Context for Object Detection via Lightweight Global and Mid-level Representations

Mesut Erhan Unal
School of Computing and Information
University of Pittsburgh
Pittsburgh, Pennsylvania 15260
Email: erhanunal@pitt.edu

Adriana Kovashka
School of Computing and Information
University of Pittsburgh
Pittsburgh, Pennsylvania 15260
Email: kovashka@cs.pitt.edu

***Abstract*—We propose an approach for explicitly capturing context in object detection. We model visual and geometric relationships between object regions, but also model the global scene as a first-class participant. In contrast to prior approaches, both the context we rely on, as well as our proposed mechanism for belief propagation over regions, is lightweight. We also experiment with capturing similarities between regions at a semantic level, by modeling class co-occurrence and linguistic similarity between class names. We show that our approach significantly outperforms Faster R-CNN, and performs competitively with a much more costly approach that also models context.**

I. Introduction

Context is an important mechanism that makes visual recognition easy for humans [1]–[3]. It is natural to also model context in machine perception. Both classic recognition work and recent neural techniques have incorporated context [4]–[10]. However, prior work does this in a manner which is expensive from both a computational and human labeling point of view. One line of work models context through the use of a knowledge graph constructed at the semantic (category) level [10]–[12]. To construct it, authors rely on human annotations, e.g. the 2.3 million subject-verb-object relations in the Visual Genome dataset. Another line of research [9], [13], [14] relies on visual and geometric similarity between regions. [9] propagate belief over the features of image regions through a gating mechanism, namely Gated Recurrent Units (GRUs) [15]. However, GRUs require a large number of parameters. Finally, many prior works [8], [9], [12] model relations between regions only, but do not propagate these relations to update the global, image/scene-level representation.

We propose a novel approach for context-aware object detection that tackles the problems mentioned above. First, rather than modeling class similarity through expensive annotations, we propose to model it through mid-level attributes as well as linguistic similarity between class names. Second, we show that expensive GRUs can be replaced with a Graph Convolutional Network (GCN) [16] to achieve comparable performance with a much smaller number of parameters. Third, we demonstrate that propagating belief to the global scene level, which in turn affects region-level features, greatly improves performance.

In more detail, we experiment with a suite of techniques for modeling context between object regions. The first set of those use a variant of Structure Inference Net (SIN) [9] as their basis. We add similarity between regions in terms of predicted attributes to these regions, to the computation of region-to-region edge weights in SIN. We also propagate belief between regions and a global scene node. In a second framework, we replace the gated belief propagation in SIN with two GCNs that separately model visual and geometric relationships. The resulting model is six times more efficient than SIN in terms of number of parameters, and achieves competitive results. We add a global scene node to one of these GCNs, as well as additional modeling of the linguistic (word embedding) similarity between classes, such that a model receives extra penalty if it confuses semantically related classes (which may also be easy to confuse visually).

We show the benefit of each of the techniques we propose: global representation capture, category similarity modeling through mid-level attributes and word embeddings, and the use of GCN instead of GRU. We conduct experiments on two established object detection datasets, PASCAL VOC and COCO. We show that our approach significantly outperforms the standard detection algorithm, Faster R-CNN [17]. We also show it achieves comparable results with the recent but more expensive Structure Inference Net [9]. We show that for several categories and super-categories, our method outperforms the more expensive SIN.

II. Related work

Implicit context modeling: Standard detection methods [17]–[19] implicitly capture context, e.g. through shared convolution weights, expansion of the perimeter of boxes, sharing of parameters at different scales, etc. Some general-purpose works [20]–[22] complement convolutional networks with global context and cross-region coordination. Instead, we show that explicit, pairwise region context modeling improves performance beyond the implicit modeling that detection methods such as [17] provide.

