Spatial-aware Graph Relation Network for Large-scale Object Detection

Hang Xu[1*] ChenHan Jiang[2*] Xiaodan Liang[2†] Zhenguo Li[1]
[1]Huawei Noah's Ark Lab [2]Sun Yat-sen University

Abstract

How to proper encode high-order object relation in the detection system without any external knowledge? How to leverage the information between co-occurrence and locations of objects for better reasoning? These questions are key challenges towards large-scale object detection system that aims to recognize thousands of objects entangled with complex spatial and semantic relationships nowadays.

Distilling key relations that may affect object recognition is crucially important since treating each region separately leads to a big performance drop when facing heavy long-tail data distributions and plenty of confusing categories. Recent works try to encode relation by constructing graphs, e.g. using handcraft linguistic knowledge between classes or implicitly learning a fully-connected graph between regions. However, the handcraft linguistic knowledge cannot be individualized for each image due to the semantic gap between linguistic and visual context while the fully-connected graph is inefficient and noisy by incorporating redundant and distracted relations/edges from irrelevant objects and backgrounds. In this work, we introduce a Spatial-aware Graph Relation Network (SGRN) to adaptive discover and incorporate key semantic and spatial relationships for reasoning over each object. Our method considers the relative location layouts and interactions among which can be easily injected into any detection pipelines to boost the performance. Specifically, our SGRN integrates a graph learner module for learning a interpatable sparse graph structure to encode relevant contextual regions and a spatial graph reasoning module with learnable spatial Gaussian kernels to perform graph inference with spatial awareness. Extensive experiments verify the effectiveness of our method, e.g. achieving around 32% improvement on VG(3000 classes) and 28% on ADE in terms of mAP.

1. Introduction

Most recent advancement of CNN detectors is focused on the task with a limited number of categories (say, 20 for

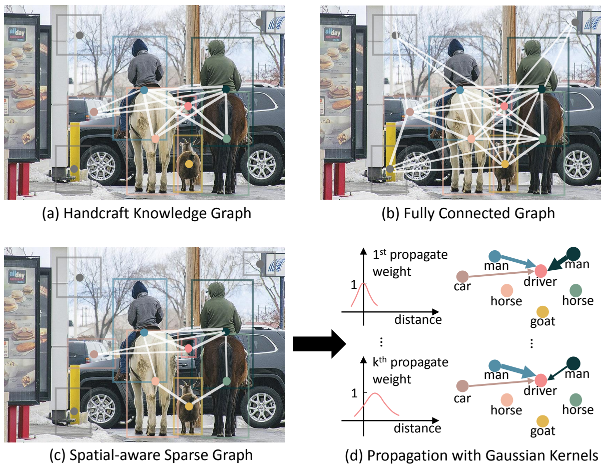
(a) Handcraft Knowledge Graph
(b) Fully Connected Graph
(c) Spatial-aware Sparse Graph
(d) Propagation with Gaussian Kernels
1st propagate weight
kth propagate weight
1
distance
man
man
driver
car
horse
horse
goat

Figure 1. Different choice of constructing a graph to encode relation: (a) Using handcraft knowledge to build class-to-class graph. Some spatial relations are ignored and the fixed graph cannot adapt to the image due to the gap between linguistic and visual context. (e.g. "goat" is alone). (b) Implicitly learning a fully-connected graph between regions. The learned edges are redundant since background regions are also connected and the fully-connected graph ignore the pairwise spatial information. (c) Our proposed SGRN learns a spatial-aware sparse graph, leveraging semantic and spatial layout relationship. (d) An example of propagation with multiple learnable spatial Gaussian kernels about the "driver" node. Different spatial kernels allows the propagation of the graph behaves differently according to the pairwise spatial information(different thickness of the edges).

VOC [11] and 80 for MS-COCO dataset [32]). However, there is an increasing need for recognizing more kinds of objects (e.g. 3000 categories in VG [25]) so that large-scale object detection [19, 23] has received a lot of attention because of its practical usefulness in the industry. Current detection pipelines mostly treat the recognition of each region separately and suffer from great performance drops when facing heavy long-tail data distributions and plenty of confusing categories. It has been well recognized in the community that relation between objects can help to improve object recognition before the prevalence of deep learning [12, 17, 47, 48, 49]. Adding more contextual information by evolving the relation information will relieve the above

*Both authors contributed equally to this work.
†Corresponding Author: xdliang328@gmail.com

problems. Therefore, a crucial challenge for large-scale object detection is how to capture and unify semantic and spatial relationships and boost the performance.

With the advancement of geometric deep learning, using graph seems to be the most appropriate way to model relation because of its flexible structure of modeling pairwise interaction. Figure 1 gives an illustration of different choice of design a graph to encode pairwise relation for the task of detection. Figure 1a uses handcraft linguistic knowledge [24, 36, 23, 6] to build a class-to-class graph. For example, Jiang et al. [23] recently try to incorporate semantic relation reasoning in large-scale detection by different kinds of knowledge forms. However, their method heavily relied on the annotations of attribution and relationship from VisualGenome data. Moreover, some spatial relations may be ignored and the fixed graph cannot adapt to the image due to the semantic gap between linguistic and visual context. (e.g. "goat" is alone). On the other hand, some works [34, 5, 20, 51] try to implicitly learns a fully-connected graph between regions from visual features as shown in Figure 1b. For example, Hu et al. [20] introduced the Relation Networks which use an adapted attention module to allow interaction between the object's visual features. However, their fully-connected relation is inefficient and noisy by incorporating redundant and distracted relationships/edges from irrelevant objects and backgrounds while the pairwise spatial information is also not fully utilized. Moreover, it is not clear what is learned in the module as mentioned in their paper. Thus, our work aims to develop a graph-based network which can model relation with awareness of the spatial information as well as efficiently learning an interpretable sparse graph structure directly from the training images. Figure 1c shows that a spatial-aware sparse graph is learned by our method, leveraging both semantic and spatial relationship.

In this paper, we propose a novel spatial-aware graph relation network (SGRN) for large-scale object detection. Our network simply consists of two modules: one sparse graph learner module and one spatial-aware graph convolution module. Instead of building category-to-category graph [7, 40], the proposal regions are defined as graph nodes. A sparse graph structure is learned via a relation learner module. This not only identifies the most relevant regions in the image which can help to recognize the object in the image but also avoiding unnecessary overhead with the negative regions. Spatial-aware graph convolutions driven by learnable spatial Gaussian kernels is then performed to propagate and enhance the regional context representation. The design of Gaussian kernels in graph convolutions allows the graph propagation to be aware of different spatial relationship as shown in Figure 1c. Finally, each region's new enhanced context are concatenated to the original feature to improve the performance of both classification and localization in an end-to-end style. Our method is in-place and easily plugged into any existing detection pipeline for endowing its ability to capture and unify semantic and spatial relationships.

