor/outdoor scenario; (iii) object-place, wherein *place* represents place categories such as "living room", "museum", etc.; and (iv) object-attribute, wherein *attribute* represents scene attribute such as "natural light", "human-made", etc. Scene-scene relations are not required

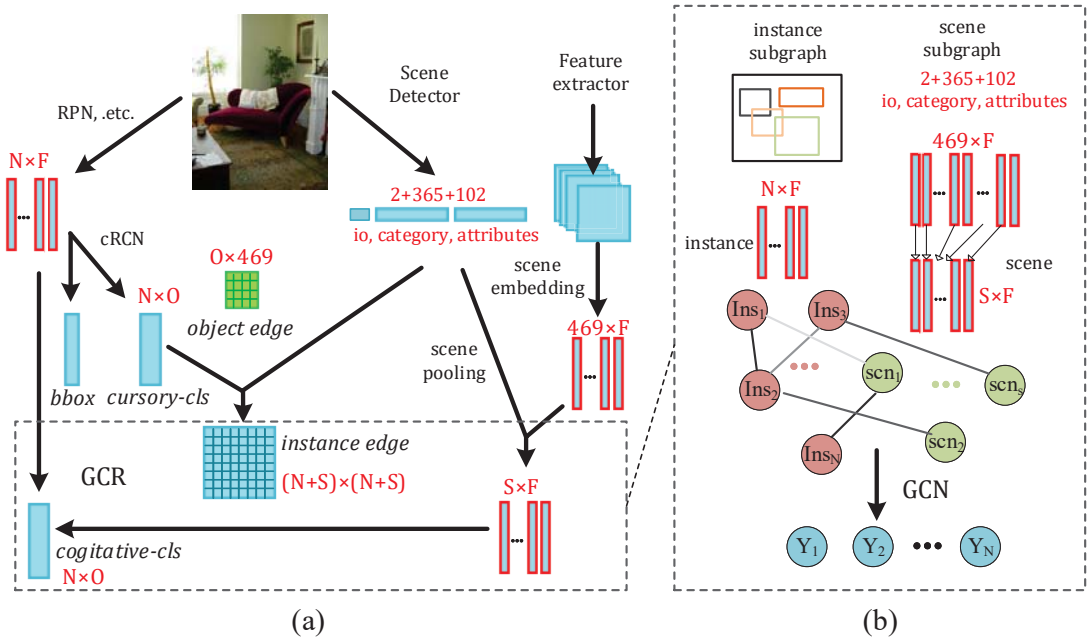

Figure 2. (a) The architecture of GAR. RPN and associated modules give N proposals (each with a size of F). The cRCN generates N cursory class scores (each with a size of O) and N bbox regression with a size of 4. Meanwhile, the scene detector generates scene-related labels with a size of 469. Scene pooling picks and concatenates S scene nodes from 469 of them, which are generated from scene embedding module. (b) The GCR module merges the instance subgraph and the scene subgraph as a heterogenous graph. *Nodes* are proposal features $ins_1, \cdots, ins_N$ and selected scene embeddings $scn_1, \cdots, scn_S$. *Edges* are created from the normalized co-occurrence matrix that are elaborated in the following section.

since GAR is object detection oriented. As classic datasets in object detection are typically lack of scene labels, we train a scene detector on the Place365 [43] recognition dataset to extract scene information for generating co-occurrence matrices for object-scene relations, i.e., (ii), (iii) and (iv). The above four relations are elaborated as follows:

Object-object relation The value of each co-occurrence entry represents the co-occurrence number enumerating the entire training images. Multiple occurrences of objects with the same class label in a single image are counted as 1. Calculation of the 2D $O \times O$ object-object co-occurrence matrix $\boldsymbol{\mathcal{E}}_{obj}^{obj}$ can be formulated as:

$$\boldsymbol{\mathcal{E}}_{obj}^{obj}(i,j) = \sum_{x=1}^{M} \begin{cases} 1, & \text{if } \mathrm{DET}_x(i) \ \& \ \mathrm{DET}_x(j) \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

where $\mathrm{DET}_x(i)$ means that the x-th training sample contains the object(s) with class index i. "&" means its left event and its right event happen at the same time. M represents the size of the training set and O is the number of classes in the selected dataset. It is worth noting that diagonal entries (self-loop) of $\boldsymbol{\mathcal{E}}_{obj}^{obj}$ are 0 instead of 1, as self-loop information is adaptively learned in GAR. The cumulative co-occurrence is normalized within GAR computing which is described in the following content.

Object-indoor/outdoor relation Indoor/outdoor is a binary label for an image. The output of the scene detector is composed of "indoor/outdoor" label (scalar), place categories (a vector containing 365 elements) and scene attributes (a vector containing 102 elements).

The calculation of the $O \times 2$ object-indoor/outdoor co-occurrence $\boldsymbol{\mathcal{E}}_{obj}^{io}$ is:

$$\boldsymbol{\mathcal{E}}_{obj}^{io}(\mathbf{i},:) = \sum_{x=1}^{M} \begin{cases} [1 \ \ 0], & \text{if } \mathrm{DET}_x(\mathbf{i}) \ \& \ \mathrm{INOUT}_x(indoor) \\ [0 \ \ 1], & \text{if } \mathrm{DET}_x(\mathbf{i}) \ \& \ \mathrm{INOUT}_x(outdoor) \end{cases} \tag{2}$$

Where $\mathrm{INOUT}_x(indoor)$ means that the image x is classified as "indoor". $\mathrm{DET}_x(\mathbf{i})$ is a vector composed by class indices of all objects detected in image x.