Using graphs to help detection: The most related work to ours is Structure Inference Net [9], which performs object detection by relying on context from other objects and the scene. [9] formulate a graph connecting regions output by Faster R-CNN's Region Proposal Network (RPN). Feature representations are propagated over the edges of the graph,

through a gating mechanism: specifically, Gated Recurrent Unit (GRU). [8] is similar, but unlike [9] does not model the scene (image) level; it also relies on expensive iterative memory updates and needs an explicit iteration limit. Another work [23] uses context to refine proposal features, but does so through simple weighted concatenation; in contrast, we explicitly model pairwise relationships between regions, which increases the interpretability of our method. [24] model relations between objects through attention but do not model the global scene level, which we show is critical in our work.

Graphs for related tasks: Other methods also use graphs to model context, but perform different types of detection tasks than our goal. For example, [10], [11], [13], [25] construct a knowledge graph at the semantic (category) rather than region level, but evaluate in the large-scale detection setting (Visual Genome and ADE datasets) rather than common object detection (PASCAL VOC, COCO) as we do. Moreover, [10] evaluate their model on a region classification task, using provided bounding-box coordinates in both training and testing. Several of these also use additional computation or "bells and whistles". For example, [13] use an attention model and an intricate framework with many steps; they show in ablation studies that each of the many modules of their network aid performance. Similarly, [25] compute semantic relationships from annotations on the Visual Genome dataset [26] (e.g. subject-object-verb, spatial/prepositional relations) which are expensive to obtain and not available on the datasets we consider. In contrast, we propose a set of simpler models that rely on visual, geometric, or mid-level (attribute) similarity. [12] use bounding box annotations for a set of source categories but only have image-level labels for target categories. While they consider weakly supervised detection and compare to Fast R-CNN [27] as an upper bound, we consider fully supervised detection and show that we outperform the stronger Faster R-CNN [17] model. [7] use semantic segmentation as an auxiliary task for detection, but this requires additional annotations. There are also relevant works in the pre-convnet era, which model diverse context ranging from geometric to cultural [28], expressed using support trees [6], HMMs [29] or MRFs [30], and spatially organized SIFT features [31].

Contextual graphs for tasks beyond object detection: Context has been modeled, often through graphs, for other recognition tasks beyond detection. For example, [32] devise a relation proposal network and attention graph convolutional network to prune scene graph edges. [33], [34] use graphs to compute human-object interactions. [35] use a knowledge graph to aid in semantic navigation, and [36] use one to reason about object affordances. [37]–[39] use information from a knowledge graph for zero-shot classification, and [40] use graphs for domain adaptation. None of these methods show object detection results.

III. Approach

Our work builds on Structure Inference Net (SIN) [9]. We extend this work through three main techniques: (1) semantic similarity between object categories grounded in mid-level representations i.e. attributes, (2) belief propagation at the global scene level, (3) replacement of the GRU mechanism for belief propagation with GCN, to achieve a simpler and more efficient model, and (4) modeling of class similarity via embeddings of class names. We describe our baseline model in Sec. III-A. We then describe the global scene updates in Sec. III-B, our mid-level semantic similarity in Sec. III-C, the GCN adaptation in Sec. III-D, and modeling linguistic similarity in Sec. III-E. We conclude with implementation details in Sec. III-F.

A. Baseline method

Our baseline model builds on SIN, but adds modeling of semantic category co-occurrences. SIN uses GRU cells as write functions to update region feature representations, by incorporating signals from other regions as well as a global node, but not updating the global node. When computing the representation h_t, a GRU cell takes information from the previous hidden state h_{t-1} and current input x_t. In SIN, h^i_{t-1} is simply the region i's state at time $t-1$ and $h^i_0 = f^i$ where f^i is the FC6 output for the region i. Further, x^i_t are messages m^i_t passed from (a) the global scene node, or (b) other regions. This results in two GRUs which we call Scene GRU and Edge GRU, respectively. In the following, we omit the subscript t.