Our SGRN thus enables adaptive graph reasoning over regions with an interpretable learned graph (see Figure 4). Both the contextual information and spatial information is distilled and propagated through the graph efficiently. The problem of imbalanced categories can then be alleviated by sharing essential characteristics among frequent/rare categories. Also, the recognition of difficult regions with heavy occlusions, class ambiguities and tiny-size problems can be thus remedied by the enhanced context information of related regions. Moreover, our method has shown great domain transferability by reusing the graph learner and reasoning module as shown in the Section 4.5.

The proposed SGRN outperforms current state-of-art detection methods without adding any additional information, i.e., [30], Faster R-CNN [45], Relation Network [20], HKRM [23] and RetinaNets [31]. Consistent improvements on the base detection network FPN and Faster R-CNN on several object detection benchmarks have been observed, i.e., VG [25] (1000/3000 categories), ADE [53](445 categories) , MS-COCO [32](80 categories). In particular, SGRN achieves around 32% of mAP improvement on VG (3000 categories), 14% on VG (1000 categories), 28% on ADE, 8% on MS-COCO.

2. Related Work

Object Detection.

Object detection is a core problem in computer vision. Big progress has been made in recent years due to the usage of CNN (backbone such as Resnet 101 [18]). Modern object detection methods can usually be categorized in two groups: 1) two-stage detection methods: Faster R-CNN [45], R-FCN [8], FPN [30]. They use a Region Proposal Network to generate regions of interests in the first stage and then send the region proposals down the pipeline for object classification and bounding-box regression. 2) one-stage detection methods such as SSD [33] and YOLO [43]. They use regression by taking an input image and learning the class probabilities and bounding box coordinates. Such models reach lower accuracy rates but are much faster than two-stage object detectors. However, the number of categories being considered usually is small: 20 for PASCAL VOC [11] and 80 for COCO [32]. Also, those methods are usually performed on each proposal individually without considering the relationship between regions.

Visual Reasoning.

Visual reasoning aims to combine different information or interactions between objects or scenes. Examples can be

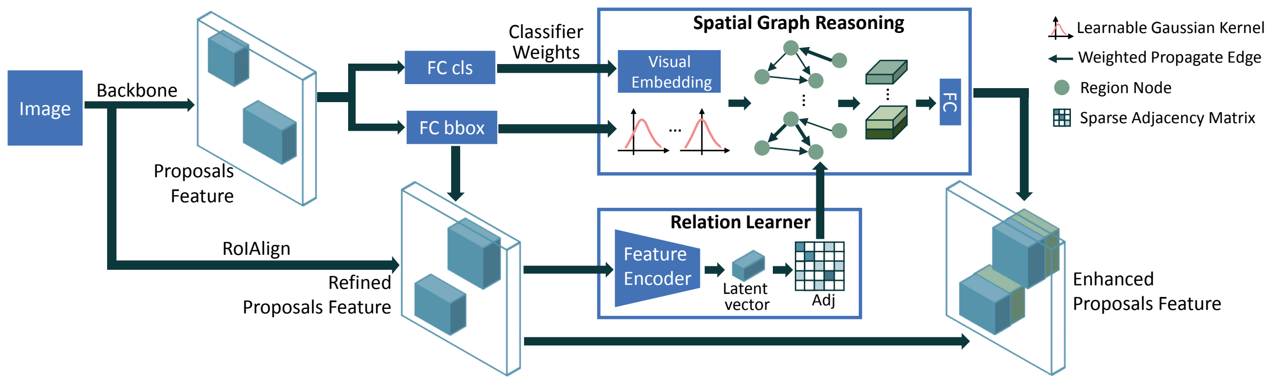

Figure 2. An overview of the proposed SGRN. Our method can be stacked on any modern detection network. SGRN encodes the relation between regions as an undirected graph. The relation graph learner module first learns a sparse adjacency matrix from the visual feature which retains the most relevant connections. Then the weights of the previous classification layer are collected and soft-mapped to the regions to become visual embeddings of each regions. The pairwise spatial information between regions (distance, angle) is feed into Gaussian kernels to determine the patterns of graph convolution. In the spatial-aware graph reasoning module, visual embedding of different regions are evolved and propagated according to the sparse adjacency matrix and the Gaussian kernels. The output of spatial graph reasoning module is then concatenated to the original region features to improve both classification and localization.

found in the task of classification [36, 2], object detection [6, 20, 23] and visual relationship detection [7]. Early works usually involve handcraft relationships or shared attribute among objects [1, 2, 26, 37]. For example, [14, 35, 44] rely on finding similarity such as the attributes in the linguistic space. [13, 15, 39] use object relationships as a post-processing step. Recent works consider a graph structure [6, 7, 24, 36] to incorporate external knowledge for various tasks. Deng et al. [10] employed a label relation graph to guide the classification. Chen et al. [6] leverage local region-based reasoning and global reasoning to facilitate object classification. However, their method heavily relied on external handcraft linguistic knowledge (word embedding). Those handcraft graph may not be appropriate because of the gap between linguistic and visual context. Other works [20, 34, 41] encode the relation in an implicit way. Liu et al. [34] proposed Structure Inference Network (SIN) which learns a fully-connect graph implicitly with stacked GRU cell to encode the message. However, the usage of fully-connected-graph allows redundant information flow and make the GRU cell less efficient which leads to a low reported performance (mAP: 23.2% on MSCOCO). By contrast, our SGRN learns a sparse relation graph which can be used to facilitate our spatial-aware GCN module.

Graph Convolutional Neural Networks

Graph CNNs (GCNs) aims to generalize Convolutional Neural Networks (CNNs) to graph-structured data. Advances in this direction are often categorized as spectral approaches and spatial approaches. Spectral GCNs [9, 24] use analogies with the Euclidean domain to define a graph Fourier transform, allowing to perform convolutions in the spectral domain as multiplications. Spatial GCNs [3, 38, 50] define convolutions with a patch operator directly on the graph, operating on groups of node neighbors. Monti et al. [38] presented mixture model CNNs (MoNet), a spatial approach which provides a unified generalization of CNN architectures to graphs. Graph Attention Networks [50] model the convolution operator as an attention operation on node neighbors. Inspired by those work, our model also defined the graph convolution as a mixture model. However, while these methods which learn a fixed graph structure, we aim to learn a dynamic adaptive graph for each image which can leverage the information between co-occurrence and locations of objects.