Object-place relation The scene detector infers place labels among 365 place categories. We will calculate the $O \times 365$ object-place co-occurrence $\boldsymbol{\mathcal{E}}_{obj}^{plc}$ by:

Algorithm 1: cls-roi edge

Data: co-occurance matrix $\boldsymbol{\mathcal{E}_{obj}^{obj}}$
cursory instance class score $\boldsymbol{Ins_{score}}$
self-loop edges for instances A
Result: Instance relation edges $\boldsymbol{\mathcal{E}_{ins}^{ins}}$

Softmax $\boldsymbol{\mathcal{E}_{obj}^{obj}}$ in a row-wise manner;
for *every instance pair: {i, j}* **do**
  get cursory instance class label:
    $cls^i = \arg\max_{k \in O} \{Ins_{score}^{i,k}\}$;
    $cls^j = \arg\max_{k \in O} \{Ins_{score}^{j,k}\}$;
  get instance relation:
    $\mathcal{E}_{ins}^{i,j} = \boldsymbol{\mathcal{E}_{obj}^{obj}}(cls^i, cls^j)$;
  add self-loop edge:
    $\mathcal{E}_{ins}^{i,i} = \mathcal{E}_{ins}^{i,i} + A(i), \forall\, i \in [0, \mathrm{N})$.
end
$\widetilde{\boldsymbol{\mathcal{E}}}_{\boldsymbol{ins}}^{\boldsymbol{ins}}$ = Softmax $\boldsymbol{\mathcal{E}_{roi}}$ in a row-wise manner;

Algorithm 2: scene-roi edge

Data: scenic co-occurrence matrices $\boldsymbol{\mathcal{E}_{obj}^{io}}$, $\boldsymbol{\mathcal{E}_{obj}^{plc}}$, $\boldsymbol{\mathcal{E}_{obj}^{attr}}$,
cursory ROI class score $\boldsymbol{ROI_{score}}$,
indoor/outdoor score $\boldsymbol{In_{score}}$/ $\boldsymbol{Out_{score}}$,
place category scores $\boldsymbol{Plc_{score}}$,
attribute scores $\boldsymbol{Attr_{score}}$
Result: scene-ROI relation graph $\boldsymbol{\mathcal{E}_{scene-roi}}$

Softmax co-occurrence matrices in a row-wise manner;
for *every ROI pair: {i, j}* **do**
  get ROI cls:
    $R_{cls}^i = \arg\max_{k \in K} \{ROI_{score}^{i,k}\}$;
    $R_{cls}^j = \arg\max_{k \in K} \{ROI_{score}^{j,k}\}$;
  get ROI relation:
    $\mathcal{E}_{roi}^{i,j} = ROI_{score}^{R_{cls}^i, R_{cls}^j}$
end
Softmax $\boldsymbol{\mathcal{E}_{roi}}$ in a row-wise manner;

$$\boldsymbol{\mathcal{E}_{obj}^{plc}}(\mathbf{i}, \mathbf{p}) = \sum_{x=1}^{M} \begin{cases} \mathbf{1}, & \text{if DET}_x(\mathbf{i}) \ \&\ \text{PLACE}_x(\mathbf{p}) \\ \mathbf{0}, & \text{otherwise} \end{cases} \quad (3)$$

where $\text{PLACE}_x(\mathbf{p})$ means that place labels with indices $\mathbf{p}$ are detected in the image x. Multiple place categories are taken since they could be synonyms in the sense of "scene", sharing the similar scene context.

Object-attribute relation The scene attributes are also generated by the scene detector among 102 classes. The $O \times 102$ object-attribute co-occurrence $\boldsymbol{\mathcal{E}_{obj}^{atr}}$ is calculated by:

$$\boldsymbol{\mathcal{E}_{obj}^{atr}}(\mathbf{i}, \mathbf{q}) = \sum_{x=1}^{M} \begin{cases} \mathbf{1}, & \text{if DET}_x(\mathbf{i}) \ \&\ \text{ATTR}_x(\mathbf{q}) \\ \mathbf{0}, & \text{otherwise} \end{cases} \quad (4)$$

Similarly, $\text{ATTR}_x(\mathbf{q})$ means scene attribute labels with indices $\mathbf{q}$ are detected in image x.

3.2. GAR Design

Different from existing works that use implicit visual appearance context, GAR is designed to make use of explicit object-object/scene relation to reward or penalize object proposals and thus assist object detection.

GAR is composed of four major modules: (i) a backbone object detector that generates object proposals, (ii) a scene detector that generates scene labels, (iii) a cursory Regression and Classification Network (cRCN) that returns the cursory detection scores as well as a spatial adjustment vector for each object proposal, and (iv) Graph Convolutional Reasoning (GCR) module which takes cursory detection, object/scene features (nodes) and prior relation knowledge (edges) as inputs and generates the graph reasoning scores. The entire framework of GAR is illustrated in Figure 2.

GAR is a general method. In this work, we use Faster R-CNN [37] as the backbone object detector for demonstration purpose. Other CNN-based detectors are also compatible with it.