Messages are computed as follows. For (a), $m^i = f^s$, where f^s is the FC6 output for the RoI-pooled whole image. For (b), $m^i = \max pool_{j \in \mathcal{V}}(e_{j \to i} * f^j)$, where $\mathcal{V}$ is the set of regions proposed by Region Proposal Network (RPN), and $K = |\mathcal{V}|$. The edge weights $e_{i \to j}$ are computed from both visual and geometric cues. In particular, in the original SIN paper, the edge weight from region i to region j is computed as $e_{i \to j} = \text{ReLU}(W_g R_{i \to j}) * \tanh(W_v[f^i, f^j])$. In this formulation, the first part captures *geometric* and the latter captures *visual* relationships between regions. Pairwise spatial features are defined as:

$$R_{i \to j} = [w_i, h_i, s_i, w_j, h_j, s_j, (\frac{x_i - x_j}{w_j}), (\frac{y_i - y_j}{h_j}), (\frac{x_i - x_j}{w_j})^2, (\frac{y_i - y_j}{h_j})^2, \log(\frac{w_i}{w_j}), \log(\frac{h_i}{h_j})] \quad (1)$$

where w_i, h_i, s_i are width, height, area of region i, (x_i, y_i) are coordinates of the region center. The final region representation is computed as the mean of the hidden layer in the Scene GRU and Edge GRU.

In our approach, we first add an additional component, namely capturing similarity of regions at a semantic level. We model the co-occurrence of object categories based on the model's best set of guesses about the category in the region. This is an intuitive relationship to capture since a human similarly uses knowledge about co-occurring object categories to quickly perform recognition. Prior work has used category relationships but has leveraged more expensive annotations, as discussed above. In particular, we modify the edge computation as follows:

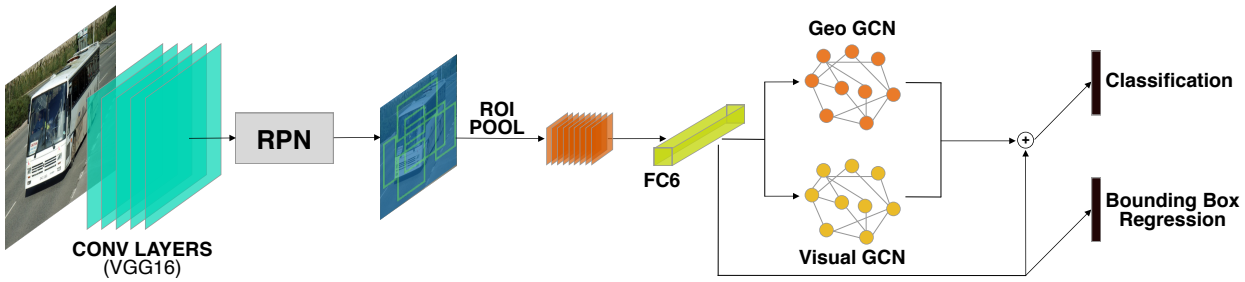

Fig. 1. An overview of our most efficient approach. We model context between regions through single-layer GCNs which capture visual and geometric relationships.

$$e_{i\to j} = \text{ReLU}(W_g R_{i\to j}) * \tanh(W_v[f^i, f^j]) \\ *\text{ReLU}(W_o O_{i\to j}) \quad (2)$$

The term $O_{i\to j} \in \mathcal{R}^{25}$ is the flattened version of a 5×5 row-normalized class co-occurrence matrix in which rows are top-5 predicted classes C_i for region i and columns are top-5 predicted classes C_j for region j (*ignoring the background class*). This model and following two variants require class predictions for regions at training time as they utilize class-level semantics for regions. Due to this fact, we place the structure inference module after the R-CNN head, and train it separately using a fully-trained Faster R-CNN as backbone detector. We use the current predictions from Faster R-CNN for C_i and C_j. This completes our baseline model, which we refer to as BASE in Sec. IV.