3. The Proposed Approach

3.1. Overview

An overview of SGRN can be found in Figure 2. We develop a spatial-aware graph relation network which can be implemented on any modern dominant detection system to further improve their performance. In our network, the relation is formulated as a region-to-region undirected graph $\boldsymbol{G} : \boldsymbol{G} =< \mathcal{N}, \mathcal{E} >$. The relation graph learner module first learns a interpretable sparse adjacency matrix from the visual feature which only retains the most relevant connections for recognition of the objects. Then the weights of the previous classification layer are collected and soft-mapped to the regions to become visual embeddings of each region. The pairwise spatial information between regions (distance, angle) is calculated and feeds in Gaussian kernels to determine the patterns of graph convolution. In the spatial-aware graph reasoning module, visual embeddings of different regions are evolved and propagated according to the sparse adjacency matrix and the Gaussian kernels. The output of the spatial graph reasoning module is then concatenated to the original region features to improve both classification and localization.

3.2. Relation Learner Module

This module aims to produce a graphical representation of the relationship between proposal regions which is relevant to the object detection. We formulate the relation as a region-to-region undirected graph $\boldsymbol{G}$: $\boldsymbol{G} =< \mathcal{N}, \mathcal{E} >$, where each node in $\mathcal{N}$ corresponds to a region proposals and each edge $e_{i,j} \in \mathcal{E}$ encodes relationship between two nodes.. We then seek to learn the $\mathcal{E} \in \mathbb{R}^{N_r \times N_r}$ thus the node neighborhoods can be determined.

Formally, given the regional visual features of D dimension extracted from the backbone network for region proposals, we use the regional visual features $\mathbf{f} = \{\boldsymbol{f}_i\}_{i=1}^{N_r}, f_i \in \mathbb{R}^D$ as the input of our module. We first transform the visual features to a latent space $\mathbf{Z}$ by non-linear transformation denoted by

$$z_i = \phi(\mathbf{f}), i = 1, 2, ..., N_r \tag{1}$$

, where $z_i \in \mathbb{R}^L$, L is the dimension of the latent space and $\phi(.)$ is a non-linear function. In this paper, we consider two fully-connected layers with ReLU activation as the non-linear function $\phi(.)$. Let $\mathbf{Z} \in \mathbb{R}^{N_r \times L}$ be the collection of $\{z_i\}_{i=1}^{N_r}, z_i \in \mathbb{R}^L$, the adjacency matrix for the undirected graph $\boldsymbol{G}$ with self loops can be then calculated by a matrix multiplication as $\mathcal{E} = \mathbf{Z}\mathbf{Z}^T$, so that $e_{i,j} = z_i z_j^T$.

Note that a lot of background (negative) samples exist among those N_r region proposals. Using a fully connected adjacency matrix $\mathcal{E}$ will establish relationship between backgrounds(negative) samples. Those redundant edges will lead to greater computation cost. Moreover, the subsequent spatial-aware graph convolution will over-propagate the information and the output of the graph convolution will be the same for all nodes. To solve this problem, we need to impose constraints on the graph sparsity. For each region proposal i, we only retain the top t largest value of each row of $\mathcal{E}$. In other words, most t relevant nodes are picked as the neighbourhood of each region proposal i:

$$\text{Neighbour}(\text{Node } i) = \text{Top-t}_{j=1,..,N_r}(e_{i,j})$$

. This ensures a spare graph structure focusing on the most relevant relationship for recognition of the objects.

3.3. Visual Embeddings of the Regions

Most existing graph-based approaches [24, 36, 7, 6] propagate visual features locally among regions in each image according to the edges. However, their methods will fail when the regional visual features are poor thus the propagation is inefficient or even wrong. Note that this situation often happens in large-scale detection when heavy occlusions and ambiguities exist in the image. To alleviate this problem, our method tries to propagate information globally over all the categories. In other words, our method needs to create a high-level semantic visual embedding for each category which can be regarded as an ideal prototype for one particular object category.

In some zero/few-shot problems, they [47, 52, 16] use the classifier's weights as the embedding or representation of an unseen/unfamiliar category, we try to use the weights as the visual embedding for each category. This is due to the fact that the weights of the classifier actually contains high-level semantic information since they record the feature activation trained from all the images. Formally, let $\mathbf{W} \in \mathbb{R}^{C \times (D+1)}$ denotes the weights (parameters) of the previous classifiers where C is the number of categories and D is the dimension of the visual features. The category visual embedding can be obtained by copying the parameters $\mathbf{W}$ (including the bias) from the previous classification layer of the base detection networks. Note that the $\mathbf{W}$ are updated during training so that our visual embeddings are more accurate through time. Furthermore, our model can be trained in an end-to-end fashion which avoids taking average or clustering across all the dataset [27].

Since our graph $\boldsymbol{G}$ is a region-to-region graph, we need to find most appropriate mappings from the category visual embedding $w \in \mathbf{W}$ to the regional representations of nodes $x_i \in \boldsymbol{X}$ (the input of our spatial graph reasoning module). Chen et al. [6] suggest a *hard-mapping* which directly uses the one-to-one previous classification results as the category-to-region mapping. However, their mapping will be wrong if previous classification results is wrong. Instead, we use a *soft-mapping* which compute the mapping weights $m_{w \to x_i} \in \mathbf{M}^s$ as $m_{w \to x_i} = \frac{\exp(s_{ij})}{\sum_j \exp(s_{ij})}$, where s_{ij} is the classification score for the region i towards category j from the previous classification layer of the base detector. Thus the input $\boldsymbol{X} \in \mathbb{R}^{N_r \times (D+1)}$ of our spatial-aware graph reasoning module can be computed as $\boldsymbol{X} = \mathbf{M}^s\mathbf{W}$,where $\mathbf{M}^s \in \mathbb{R}^{N_r \times C}$ is the soft-mapping matrix.

3.4. Spatial-aware Graph Reasoning Module

Based on the regional input (nodes) $\boldsymbol{X} \in \mathbb{R}^{N_r \times (D+1)}$ and the learned graph edges $\mathcal{E} \in \mathbb{R}^{N_r \times N_r}$, the graph reasoning guided by the edges is employed to learn a new object representations for further enhancement of classification and localization. As the locations of the regions in the image are also crucial for the graph reasoning, spatial information should also be considered in our graph reasoning module. Here, we introduce our Spatial-aware Graph Reasoning Module to use Graph Convolutional Neural Networks (GCN) for modeling the relation and interaction coherently with spatial information.