Object edges to instance edges In Faster R-CNN, thousands of region proposals that might contain objects are obtained after Region Proposal Network (RPN). Non-Maximum Suppression (NMS) [14] is then used to select a fixed number (e.g., N=300) of ROIs. Next, for each ROI i, its visual feature v_i is processed by the ROI pooling and a fully connected projection layer. Consequently, the instance feature matrix $\mathcal{V}_{\mathbf{roi}}$ concatenated by all N ROI vectors is fed into cRCN to get N cursory class scores and bbox adjustment.

The $N \times N$ relation edges among N instances, $\boldsymbol{\mathcal{E}_{ins}^{ins}}$, is obtained by utilizing the prior $O \times O$ object-object co-occurrence $\boldsymbol{\mathcal{E}_{obj}^{obj}}$ and the cursory detection score, as shown in Algorithm 1.

Scene nodes and instance-scene relation In GAR, instance nodes of the GCR module input are the instance feature matrix $\mathcal{V}_{\mathbf{ins}}$ in the shape of $N \times F$, which are naturally compatible with GCN. However, the latent feature of the whole image is in the shape of $512 \times 14 \times 14$ (conv5_3 of VGG-16 [39]). Therefore, we design a scene nodes embedding module to project the latent scene features in the same

Net	mAP	arpl.	bike	bird	boat	bot.	bus	car	cat	chr.	cow	tbl.	dog	hrs.	mbk.	prs.	plt.	shp.	sofa	trn.	tvm.
FS-N	70.0	77.0	78.1	69.3	59.4	38.3	81.6	78.6	86.7	42.8	78.8	68.9	84.7	82.0	76.6	69.9	31.8	70.1	74.8	80.4	70.4
FR-N	73.2	76.5	79.0	70.9	65.5	52.1	83.1	84.7	86.4	52.0	81.9	65.7	84.8	84.6	77.5	76.7	38.8	73.6	73.9	83.0	72.6
SSD500 [29]	75.1	**79.8**	79.5	74.5	63.4	51.9	84.9	85.6	87.2	56.6	80.1	70.0	85.4	84.9	80.9	78.2	49.0	**78.4**	72.4	**84.6**	75.5
ION [5]	75.6	79.2	**83.1**	**77.6**	65.6	54.9	85.4	85.1	87.0	54.4	80.6	73.8	85.3	82.2	**82.2**	74.4	47.1	75.8	72.7	84.2	**80.4**
SIN [30]	76.0	77.8	73.6	61.2	61.7	**83.6**	**85.9**	85.1	56.6	**83.4**	67.2	**81.7**	83.9	78.4	78.2	47.5	**73.3**	68.3	**77.8**	76.7	70.0
GAR	**76.1**	77.4	81.3	74.8	**65.9**	59.9	85.0	**86.6**	**88.6**	56.2	**84.6**	72.2	**86.9**	**86.2**	77.3	**79.2**	46.8	77.4	4.6	83.2	77.3

Table 1. Detection on PASCAL VOC 2007, trained on VOC 2007 and VOC 2012 trainval combined. **Abbreviation:** Fast R-CNN (FS-N) [16], Faster R-CNN (FR-N), aeroplane (arpl.), bottle (bot.), chair (chr.), table (tbl.), horse (hrs.), motorbike (mbk.), person (prs.), plant (plt.), sheep (shp.), train (trn.), tvmonitor (tvm.).

Net	mAP	arpl.	bike	bird	boat	bot.	bus	car	cat	chr.	cow	tbl.	dog	hrs.	mbk.	prs.	plt.	shp.	sofa	trn.	tvm.
FS-N	68.3	82.3	78.4	70.8	52.3	38.7	77.8	71.6	89.3	44.2	73.0	55.0	87.5	80.5	80.8	72	35.1	68.3	**65.7**	80.4	64.2
FR-N	70.4	**84.9**	79.8	74.3	53.9	49.8	77.5	75.9	88.5	45.6	77.1	55.3	86.9	81.7	80.9	79.6	40.1	72.6	60.9	81.2	61.5
SIN	**73.1**	84.8	79.5	74.5	**59.7**	**55.7**	79.5	78.8	**89.9**	**51.9**	76.8	58.2	87.8	82.9	**81.8**	81.6	**51.2**	75.2	63.9	81.8	**67.8**
GAR	**73.1**	**84.9**	**80.1**	**74.6**	58.0	53.9	**80.0**	**78.9**	89.5	49.8	**77.6**	**58.4**	**88.3**	**83.2**	81.5	**82.4**	50.7	**76.4**	63.6	**82.2**	67.7

Table 2. Detection results on PASCAL VOC 2012, trained on VOC 2007 trainval, 2012 trainval and 2007 test combined.

space as instance nodes. There are $S = 2 + 2K$ scene nodes selected for scene subgraph, including an indoor node, an outdoor node, K place category nodes, and K scene attribute nodes. Instance-scene relation edges $\boldsymbol{\mathcal{E}}_{ins}^{scn}$ obtained in a similar manner as Algorithm 1. By performing softmax on $\boldsymbol{\mathcal{E}}_{ins}^{ins}$ in a row-wise manner, we are able to get a normalized relation measurement of nodes in the heterogeneous relation graph and maintain the numeric stability while training the GCR module.