B. Scene updates

Our first insight is that since the scene representation is used to propagate belief towards region representations, then these region representations should also be used to update the scene representation. In particular, we add another GRU cell that updates f^s using new region states after each unroll of Edge GRU and Scene GRU. Specifically, this new GRU takes mean-pooled new region states as input and changes its hidden state accordingly. Its hidden state is initialized as f^s, and gets updated in each unroll. Then, this updated scene representation is used as input to Scene GRU in its next unroll. We show in our experiments that this approach, which we refer to as SCENE, improves the performance of BASE greatly. This is our only approach which is computationally more expensive than SIN.

C. Attribute-based similarity

Further, we incorporate class similarity based on attributes. We compute similarity between regions based on their highest class predictions, and the co-occurrence of these attributes and classes. We first build a row-normalized attribute matrix $A \in \mathcal{R}^{C\times J}$where C is number of classes, and J is the length of the attribute dictionary. We then project this matrix onto a $16D$ latent space $M \in \mathcal{R}^{C\times 16}$. As attribute vocabularies usually contain highly correlated entries (e.g. eye, nose, head, mouth, face in [41] for the PASCAL dataset), we consider this projection as an essential step towards re-expressing them more compactly. Furthermore, we let the network learn the best projection while training instead of applying an offline dimensionality reduction.

After projection, we multiply our new attribute matrix M with its transpose, resulting in $Q = MM^T$, then Q_{c_i,c_j} corresponds to region i and region j's attribute similarity based on their highest predicted classes c_i and c_j (using Faster R-CNN's current predictions). The final edge weight calculation is based on the formula:

$$e_{i\to j} = \text{ReLU}(W_g R_{i\to j}) * \tanh(W_v[f^i, f^j]) \\ *\text{ReLU}(W_o O_{i\to j}) * Q_{c_i,c_j} \quad (3)$$

We refer to the above method as ATTR1. We also propose an alternative, ATTR2. In this approach, we first map attributes to regions based on the regions' predicted class scores, by multiplying region scores with the attribute matrix $\mathcal{A}$. We then project this result onto a latent space $B \in \mathcal{R}^{K\times 16}$, where K is the number of regions in the image. After projection, we multiply this region-attribute matrix B by its transpose, to get a matrix $Z = BB^T$ such that $Z_{i,j}$ captures the similarity of regions i and j. The edge weights are then computed as:

$$e_{i\to j} = \text{ReLU}(W_g R_{i\to j}) * \tanh(W_v[f^i, f^j]) \\ *\text{ReLU}(W_o O_{i\to j}) * Z_{i,j} \quad (4)$$

Class-level attribute representations have been employed in zero-shot learning [42], [43] to bridge the gap between seen and unseen classes, to transfer knowledge for weakly-supervised detection in [12], and to supervise learning of region graph structure for fully-supervised detection in [11]. However, this work relies on the expensive annotations of Visual Genome and ImageNet Attributes [26], [44] while we use annotations of much smaller scale [41].

D. GCN instead of GRU

Next, we show that the expensive GRU approach is not necessary to achieve an improvement over Faster R-CNN. We model context in a simpler, more light-weight fashion, through a Graph Convolutional Network (GCN). Specifically, we use two GCNs, one for message passing based on visual dependencies between regions (referred to as Visual GCN)

and one for spatial dependencies between regions (Geo GCN). Finally, we take the mean of the representations from the two GCNs, and use it as the final region representations for the classification task in Faster R-CNN. These final region representations are obtained as:

$$F' = F + \frac{1}{2}(\text{ReLU}(\tilde{A}_g F W_g) + \text{ReLU}(\tilde{A}_v F W_v)) \quad (5)$$