To capture the pairwise spatial information, we use a pairwise pseudo-coordinate function $\boldsymbol{u}(a, b)$ which defines, for each node a, $\boldsymbol{u}(a, b)$ will returns the coordi-

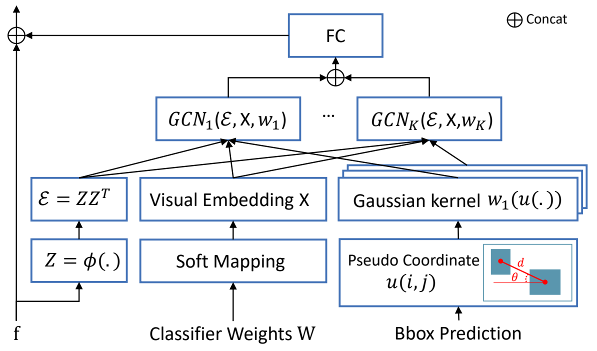

Figure 3. Flowchart of the Spatial-aware Graph Reasoning Module. The relation learner module learns a sparse adjacency matrix $\mathcal{E}$ from the visual feature $\mathbf{f}$. The classifier weights $\mathbf{W}$ is soft-mapped to regions as the Visual embedding $\boldsymbol{X}$. Then the pairwise pseudo coordinates $\boldsymbol{u}(i,j)$ are calculated and determine $w_k(.)$ from the K Gaussian kernels with learnable means and covariances. Finally, $\mathcal{E}$, $\boldsymbol{X}$, and $w_k(.)$ are feed into Graph Convolutional Neural Networks(GCN) defined by Equation (2). The output of our Spatial-aware Graph Reasoning Module is concatenated to the $\mathbf{f}$ to improve both classification and localization.

nated of node b in that system. Naturally, the relative positions of each region (node) in the image can be recognized as the pseudo-coordinate system. In this paper, we use a polar function $\boldsymbol{u}(a,b) = (d,\theta)$ which returns a 2-d vector that calculates the distance and the angle of two centers($[c_a, y_a]$,$[c_b, y_b]$) of the region proposals a and b, e.g. $d = \sqrt{(c_a - c_b)^2 + (y_a - y_b)^2}$ and $\theta = \arctan\left(\frac{y_b - y_a}{c_b - c_a}\right)$.

Then we need to formulate our spatial-aware graph reasoning by defining a patch operator to describe the influence and propagation of each neighboring node in the graph. Similar to MoNet [38], we define the patch operator by a set of K Gaussian kernels of learnable means and covariances. Formally, given the regional semantic input $x_j \in \boldsymbol{X}$ and the graph structure $\boldsymbol{G} =< \mathcal{N}, \mathcal{E} >$, the patch operator at each kernel k for node i is given by:

$$\mathrm{f}'_k(i) = \sum_{j \in \mathrm{Neighbour}(i)} w_k(\boldsymbol{u}(i,j)) x_j e_{ij}, \tag{2}$$

where Neighbour(i) denotes the neighborhood of node i and $w_k(.)$ is the kth Gaussian kernel:

$$w_k(\boldsymbol{u}(i,j)) = \exp\left(-\frac{1}{2}\left(\boldsymbol{u}(i,j) - \boldsymbol{\mu}_k\right)^T \boldsymbol{\Sigma}_k^{-1}\left(\boldsymbol{u}(i,j) - \boldsymbol{\mu}_k\right)\right),$$

and $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ are learnable 2×1 mean vector and 2×2 covariance matrix, respectively. For each node i, $\mathrm{f}'_k(i)$ is a weighted sum of the neighbouring semantic representation $\boldsymbol{X}$ and the Gaussian kernel $w_k(.)$ encodes the spatial informations of regions. Then $\mathrm{f}'_k(i)$ for each node is concatenated over K kernels and go through a linear transformation $\boldsymbol{L} \in \mathbb{R}^{E \times (D+1)}$: $\boldsymbol{h}_i = \boldsymbol{L}[\mathrm{f}'(i)]$, and E is the dimension of the output enhanced feature for each region. Finally, the $\boldsymbol{h}_i$ for each region is concatenated to the original region features $\boldsymbol{f}_i$ to improve both classification and localization.

3.5. SGRN for Multiple Domains

In recent years, a few open large detection datasets have appeared with a different number of categories. For example, MSCOCO has 80 categories and VisualGenome has 3000 categories. However, detectors are typically trained under full supervision and have to be retrained when facing new dataset or new categories. This is tedious and very time-consuming. Since our relation graph learner module and spatial-aware graph reasoning module can be reused in different datasets, we are particularly interested in the domain transferability of SGRN. Specifically, to train a new model on a new dataset, we first copy all the parameters including SFRN from the model trained from the source dataset except the bbox regression and classification layer. The weights $\mathbf{W}_{source}$ of the bbox regression and classification layer can be transformed to the target dataset by $\mathbf{W}_{target} = \Gamma \mathbf{W}_{source}$, where $\Gamma \in \mathbb{R}^{C_{target} \times C_{source}}$. The Γ is a transform matrix which can be obtained by calculating the cosine distance between the category name word embedding [2, 16]. Experiments of transferring from multiple datasets the can be found in Section 4.5. Our SGRN shows great transfer capability which can be used to shorten the training schedule.

4. Experiments

4.1. Datasets and Evaluation.

We first conduct experiments on large-scale object detection benchmarks with a large number of classes: Visual Genome (VG) [25] and ADE [53]. Note that these two datasets have long-tail distributions. The task is to localize an object and classify it, which is different from the experiments with given ground truth locations in Chen et al. [6]. For VG, we use the synsets [46] instead of the raw names of the categories due to inconsistent label annotations, following [21, 6, 23]. We consider two set of target classes: 1000 most frequent classes and 3000 most frequent classes:VG$_{1000}$ and VG$_{3000}$. We split the remaining 92960 images with objects on these class sets into 87960 and 5,000 for training and testing, following [23]. For ADE dataset, we use 20,197 images for training and 1,000 images for testing with 445 categories, following [6, 23]. Since ADE is a segmentation dataset, we convert segmentation masks to bounding boxes [6] for all instances.

Moreover, we are curious about whether our SGRN also works on a smaller scale dataset (fewer categories) so that experiments are also conducted on common object detection datasets: MSCOCO [32] with 80 classes. MSCOCO 2017 contains 118k images for training, 5k for evaluation (also denoted as *minival*) as common practice [20, 31, 29].