Acquiring of instance-scene edges $\boldsymbol{\mathcal{E}}_{ins}^{scn}$ is distinct from that of instance-instance edges in two-fold: (i) It selects S scene nodes rather than all the $469 = (2 + 365 + 102)$ nodes to reduce computational complexity and to avoid over-smoothing [26] induced by overwhelming irrelevant information. (ii) It normalizes complementary relations by row-wise softmax. Concretely, $\boldsymbol{\mathcal{E}}_{ins}^{plc}$ and $\boldsymbol{\mathcal{E}}_{ins}^{atr}$ are normalized individually, while instance-indoor/outdoor relation are normalized by softmax([$\boldsymbol{\mathcal{E}}_{ins}^{in}$, $\boldsymbol{\mathcal{E}}_{ins}^{out}$]). Detailed computing flow is elaborated in Algorithm 2.

GCR module Now we get our heterogeneous graph nodes (instance nodes and scene nodes) and edges ready. It is time to perform graph reasoning.

The graph is fed into a similar two-layer GCN as used in [23]. In the first layer, each node has a size of 4096. For the second layer, each node has a size of 512. The output of a node is in the same size as the number of object classes. The scores of instance nodes are generated and then fused to the cursory scores weighted by learnable factors, generating the final cogitative scores:

$$\mathbf{Y_g} = \tilde{A}\,\mathrm{ReLU}(\tilde{A}XW_0)W_1 \tag{5}$$

$$Z = \mathrm{softmax}\left(\frac{exp(\mathbf{w_b})}{exp(\mathbf{w_b}) + exp(\mathbf{w_g})} \cdot \mathbf{Y_b} + \frac{exp(\mathbf{w_g})}{exp(\mathbf{w_b}) + exp(\mathbf{w_g})} \cdot \mathbf{Y_g}\right) \tag{6}$$

where $\mathbf{Y_b}$ is the cursory detection scores, $\tilde{A}$ is the normalized adjacent matrix of the heterogeneous graph, X are the nodes. W_0, W_1 are parameters of the two-layer GCN. $\mathbf{w_b}$ and $\mathbf{w_p}$ are fusion factors used for adding graph reasoning scores with the cursory scores.

4. Experiments

In this section, we evaluate the proposed GAR on PASCAL VOC [13] and MS COCO [28] object detection datasets. The base detection framework is Faster R-CNN [37] whose feature extractor is by default a VGG-16 [39] that is pre-trained on ImageNet classification dataset [25].

Following the same practice as [30], we trained the Faster R-CNN from scratch as the baseline. We find that training backbone Faster R-CNN for several epochs and then jointly training several epochs with GAR performs better than jointly training from the beginning. This is because that GAR constructs the instance and scene edges based on the cursory detection. This training strategy is denoted as *two-stage (M, N)* training, where M and N represent the number of training epochs in the first and the second stage, respectively. On the contrary, training the whole system from scratch is denoted as *one-stage (M+N)* training wherein the network is trained for $M + N$ epochs in total.

Specifically, when training on VOC 2007 dataset with *two-stage (5, 5)* strategy, we use a learning rate of 5×10^{-4} for the first 5 epochs, then 5×10^{-5} for the last 5 epochs. When training on VOC 2012 trainval with VOC 2007 trainval combined following *two-stage (4, 6)* strategy, we use a learning rate of 5×10^{-4} for the first 4 epochs, then 5×10^{-5} for the following 6 epochs. When training on COCO 2014 dataset with *two-stage (4, 6)* strategy, we use a learning rate 5×10^{-4} for the first 4 epochs and 5×10^{-5} for the last 6 epochs.

Net	AP	AP^{50}	AP^{70}	AP^S	AP^M	AP^L	AR^1	AR^{10}	AR^{100}	AR^S	AR^M	AR^L
FS-N	20.5	39.9	19.4	4.1	20.0	35.8	21.3	29.5	30.1	7.3	32.1	52.0
FR-N	21.1	40.9	19.9	6.7	22.5	32.3	21.5	30.4	30.8	9.9	33.4	49.4
ION	23.0	42.0	23.0	6.0	23.8	**37.3**	23.0	**32.4**	**33.0**	9.7	**37.0**	**53.5**
SIN	**23.2**	**44.5**	22.0	**7.3**	**24.5**	36.3	22.6	31.6	32.0	**10.5**	34.7	51.3
GAR	23.1	44.0	**23.1**	7.0	23.8	37.1	**23.1**	32.0	32.4	10.3	35.9	51.7

Table 3. Detection on COCO 2014 test-dev.

GAR	mAP	arpl.	bike	bird	boat	bot.	bus	car	cat	chr.	cow	tbl.	dog	hrs.	mbk.	prs.	plt.	shp.	sofa	trn.	tvm.
K=1	70.2	70.5	77.2	70.1	55.3	52.8	78.4	83.0	83.6	**49.8**	80.7	59.1	78.0	83.9	75.6	77.1	42.5	72.4	64.2	75.2	**73.8**
K=3	**70.7**	**71.5**	77.7	**71.2**	**56.8**	54.4	78.2	83.8	**85.0**	49.2	**81.4**	59.0	77.5	83.6	**76.1**	78.8	**43.9**	72.0	64.8	**76.6**	72.9
K=5	70.4	70.9	77.9	69.6	55.1	**55.1**	76.6	**85.0**	83.6	48.3	81.4	59.3	**80.6**	**84.1**	74.9	77.5	43.9	71.7	**65.3**	74.2	72.6
K=10	70.2	69.1	**78.2**	69.8	54.3	53.7	**78.9**	84.2	83.2	48.3	78.9	**61.5**	80.5	84.1	75.6	**79.0**	42.7	**73.2**	64.1	74.1	71.3

Table 4. Performance on VOC 2007 validation set using different K for scene nodes selection.