where $F \in \mathcal{R}^{K \times 4096}$ denotes FC6 output wherein each region's feature $f^{1:K} \in \mathcal{R}^{4096}$ reside as row vector, $W_g, W_v \in \mathcal{R}^{4096 \times 4096}$ are learnable weight matrices for feature projection, $\tilde{A}_g, \tilde{A}_v \in \mathcal{R}^{K \times K}$ are adjacency matrices normalized by applying softmax on each row: $\tilde{A}_{g_i} = \text{softmax}(A_{g_i})$, and unnormalized adjacency matrices, A_g and A_v, are defined as:

$$A_{g_{i,j}} = U_g^T R_{i \to j}, \qquad A_{v_{i,j}} = U_v^T [f^i, f^j] \quad (6)$$

where $U_g \in \mathcal{R}^{13}$ and $U_v \in \mathcal{R}^{8192}$ are learnable weights, f^i and f^j are FC6 outputs for region i and j, $[\,,\,]$ denotes vector concatenation. Lastly, $R_{i \to j} \in \mathcal{R}^{13}$ is a vector that encodes pairwise spatial features and is identical to the formulation in [9], with concatenation of $\text{IoU}_{i,j}$.

We would like to note that final region representations, F', are used for classification only. For the bounding box regression task, we use FC6 features as we empirically found that it yields better performance (74.2% vs 74.9% mAP on VOC07 test). We refer to this model as GEOVIS.

However, we show that incorporating the global representation of the scene as a node in the graph, and thus propagating messages between region nodes and the scene node, improves performance (74.9% vs 75.4% mAP on VOC07 test). In particular, we formulate a method that adds the FC6 feature of the whole image as the scene representation in the Visual GCN. More specifically, we can formulate new region representations as:

$$F' = F + \frac{1}{2}(\text{ReLU}(\tilde{A}_g F W_g) + \langle \text{ReLU}(\tilde{A}_v^* \langle F + f^s \rangle W_v) - f^{s'} \rangle) \quad (7)$$

where $\tilde{A}_v^* \in \mathcal{R}^{(K+1) \times (K+1)}$ is normalized adjacency matrix of the visual graph wherein both regions and the scene reside as nodes, $\langle + \rangle$ denotes vector attachment, and $\langle - \rangle$ denotes vector detachment. We refer to this method as GEOVIS-S.

E. Linguistic similarity loss

Finally, we also experiment with a weighted loss formulation that seeks to increase the ability of a model to discriminate between semantically similar categories based on their pairwise distance in a word embedding space. For example, we wish to penalize cats being classified as dogs, more so than we penalize cats being classified as trucks. The intuition is that the visual discrepancy between semantically unrelated categories (cat and truck) is already large, while the discrepancy between similar categories (cat and dog) is small. As a reminder, Faster R-CNN's optimization objective is to minimize the following multi-task loss formulation when it trains end-to-end (weights are omitted): $\mathcal{L} = \mathcal{L}_{\text{reg}}^{\text{RPN}} + \mathcal{L}_{\text{cls}}^{\text{RPN}} + \mathcal{L}_{\text{reg}} + \mathcal{L}_{\text{cls}}$ (the first two terms denote regression and classification losses of region proposal network and the latter two denote regression and classification losses of the R-CNN part). Our new loss formulation can be written:

$$\mathcal{L} = \mathcal{L}_{\text{reg}}^{\text{RPN}} + \mathcal{L}_{\text{cls}}^{\text{RPN}} + \mathcal{L}_{\text{reg}} + \mathcal{L}_{\text{cls}} + \lambda \mathcal{L}_{\text{sim}} \quad (8)$$

where $\mathcal{L}_{\text{sim}}$ is our additional penalty term for misclassification and averaged over all regions in a minibatch. For a particular region i, it is calculated as:

$$\mathcal{L}_{\text{sim}_i} = [A_{\text{sim}} \hat{Y}_{\text{cls}_i}]^T Y_{\text{cls}_i} \quad (9)$$