For all the evaluation, we adopt the metrics from COCO detection evaluation criteria [32], that is, mean Average Precision (mAP) across different IoU thresholds (IoU= $\{0.5 :$

%	Method	AP	AP$_{50}$	AP$_{75}$	AP$_S$	AP$_M$	AP$_L$	AR$_1$	AR$_{10}$	AR$_{100}$	AR$_S$	AR$_M$	AR$_L$
VG$_{1000}$	Light-head RCNN[28]	6.2	10.9	6.2	2.8	6.5	9.8	14.6	18.0	18.7	7.2	17.1	25.3
	Cascade RCNN[4]	6.5	12.1	6.1	2.4	6.9	11.2	15.3	19.4	19.5	6.1	19.2	27.5
	HKRM[23]	7.8	13.4	8.1	4.1	8.1	12.7	18.1	22.7	22.7	9.6	20.8	31.4
	Faster-RCNN[45]	5.7	9.9	5.8	2.7	6.9	8.9	13.8	17.0	17.0	6.6	15.8	23.5
	Faster-RCNN w SGRN	$6.8^{+1.1}$	$11.1^{+1.2}$	$7.1^{+1.3}$	$3.3^{+0.6}$	$7.0^{+0.1}$	$10.8^{+1.9}$	$15.3^{+1.5}$	$19.5^{+2.5}$	$19.6^{+2.6}$	$8.3^{+1.7}$	$17.8^{+2.0}$	$26.7^{+3.2}$
	FPN[30]	7.1	12.9	7.3	4.2	7.9	10.7	14.9	19.8	20.0	11.1	19.3	23.6
	FPN w SGRN	**8.1**$^{+0.9}$	**13.6**$^{+0.7}$	**8.4**$^{+1.1}$	**4.4**$^{+0.2}$	**8.2**$^{+0.3}$	**12.8**$^{+2.1}$	**19.5**$^{+4.6}$	**26.0**$^{+6.2}$	**26.2**$^{+6.2}$	**12.4**$^{+1.3}$	**23.9**$^{+4.6}$	**34.0**$^{+10.4}$
VG$_{3000}$	Light-head RCNN[28]	3.0	5.1	3.2	1.7	4.0	5.8	7.3	9.0	9.0	4.3	10.3	15.4
	Cascade RCNN[4]	3.8	6.5	3.4	1.9	4.8	4.9	7.1	8.5	8.6	4.2	9.9	13.7
	HKRM[23]	4.3	7.2	4.4	2.6	5.5	8.4	10.1	12.2	12.2	5.9	13.0	20.5
	Faster-RCNN[45]	2.6	4.4	2.7	1.7	3.6	4.8	6.2	7.6	7.6	4.3	9.1	12.9
	Faster-RCNN w SGRN	$3.2^{+0.6}$	$5.0^{+0.6}$	$3.4^{+1.3}$	$2.0^{+0.3}$	$4.2^{+0.6}$	$6.5^{+1.7}$	$7.3^{+0.9}$	$9.2^{+1.6}$	$9.2^{+1.6}$	$4.9^{+0.6}$	$11.4^{+1.7}$	$16.2^{+3.3}$
	FPN[30]	3.4	6.1	3.4	2.6	4.8	6.3	6.9	9.1	9.1	6.7	11.5	13.4
	FPN w SGRN	**4.5**$^{+1.1}$	**7.4**$^{+1.3}$	**4.3**$^{+1.0}$	**2.9**$^{+0.3}$	**6.0**$^{+1.2}$	**8.6**$^{+2.3}$	**10.8**$^{+3.9}$	**13.7**$^{+4.6}$	**13.8**$^{+4.7}$	**8.1**$^{+1.4}$	**15.1**$^{+3.6}$	**21.8**$^{+8.4}$
ADE	Light-head RCNN[28]	7.0	11.7	7.3	2.4	5.1	11.2	9.6	13.3	13.4	4.3	10.4	20.4
	Cascade RCNN[4]	9.1	16.8	8.9	3.5	7.1	15.3	12.1	16.4	16.6	6.4	13.8	25.8
	HKRM[23]	10.3	18.0	10.4	4.1	7.8	16.8	13.6	18.3	18.5	7.1	15.5	28.4
	Faster-RCNN[45]	6.9	12.8	6.8	3.1	6.4	12.3	9.3	13.3	13.6	7.9	13.4	20.5
	Faster-RCNN w SGRN	$9.5^{+2.6}$	$15.3^{+2.5}$	$10.1^{+3.3}$	$4.9^{+1.8}$	$8.4^{+2.0}$	$16.0^{+3.7}$	$12.5^{+3.2}$	$17.6^{+4.3}$	$17.7^{+4.1}$	$8.4^{+0.5}$	$16.0^{+2.6}$	$27.3^{+6.8}$
	FPN[30]	10.9	21.0	12.0	7.3	12.1	18.4	13.5	20.3	20.9	13.3	21.9	29.0
	FPN w SGRN	**14.0**$^{+3.1}$	**23.1**$^{+2.1}$	**14.8**$^{+2.8}$	**8.1**$^{+0.8}$	**13.7**$^{+1.6}$	**21.4**$^{+3.0}$	**16.5**$^{+3.0}$	**25.5**$^{+5.2}$	**26.2**$^{+5.3}$	**17.7**$^{+4.4}$	**27.5**$^{+5.6}$	**35.3**$^{+6.3}$

Table 1. Main results of test datasets on VG$_{1000}$(Visual Genome), VG$_{3000}$ and ADE$_{445}$. "w SGRN" is the baseline model Faster-RCNN [45] and FPN [30] adding the proposed SGRN method. Note that comparison of HKRM [23] is not fair since their method here used the relation and attribute annotations of Visual Genome.

$0.95, 0.5, 0.75\}$) and scales (small, medium, big). We also use Average Recall (AR) with different number of given detection per image ($\{1, 10, 100\}$) and different scales (small, medium, big).

4.2. Implementation Details.

We conduct all experiments using Pytorch [42], 8 Tesla V100 cards on a single server. ResNet-101 [18] pretrained on ImageNet [46] is used as our backbone network. We use two widely-adopted state-of-the-art detection methods i.e. Faster-RCNN [45] and FPN [30] as our baselines.

Faster-RCNN [45]. The hyperparameters in training mostly follow [45]. The parameters before conv2 are fixed, same with [28]. During training, we augment with flipped images and multi-scaling (pixel size=$\{600 \sim 1000\}$). During testing, pixel size= 600 is used. Following [45], RPN is applied on the conv4 feature maps. The total number of proposed regions after NMS is $N_r = 128$. Features in conv5 are avg-pooled to become the feature vector for each proposed regions and feed directly into the bbox regression and classification layers. We use the final conv5 for 128 regions after avg-pool (D= 2048) as the proposals visual features for our relation learner module's inputs. Class agnostic bbox regression is not adopted.

FPN[30]. The hyper-parameters in training mostly follow [30]. During testing, pixel size= 800 is used. RPN is applied to all the feature maps. The total number of proposed regions after NMS is $N_r = 512$. Features in conv5 are avg-pooled to become the proposals visual features (D= 512) as our relation learner module's inputs. In the bbox-head, 2 shared FC layer is used for the proposals visual features and the output is a 1024-d vector feed into the bbox regression and classification layers. Class agnostic bbox regression is adopted. Unless otherwise noted, settings are same for all experiments.