4.1. Overall Performance

PASCAL VOC. There are 20 classes of objects in the VOC dataset. The VOC 2007 dataset consists of about 5k training and validation combined (trainval) images and 5k testing images, while VOC 2012 dataset includes about 11k trainval images and 11k test images. We set two kinds of training datasets. The evaluations that are performed on the VOC 2007 and VOC 2012 testing sets are shown in Table 1 and Table 2, respectively. By applying GAR, we get the mAP of 76.1% on VOC 2007 testing set and mAP of 73.1% on VOC 2012 testing set.

MS COCO To validate the efficacy of GAR on a larger dataset, we conduct experiments on COCO and summarize the results in Table 3. COCO dataset involves 80 object categories. Different from VOC, COCO dataset uses AP as its evaluation metric. The overall performance AP averages mAP over different intersection over union (IOU) thresholds from 0.5 to 0.95, placing more weight on localization. In this more challenging dataset, GAR achieves 23.1% on test-dev score and brings about 2.1% improvement over baseline detector, again verifying the advantage of its efficacy.

4.2. Design Analysis and Ablation Study

Top K place labels and scene attributes As aforementioned, a lot of place labels are synonyms which can be hardly differentiated. For example, "cafeteria", "restaurant" and "dining hall" are all places for dining and share a lot of common features. Though making use of more possible place labels and scene attributes tends to provide more information about the scene. However, too much irrelevant information involved aggravates over-smoothing problem [26] of GCN. To find the optimal design hyper-parameter K, we conduct evaluations on VOC 2007 validation set with different K by tuning K, as shown in Table 4. It is observed that $K = 3$ achieves the optimal mAP for GAR.

To get a better understanding of object-object/scene relation, we summarize the top 3 related entities (object classes, place categories, scene attribute, and indoor/outdoor labels) in terms of co-occurrence for each object class, as shown in Table 5. Some interesting phenomenons are observed: First, the object "person" is highly correlated with other objects in the VOC dataset; Second, besides "person", "car" usually appears with "bus" and "motorbike". Meanwhile, these three methods of transportation are all labeled as "outdoor" usually appear at "street" and "parking lot" which are featured by "man-made", "natural-light" and "open-area". In Table 5, "NA" in the indoor/outdoor field means that neither the probability of "indoor" nor "outdoor" exceeds 30%.

Scene/Object Ablative Comparison We evaluate the effectiveness of object-object reasoning (edge) and object-scene reasoning (scene) separately and compare their performance with the previous work SIN [30]. As shown in Table 7, all methods are trained on VOC 2007 trainval and testing set on VOC 2007 testing set. GAR-scene module achieves better mAP of 70.29% as compared to SIN-scene with mAP of 70.23%. SIN-edge module provides higher mAP of 70.31% than GAR-edge with mAP of 70.29%. The reason is that SIN-edge takes more complicated spatial and geometric relations, which might contain more information than co-occurrence relation used in GAR-edge.

Interestingly, it is observed that the edge/scene module boosts mAP in some categories, such as "boat", "cow", "horse", "sheep", "tvmonitor", etc. This is expected since such categories are generally correlated with scene context and other objects occurrence. However, we observed that the mAP of "table" is suffering from degradation. One possible reason is that "table" is so similar to "chair" and "sofa". Therefore, the possibility of mislabeling as well as the IOU

aeroplane	**bicycle**	**bird**	**boat**	**bottle**	**bus**	**car**	**cat**	**chair**	**cow**
person car boat	person car bus	person cow boat	person car bird	person diningtable chair	car person bicycle	person bus motorbike	chair person sofa	person diningtable sofa	person horse bird
outdoor	outdoor	outdoor	outdoor	indoor	outdoor	outdoor	indoor	indoor	outdoor
airfield runway sky	raceway crosswalk street	water-hole pond field	harbor boat-deck ocean	pub beer-hall bar	bus-station street park-lot	park-lot street raceway	vetr-office pet-shop kennel	din-room din-hall liv-room	corral pasture farm
natur-light open-area man-made	natur-light man-made no-horizon	natur-light no-horizon open-area	natur-light open-area man-made	no-horizon enclosed man-made	man-made natur-light open-area	man-made natur-light open-area	no-horizon enclosed man-made	no-horizon enclosed man-made	natur-light open-area no-horizon
diningtable	**dog**	**horse**	**motorbike**	**person**	**pottedplant**	**sheep**	**sofa**	**train**	**tvmonitor**
chair person bottle	person sofa chair	person car dog	person car bicycle	car chair horse	person chair sofa	person dog cow	person chair tvmonitor	person car boat	chair person sofa
indoor	NA	outdoor	outdoor	NA	NA	outdoor	indoor	outdoor	indoor
din-hall restaurant din-room	veter-ofc outdoor pet-shop	corral racecourse stable	raceway street highway	street indoor nurs-home	roof-grdn vege-grdn liv-room	pasture farm hayfield	liv-room wait-room drm-room	rail-track platform platform	home-ofc office cmpt-room
no-horizon enclosed man-made	no-horizon enclosed man-made	natur-light open-area man-made	natur-light man-made no-horizon	no-horizon man-made natur-light	no-horizon man-made enclosed	natur-light open-area grass	no-horizon enclosed man-made	man-made natur-light open-area	no-horizon enclosed man-made

Table 5. Top 3 related object/scene entities in terms of co-occurrence, on VOC 2007 trainval. From top to bottom: three categories of mostly co-occurred objects, the indoor/outdoor label, three categories of mostly co-occurred places and three mostly related scene attributes.

Net	GAR			SIN		
Mode	Edge	Scene	Total	Edge	Scene	Total
#FLOPS	540M	17.4M	558M	25.8G	102M	25.9G
#params	<1K	2.35M	2.35M	101M	106M	207M

Table 6. Number of FLOPS and number of parameters required by GAR and SIN modules.

loss are largely increased due to similar relations.

Qualitative Analysis We show representative qualitative results in Figure 3 to present how GAR with graph reasoning helps object detection. GAR benefits object detection in two folds:

(1) It detects obscure objects better with reliable scene inference. For example, Figure 3a depicts a car in front of a gas station. With the detected scene and prior knowledge that "car" is highly correlated with "person", GAR successfully detects the driver inside the car. Similar reasoning is applied to the dog in Figure 3b and the person at the left-bottom corner of Figure 3d.

(2) It helps to drop irrelevant objects which are, in some sense, ridiculous. For example, the baseline detector detects the car door as a "tvmonitor" in Figure 3c. While based on the prior knowledge in Table 5, we know "tvmonitor" is typically related with "indoor", "enclosed area" and frequently appears in "home office", "office" and "computer room". Thus, GAR drops this wrong detection correctly. Other similar cases also demonstrate the efficacy of GAR, Another example is that the "boat" detected by the baseline detector in Figure 3d is successfully eliminated by GAR.

Sensitivity of Object Characteristics To further quatitively measure the approvement achived by GAR, we look at a detailed breakdown of results of VOC 2007 using the detection analysis tool from [20]. Figure 4 provides a compact summary of the sensitivity to each characteristic group and the potential impact of improving robustness on seven categories selected by [20], which are 'aeroplane', 'bicycle', 'bird', 'boat', 'cat', 'chair' and 'diningtable'. Overall, our method is more robust than baseline and SIN method against occlusion, truncation and area size.

Computational Overhead The proposed GAR is efficient for both training and inference thanks to its paralleled computing flow and small model size. Use the same feature extractor network (VGG-16) as the backbone object detection, we take the output feature of conv5_3 and re-train the fully connected layer for the scene detector on Place365 scene recognition dataset [43]. Table 6 demonstrates the number of floating point operations (FLOPs) as well as the number of parameters required by GAR and compares it with the previous work SIN [30]. It is observed that SIN requires much more computing and parameter memory than our GAR due to its complicated edge calculation and sequential GRU propagation. We compare the training and inference speed of baseline Faster R-CNN, SIN and GAR on a single Nvidia RTX 2080 GPU. For sake of the fair comprarison, we implemente the SIN with Pytorch [35] framework. We also

Method	mAP	arpl.	bike	bird	boat	bot.	bus	car	cat	chr.	cow	tbl.	dog	hrs.	mbk.	prs.	plt.	shp.	sofa	trn.	tvm.
FR-N	68.89	68.9	77.7	67.5	54.0	53.8	76.0	80.0	80.0	49.0	74.0	65.8	77.2	80.2	76.5	76.9	39.0	67.0	65.5	75.6	71.5
SIN-E	**70.31**	70.0	78.2	67.5	57.6	**56.0**	**78.5**	80.0	79.9	**51.1**	74.1	**70.2**	78.0	80.6	**77.5**	**77.6**	41.0	69.0	**68.3**	76.2	74.6
SIN-S	70.23	70.1	78.4	**69.3**	**60.9**	53.1	77.0	79.6	**86.0**	49.9	75.0	68.0	78.7	80.7	74.7	77.3	41.2	68.3	65.4	**76.6**	74.5
GAR-E	70.21	**70.2**	**78.9**	67.5	56.5	54.7	75.7	84.3	84.1	48.4	78.5	61.1	79.0	**84.0**	74.8	77.2	42.4	**70.9**	65.4	75.3	**74.7**
GAR-S	70.29	69.6	76.1	68.3	57.2	54.2	76.9	**84.6**	83.6	48.6	**79.6**	62.2	**80.6**	83.5	75.2	76.7	**43.4**	69.7	65.3	75.5	74.5

Table 7. Ablative comparison with SIN on VOC 2007 test, trained on VOC 2007 trainval. Abbreviation: edge module (E), scene module (S).