SGRN. In the relation learner module, we use two linear layers of size 256 to learns the latent $\mathbf{Z}$ ($L = 256$) in Equation (1) and most $t = 32$ relevant nodes is retained. Visual embeddings of the regions are collected from the classification layer (2048-d for Faster RCNN, 1024-d for FPN). For the spatial-aware graph reasoning module, we use two spatial-aware graph convolution layers with dimensions 512 and 256 so that the output size of the module for each region is $E = 256$. All linear layers are activated using Rectified Linear Unit (ReLU) activation functions.

For all training, stochastic gradient descent(SGD) is performed on 8 GPUs with 2 images on each. The initial learning rate is 0.02, reduce three times ($\times 0.01$) during finetuning; 10^{-4} as weight decay; 0.9 as momentum. For all the datasets, we train 24 epochs for both baselines (Further training after 12 epochs won't increase the performance of baseline). For our SGRN, we use 12 epochs of the baseline as the pretrained model and train another 12 epochs with same settings with baseline. We also implement and compare the methods in Table 1 using the publicly released code. For fair comparisons, we do not use soft-nms or on-

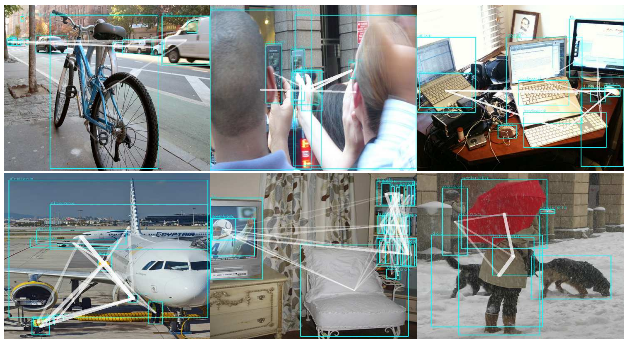

Figure 4. Examples of the learned graph structures from our SGRN. The centers of regions are plotted and connected by the learned graph edges. Edges thickness correspond to the strength of the graph edge weights.

%	Method	mAP	AP_{50}	AP_{75}
MSCOCO	SIN [34]	23.2	44.5	22.0
	Relation Network[20]	38.8	60.3	42.9
	Spatial Memory Network[5]	31.6	52.2	33.2
	RetinaNet[31]	39.1	59.1	42.3
	DetNet[29]	40.2	62.1	43.8
	IoUNet[22]	40.6	59.0	49.0
	HKRM[23]	37.8	58.0	41.3
	FPN[30]	38.3	60.5	42.0
	FPN w SGRN	**41.7**$^{+3.4}$	**62.3**$^{+1.8}$	**45.5**$^{+3.5}$

Table 2. Comparison of mean Average Precision (mAP), AP_{50} and AP_{75} on MSCOCO. "FPN w SGRN" is the the proposed SGRN method based on FPN[30]. The accuracy number except FPN and our method are directly from the original paper.

line hard example mining (OHEM) in all experiments.

4.3. Comparison with state-of-the-art

Large-scale Detection Benchmarks. We first evaluate our SGRN method on large-scale detection benchmarks: VG_{1000} (Visual Genome with 1000 categories), VG_{3000} with 3000 categories and ADE with 445 categories. Table 1 shows the result of our method SGRN added on the baseline model Faster-RCNN [45] and FPN [30]. Our SGRN achieves significant gains on both classification and localization accuracy than the baseline on all the large-scale detection benchmarks. Our method achieve an overall AP of 8.1% compared to 7.2% by FPN on VG_{1000}, 4.5% compared to 3.4% on VG_{3000}, and 14.0% compared to 10.9% on ADE, respectively. From Table 1, it can be found that our method can boost the average precision of 0.6% to 3.0%. Our method works mostly on the large item (AP_L:+2% $\sim$ 3%). Furthermore, our SGRN is able to improve upon FPN by a margin of 3%~10% on average recall (e.g. AR_L:+10.4% for VG_{1000}). This demonstrates that our method can improve both the accuracy and false discovery rate by the spatial-aware graph reasoning. We also compare our method to Light-head RCNN [28], Cascade-RCNN [4] and HKRM [23] in Table 1. Note that the method of HKRM uses the relation and attribute annotation of Visual Genome so that the comparison is not fair. From the table, the SGRN outperforms other competing methods by a large margin (relatively 10%~80%).

Figure 4 shows visual examples of the learned graph structures from our SGRN. The centers of regions are plotted and connected by the learned graph edges. Edges thickness correspond to the strength of the graph edge weights. Our method learned interpretable edges between regions. For example, objects with the same category are connected such as "books", "people", "cars". Their visual features thus are shared and enhanced. Furthermore, objects with semantic relationship or co-occurrence are connected e.g. "mouse" and "laptop"; "umbrella" and "person"; "bicycle" and "car" and "cellphone" and "people". The correct learned edges help the successful graph learning thus lead to better detection performance. Figure 5 also shows the qualitative comparisons of the baseline method FPN and our SGRN on Visual Genome with 1000 categories. Our method is more accurate than the baseline method due to the help of the encoded relationship. For example, SGRN can detect the "doughnut", "catcher", "food" and "sign" while FPN cannot. FPN detects some false positive bbox such as "paw" and "leg" in the cat image.

Common Detection Benchmark. We further investigate the performance of our SGRN a smaller scale dataset (fewer categories). Table 2 shows the performances on the *minival* of the MSCOCO dataset with 80 categories. To compare with the state-of-art methods, we also report the accuracy numbers of SIN [34], Spatial Memory Network [5], Relation Network [20], RetinaNet [31], DetNet [29], IoUNet [22] and HKRM [23] directly from the original paper. Note that our implementation of the baseline FPN has higher accuracy than that in original works (38.3 vs 36.2 [30]). As can be seen, our method performs 3.4% better than the baseline FPN and all the other competitors. This demonstrates that our SGRN method can also work on the dataset with a smaller number of categories by improving the feature representation due to its ability of relation graph reasoning.