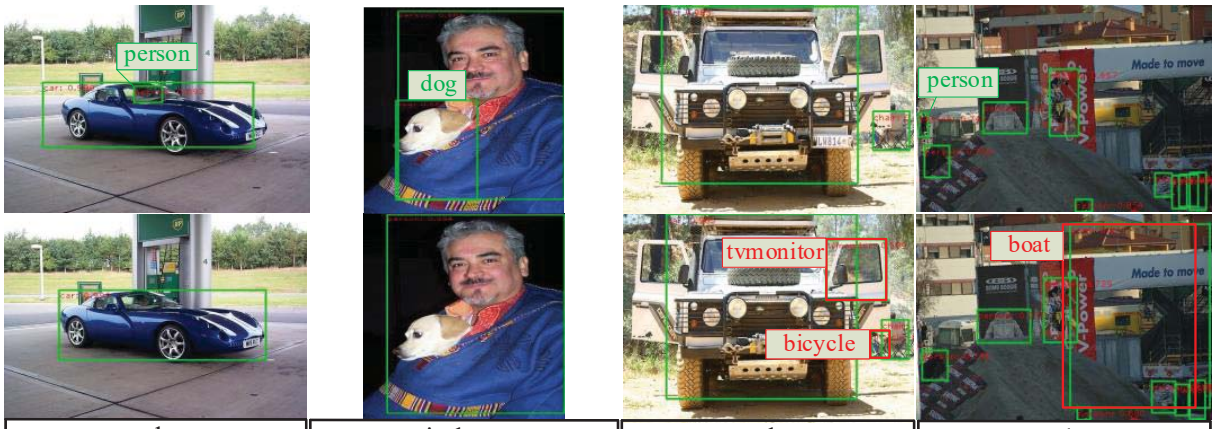

Figure 3. Qualitative results of GAR detection. From top to bottom: GAR detection, baseline detection, the indoor/outdoor label, place categories with possibilities, scene attributes.

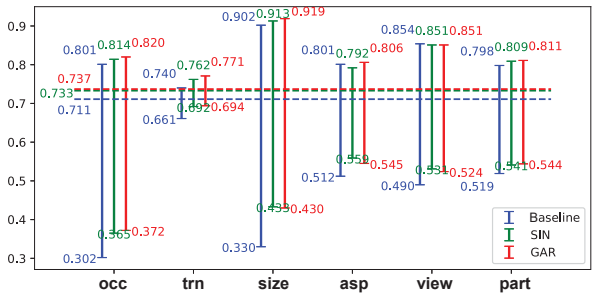

Figure 4. Summary of sensitivity of object characteristics. It presents the average (over 7 categories) Normalized AP(APN [20]) of the highest score and lowest score subsets in each characteristic group (occlusion, truncation, bounding box area, aspect ratio, viewpoint, part visibility). Overall APN is indicated by the dashed line. Red: Scene. Green: baseline.

optimize its edge calculation with parallel tensor operation instead of iterative loops where used in its original Tensorflow [1] implementation. For training, frame per second (FPS) of baseline is 6.3, SIN is 2.2 and GAR is 4.0. For inference, FPS of baseline is 15.5, SIN is 8.8 and GAR is 14.1. It can be observed that the overhead of GAR module is much lower than SIN.

5. Conclusion

In this paper, we propose a graph-assisted detection method, GAR, that leverages object-object and object-scene relations in object detection. Experiments show prominent accuracy improvement, especially on the categories which are highly correlated to scene context. Moreover, our GAR method has the advantage of computation efficiency: it requires less FLOPs and parameter memory than previous RNN-based methods, making GAR a practical solution in real-time applications.

Acknowledgment

This work was supported in part by C-BRIC, one of six centers in JUMP, a Semiconductor Research Corporation (SRC) program sponsored by DARPA. It is also partially support by National Science Foundation (CCF #1715443).

References

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, D. Murray, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] B. Alexe, N. Heess, Y. W. Teh, and V. Ferrari. Searching for objects driven by context. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 881–889. Curran Associates, Inc., 2012.

[3] M. Bar. Visual objects in context. *Nature Reviews. Neuroscience*, 5(8):617–629, Aug. 2004.

[4] J. Bastings, I. Titov, W. Aziz, D. Marcheggiani, and K. Sima'an. Graph Convolutional Encoders for Syntax-aware Neural Machine Translation. pages 1957–1967, Sept. 2017.

[5] S. Bell, C. L. Zitnick, K. Bala, and R. Girshick. Inside-Outside Net: Detecting Objects in Context with Skip Pooling and Recurrent Neural Networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2874–2883, Las Vegas, NV, USA, June 2016. IEEE.

[6] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*, 2013.

[7] H. Cai, V. W. Zheng, and K. C. Chang. A Comprehensive Survey of Graph Embedding: Problems, Techniques, and Applications. *IEEE Transactions on Knowledge and Data Engineering*, 30(9):1616–1637, Sept. 2018.

[8] P. Carbonetto, N. de Freitas, and K. Barnard. A Statistical Model for General Contextual Object Recognition. In T. Pajdla and J. Matas, editors, *Computer Vision - ECCV 2004*, Lecture Notes in Computer Science, pages 350–362. Springer Berlin Heidelberg, 2004.

[9] X. Chen and A. Gupta. Spatial Memory for Context Reasoning in Object Detection. *arXiv:1704.04224 [cs]*, Apr. 2017. arXiv: 1704.04224.

[10] J. Dai, Y. Li, K. He, and J. Sun. R-FCN: Object Detection via Region-based Fully Convolutional Networks. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 379–387. Curran Associates, Inc., 2016.

[11] M. Defferrard, X. Bresson, and P. Vandergheynst. Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 3844–3852. Curran Associates, Inc., 2016.

[12] S. K. Divvala, D. Hoiem, J. H. Hays, A. A. Efros, and M. Hebert. An empirical study of context in object detection. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1271–1278, June 2009.