4.4. Ablative Analysis

To perform a detailed ablative analysis, we have conducted experiments with FPN baseline. Table 3 shows the effectiveness of different components in our model on ADE: 1) The graph convolution of visual embedding is the most important component of our method, accounting for 1.9% improvement on FPN. 2) Reason with a fully connected graph shows redundant edges lead performance drop, while the sparse connection can improve the mAP by around 0.3%. 3) The design of spatial Gaussian Kernels can improve the mAP by 0.5%. 4) Using soft-mapping $\mathbf{M}^s$ in-

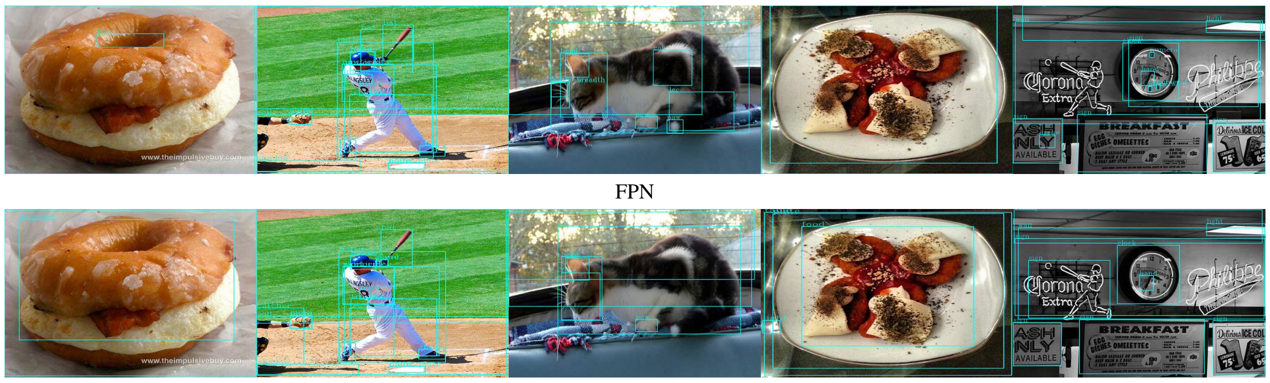

Figure 5. Qualitative comparisons of the baseline method FPN and our SGRN on VG_{1000}. Our method is more accurate than the baseline method due to the help of the encoded relation.

stead of hard-mapping can further increase the overall AP by 0.4%.

%	GCN-Visual Embedding	Sparse Connection	Spatial Gaussian	Soft-Mapping	AP	AP_{50}	AR_1	AR_{10}
FPN					10.9	21.0	13.5	20.3
	√				$12.8^{+1.9}$	$21.9^{+0.9}$	$14.5^{+1.0}$	$23.5^{+3.2}$
	√	√			$13.1^{+2.2}$	$21.6^{+0.6}$	$15.4^{+1.9}$	$23.9^{+3.6}$
	√	√	√		$13.6^{+2.7}$	$23.0^{+2.0}$	$15.1^{+1.6}$	$24.3^{+4.0}$
	√	√		√	$13.4^{+2.5}$	$21.8^{+0.8}$	$16.0^{+2.5}$	$23.7^{+3.4}$
	√		√	√	$13.7^{+2.8}$	$22.5^{+1.5}$	$16.2^{+2.7}$	$24.7^{+4.4}$
SGRN	√	√	√	√	**$14.0^{+3.1}$**	**$24.1^{+2.1}$**	**$16.5^{+3.0}$**	**$25.5^{+5.2}$**

Table 3. Ablation Study based on different components of our model on ADE. Our final model SGRN is reported in the last line. The backbones are ResNet-101 with FPN.

4.5. Domain Transferability of SGRN

We are interested in the domain transferability of SGRN e.g. transferring the trained SGRN model between multiple datasets. Table 4 shows the comparison of our transferred model and the baseline FPN. We transfer our trained model from the source dataset to the target dataset by the method mentioned in Section 4.5. We only train our SGRN for another 5 epochs for adaption to the new dataset. From the table, it can be found that our SGRN can perform better than the baseline FPN even we only trained the model for 5 epochs. And we also try FPN transferred from MSCOCO to ADE with the same setting of SGRN and trained for 5 epochs. Its mAP is only 8.6% while our SGRN can reach the same mAP with only 1 epoch training. Additional experiments in Table 4 shows the results of SGRN with frozen all layers except the weight transfer layers, denoted as Frozen-SGRN. The performance only suffers a minor drop comparing to the original results. If we froze the same layers for FPN based on ImageNet pretrained backbone (indicated as Frozen-FPN) the results deteriorate. This demonstrates the effectiveness of the domain transferability by reusing the graph learner and graph reasoning module.

Method	Target Dataset	Source Dataset	Training Epochs	AP	AP_{50}	AP_{75}	AR_1	AR_{10}	AR_{100}
FPN	VG_{3000}	-	12	3.4	6.1	3.4	6.9	9.1	9.1
SGRN	VG_{3000}	VG_{1000}	1	3.4	5.7	3.5	7.4	9.6	9.6
SGRN	VG_{3000}	VG_{1000}	5	**4.6**	**7.5**	**4.8**	**10.2**	**13.5**	**13.5**
SGRN	VG_{3000}	COCO	1	2.1	3.7	2.2	4.7	6.6	6.7
SGRN	VG_{3000}	COCO	5	3.4	5.7	3.5	7.8	10.7	10.8
Frozen-FPN	COCO	-	12	28.7	50.7	28.6	25.6	42.5	45.4
Frozen-SGRN	COCO	VG_{1000}	1	33.1	52.3	36.5	29.5	45.0	46.4
Frozen-SGRN	COCO	VG_{1000}	5	37.7	57.8	40.8	31.9	50.6	35.1
FPN	COCO	-	12	38.3	60.5	42.0	32.1	50.8	35.4
SGRN	COCO	VG_{1000}	1	33.0	51.3	36.5	29.5	45.9	47.3
SGRN	COCO	VG_{1000}	5	**38.9**	**58.7**	**42.6**	**32.9**	**53.9**	**57.2**
Frozen-FPN	ADE	-	12	7.8	16.1	7.7	10.0	15.5	16.0
Frozen-SGRN	ADE	VG_{1000}	1	6.7	11.3	7.0	8.4	12.4	12.5
Frozen-SGRN	ADE	VG_{1000}	5	10.9	17.4	11.8	13.5	20.1	20.5
FPN	ADE	-	12	10.9	21.0	12.0	13.5	20.3	20.9
FPN	ADE	COCO	5	8.6	16.5	10.1	10.7	16.6	17.1
SGRN	ADE	VG_{1000}	1	8.6	15.3	9.3	11.2	16.0	16.2
SGRN	ADE	VG_{1000}	5	**12.9**	**20.4**	**13.8**	**16.1**	**22.8**	**23.2**
SGRN	ADE	COCO	5	11.9	19.3	13.0	14.3	22.4	23.1

Table 4. Domain Transferability of our SGRN. FPN is the implemented baseline method with backbone pretrained from ImageNet. We transfer the trained SGRN model between multiple datasets by the method in Section 4.5. We only train our SGRN for another 5 epochs for adaption to the new dataset. The result of training with frozen all layers except the weight transfer layers is also reported denoted as Frozen-SGRN.

5. Conclusions

In this work, we proposed a new spatial-aware graph relation network (SGRN) for encoding object relation in the detection system without any external knowledge. Our method is in-place and easily plugged into any existing detection pipeline for endowing its ability to capture and unify semantic and spatial relationships.

References

[1] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid. Label-embedding for attribute-based classification. In *CVPR*, 2013. 3

[2] J. Almazán, A. Gordo, A. Fornés, and E. Valveny. Word spotting and recognition with embedded attributes. *IEEE transactions on pattern analysis and machine intelligence*, 36(12):2552–2566, 2014. 3, 5

[3] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017. 3

[4] Z. Cai and N. Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *CVPR*, 2018. 6, 7

[5] X. Chen and A. Gupta. Spatial memory for context reasoning in object detection. In *ICCV*, 2017. 2, 7

[6] X. Chen, L.-J. Li, L. Fei-Fei, and A. Gupta. Iterative visual reasoning beyond convolutions. In *CVPR*, 2018. 2, 3, 4, 5

[7] B. Dai, Y. Zhang, and D. Lin. Detecting visual relationships with deep relational networks. In *CVPR*, 2017. 2, 3, 4

[8] J. Dai, Y. Li, K. He, and J. Sun. R-fcn: Object detection via region-based fully convolutional networks. In *NIPS*, 2016. 2

[9] M. Defferrard, X. Bresson, and P. Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *NIPS*, 2016. 3

[10] J. Deng, N. Ding, Y. Jia, A. Frome, K. Murphy, S. Bengio, Y. Li, H. Neven, and H. Adam. Large-scale object classification using label relation graphs. In *ECCV*, 2014. 3

[11] 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. 1, 2

[12] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. In *CVPR*, 2009. 1

[13] 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, 2010. 3

[14] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, T. Mikolov, et al. Devise: A deep visual-semantic embedding model. In *NIPS*, 2013. 3

[15] C. Galleguillos, A. Rabinovich, and S. Belongie. Object categorization using co-occurrence, location and appearance. In *CVPR*, 2008. 3

[16] C. Gong, D. He, X. Tan, T. Qin, L. Wang, and T.-Y. Liu. Frage: Frequency-agnostic word representation. In *NIPS*, 2018. 4, 5

[17] S. Gould, T. Gao, and D. Koller. Region-based segmentation and object detection. In *Advances in Neural Information Processing Systems 22*, 2009. 1

[18] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 2, 6

[19] J. Hoffman, S. Guadarrama, E. S. Tzeng, R. Hu, J. Donahue, R. Girshick, T. Darrell, and K. Saenko. Lsda: Large scale detection through adaptation. In *NIPS*, 2014. 1

[20] H. Hu, J. Gu, Z. Zhang, J. Dai, and Y. Wei. Relation networks for object detection. In *CVPR*, 2018. 2, 3, 5, 7

[21] R. Hu, P. DollÃ¡r, K. He, T. Darrell, and R. Girshick. Learning to segment every thing. In *CVPR*, 2018. 5

[22] B. Jiang, R. Luo, J. Mao, T. Xiao, and Y. Jiang. Acquisition of localization confidence for accurate object detection. In *ECCV*, 2018. 7

[23] C. Jiang, H. Xu, X. Liang, and L. Lin. Hybrid knowledge routed modules for large-scale object detection. In *NIPS*, 2018. 1, 2, 3, 5, 6, 7

[24] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017. 2, 3, 4

[25] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, M. Bernstein, and L. Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 2016. 1, 2, 5

[26] C. H. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *CVPR*, 2009. 3

[27] K.-H. Lee, X. He, L. Zhang, and L. Yang. Cleannet: Transfer learning for scalable image classifier training with label noise. In *CVPR*, 2018. 4

[28] Z. Li, C. Peng, G. Yu, X. Zhang, Y. Deng, and J. Sun. Light-head r-cnn: In defense of two-stage object detector. In *CVPR*, 2017. 6, 7

[29] Z. Li, C. Peng, G. Yu, X. Zhang, Y. Deng, and J. Sun. Detnet: A backbone network for object detection. In *ECCV*, 2018. 5, 7

[30] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 2, 6, 7

[31] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. *IEEE transactions on pattern analysis and machine intelligence*, 2018. 2, 5, 7

[32] 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 *ECCV*, 2014. 1, 2, 5

[33] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *ECCV*, 2016. 2

[34] Y. Liu, R. Wang, S. Shan, and X. Chen. Structure inference net: Object detection using scene-level context and instance-level relationships. In *CVPR*, 2018. 2, 3, 7

[35] J. Mao, X. Wei, Y. Yang, J. Wang, Z. Huang, and A. L. Yuille. Learning like a child: Fast novel visual concept learning from sentence descriptions of images. In *ICCV*, 2015. 3

[36] K. Marino, R. Salakhutdinov, and A. Gupta. The more you know: Using knowledge graphs for image classification. In *CVPR*, 2017. 2, 3, 4

[37] I. Misra, A. Gupta, and M. Hebert. From red wine to red tomato: Composition with context. In *CVPR*, 2017. 3

[38] F. Monti, D. Boscaini, J. Masci, E. Rodola, J. Svoboda, and M. M. Bronstein. Geometric deep learning on graphs and manifolds using mixture model cnns. In *CVPR*, 2017. 3, 5

[39] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, and A. Yuille. The role of context for object detection and semantic segmentation in the wild. In *CVPR*, 2014. 3

[40] M. Niepert, M. Ahmed, and K. Kutzkov. Learning convolutional neural networks for graphs. In *ICML*, pages 2014–2023, 2016. 2

[41] W. Norcliffe-Brown, E. Vafeais, and S. Parisot. Learning conditioned graph structures for interpretable visual question answering. In *NIPS*, 2018. 3

[42] 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 Workshop*, 2017. 6

[43] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, 2016. 2

[44] S. Reed, Z. Akata, H. Lee, and B. Schiele. Learning deep representations of fine-grained visual descriptions. In *CVPR*, 2016. 3

[45] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. 2, 6, 7

[46] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. 5, 6

[47] R. Salakhutdinov, A. Torralba, and J. Tenenbaum. Learning to share visual appearance for multiclass object detection. In *CVPR*, pages 1481–1488. IEEE, 2011. 1, 4

[48] A. Torralba, K. P. Murphy, and W. T. Freeman. Sharing features: efficient boosting procedures for multiclass object detection. In *CVPR*, 2004. 1

[49] A. Torralba, K. P. Murphy, W. T. Freeman, and M. A. Rubin. Context-based vision system for place and object recognition. In *ICCV*, 2003. 1

[50] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio. Graph attention networks. In *ICLR*, 2017. 3

[51] X. Wang, R. Girshick, A. Gupta, and K. He. Non-local neural networks. In *CVPR*, 2018. 2

[52] X. Wang, Y. Ye, and A. Gupta. Zero-shot recognition via semantic embeddings and knowledge graphs. In *CVPR*, 2018. 4

[53] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba. Scene parsing through ade20k dataset. In *CVPR*, 2017. 2, 5