[13] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The Pascal Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision*, 88(2):303–338, June 2010.

[14] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object Detection with Discriminatively Trained Part-Based Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, Sept. 2010.

[15] C. Galleguillos, A. Rabinovich, and S. Belongie. Object categorization using co-occurrence, location and appearance. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2008.

[16] R. Girshick. Fast R-CNN. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, pages 1440–1448, Washington, DC, USA, 2015. IEEE Computer Society.

[17] K. He, X. Zhang, S. Ren, and J. Sun. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. In D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, editors, *Computer Vision – ECCV 2014*, Lecture Notes in Computer Science, pages 346–361. Springer International Publishing, 2014.

[18] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. *arXiv:1512.03385 [cs]*, Dec. 2015. arXiv: 1512.03385.

[19] G. Heitz and D. Koller. Learning Spatial Context: Using Stuff to Find Things. In *Proceedings of the 10th European Conference on Computer Vision: Part I*, ECCV '08, pages 30–43, Berlin, Heidelberg, 2008. Springer-Verlag. event-place: Marseille, France.

[20] D. Hoiem, Y. Chodpathumwan, and Q. Dai. Diagnosing Error in Object Detectors. In A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, editors, *Computer Vision – ECCV 2012*, pages 340–353, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[21] H. Hu, J. Gu, Z. Zhang, J. Dai, and Y. Wei. Relation Networks for Object Detection. Nov. 2017.

[22] L. Huang, Y. Yang, Y. Deng, and Y. Yu. DenseBox: Unifying Landmark Localization with End to End Object Detection. *arXiv:1509.04874 [cs]*, Sept. 2015. arXiv: 1509.04874.

[23] T. N. Kipf and M. Welling. Semi-Supervised Classification with Graph Convolutional Networks. *arXiv:1609.02907 [cs, stat]*, Sept. 2016. arXiv: 1609.02907.

[24] T. Kong, F. Sun, H. Liu, Y. Jiang, and J. Shi. FoveaBox: Beyond Anchor-based Object Detector. *arXiv:1904.03797 [cs]*, Apr. 2019. arXiv: 1904.03797.

[25] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[26] Q. Li, Z. Han, and X.-M. Wu. Deeper Insights into Graph Convolutional Networks for Semi-Supervised Learning. *arXiv:1801.07606 [cs, stat]*, Jan. 2018. arXiv: 1801.07606.

[27] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal Loss for Dense Object Detection. *arXiv:1708.02002 [cs]*, Aug. 2017. arXiv: 1708.02002.

[28] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: Common Objects in Context. In D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 740–755, Cham, 2014. Springer International Publishing.

[29] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. SSD: Single Shot MultiBox Detector. *arXiv:1512.02325 [cs]*, 9905:21–37, 2016. arXiv: 1512.02325.

[30] Y. Liu, R. Wang, S. Shan, and X. Chen. Structure Inference Net: Object Detection Using Scene-Level Context and Instance-Level Relationships. *arXiv:1807.00119 [cs]*, June 2018. arXiv: 1807.00119.

[31] D. Marcheggiani and I. Titov. Encoding Sentences with Graph Convolutional Networks for Semantic Role Labeling. *arXiv:1703.04826 [cs]*, Mar. 2017. arXiv: 1703.04826.

[32] M. Marszałek, I. Laptev, and C. Schmid. Actions in Context. June 2009.

[33] R. Mottaghi, X. Chen, X. Liu, N. Cho, S. Lee, S. Fidler, R. Urtasun, and A. Yuille. The Role of Context for Object Detection and Semantic Segmentation in the Wild. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 891–898, June 2014.

[34] A. Oliva and A. Torralba. The role of context in object recognition. *Trends in Cognitive Sciences*, 11(12):520–527, Dec. 2007.

[35] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in PyTorch. In *NIPS Autodiff Workshop*, 2017.

[36] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You Only Look Once: Unified, Real-Time Object Detection. *arXiv:1506.02640 [cs]*, June 2015. arXiv: 1506.02640.

[37] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 91–99. Curran Associates, Inc., 2015.

[38] A. Shrivastava and A. Gupta. Contextual Priming and Feedback for Faster R-CNN. In B. Leibe, J. Matas, N. Sebe, and M. Welling, editors, *Computer Vision – ECCV 2016*, pages 330–348, Cham, 2016. Springer International Publishing.

[39] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv:1409.1556 [cs]*, Sept. 2014. arXiv: 1409.1556.

[40] Torralba, Murphy, Freeman, and Rubin. Context-based vision system for place and object recognition. In *Proceedings Ninth IEEE International Conference on Computer Vision*, pages 273–280 vol.1, Oct. 2003.

[41] L. Yao, C. Mao, and Y. Luo. Graph Convolutional Networks for Text Classification. *arXiv:1809.05679 [cs]*, Sept. 2018. arXiv: 1809.05679.

[42] X. Zeng, W. Ouyang, B. Yang, J. Yan, and X. Wang. Gated Bidirectional CNN for Object Detection. In B. Leibe, J. Matas, N. Sebe, and M. Welling, editors, *Computer Vision – ECCV 2016*, volume 9911, pages 354–369. Springer International Publishing, Cham, 2016.

[43] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba. Places: A 10 Million Image Database for Scene Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6):1452–1464, June 2018.