In the above formulation, $\hat{Y}_{\text{cls}_i} \in \mathcal{R}^C$ denotes class soft-mapping for the region i, $Y_{\text{cls}_i} \in \mathcal{R}^C$ is one-hot ground-truth label encoding for the region i, and $A_{\text{sim}} \in \mathcal{R}^{C \times C}$ denotes the similarity matrix wherein diagonal entries are zero and other entries are calculated as $A_{a,b} = 1/d_{a,b}$ before row-wise min-max normalization where $d_{a,b}$ denotes Euclidean distance between class labels a and b in GloVe [45] space. Please note that $\mathcal{L}_{\text{sim}} \in [0, 1]$ and we choose $\lambda = 0.1$. We refer to this method as GEOVIS-LING.

F. Implementation details

We implement baselines and our models in Tensorflow[1], employing VGG16 [46] pretrained on ImageNet [47] as the backbone feature extractor. We use NMS [48] in RPN to select 128 boxes as object proposals during training, and 256 boxes during testing. Faster R-CNN is trained with the hyper-parameters suggested in the original paper except the learning rate as we found starting with 5×10^{-4} and lowering it to 5×10^{-5} after 80K iterations while training on VOC 2007 trainval combined with VOC 2012 trainval (VOC07+12) yields better results (73.2% vs 74.7% mAP on VOC07 test). SIN is trained with the suggested hyper-parameters. On VOC07+12 , we train all models 130K iterations with a staircase decay for learning rate ($0.1\times$) after 80K iterations. On COCO 2014 train, we train them 490K iterations with same decaying strategy after 350K iterations. For ATTR1 and ATTR2, we build our class-level attribute matrix, A, by aggregating instance-level attribute annotations given in [41] and normalizing each row so that their L1 norms equal one. All GCN instances employed in our models consist of a single layer as we hypothesize one round of message passing would suffice considering the graphs are dense.

IV. EXPERIMENTAL VALIDATION

We compare the following methods:

1) BASE, the baseline model incorporating visual, geometric and semantic relationships between regions, described in Sec. III-A
2) SCENE, which builds on the baseline but also updates the scene representation (Sec. III-B)

[1]https://github.com/smallcorgi/Faster-RCNN_TF

TABLE I
RESULTS ON PASCAL VOC 2012, WITH TRAINING ON VOC 2007+2012. THE TWO BEST METHODS PER ROW ARE **BOLDED**. THE TOP METHOD IS ALSO UNDERLINED.

	FRCNN	SIN	Scene	Attr1	Attr2	GeoVis-S	GeoVis-Ling
aeroplane	0.767	**0.780**	**0.771**	0.770	0.766	0.760	0.767
bicycle	0.793	**0.798**	**0.796**	0.789	0.789	0.795	0.788
bird	0.733	**0.765**	0.731	0.742	0.750	0.745	**0.757**
boat	0.660	**0.676**	0.668	**0.670**	**0.670**	0.629	0.639
bottle	0.611	**0.625**	0.596	0.600	0.592	**0.613**	0.608
bus	**0.853**	0.851	0.849	**0.852**	0.843	0.849	0.846
car	0.865	0.865	**0.868**	**0.866**	0.856	0.861	0.861
cat	0.881	0.870	0.882	**0.885**	**0.883**	0.881	0.876
chair	0.580	**0.615**	0.574	0.584	0.565	**0.588**	0.575
cow	0.831	0.838	0.837	0.819	0.841	**0.870**	**0.852**
diningtable	0.660	0.691	**0.721**	0.694	0.693	0.677	**0.708**
dog	0.848	0.846	**0.854**	0.852	0.850	**0.854**	0.846
horse	0.859	0.862	0.863	**0.877**	**0.871**	0.866	0.812
motorbike	0.774	**0.788**	**0.776**	0.768	0.768	0.755	0.758
person	0.782	0.786	0.785	**0.787**	**0.787**	0.781	0.778
pottedplant	0.418	**0.509**	0.443	0.429	0.446	**0.482**	0.442
sheep	0.756	**0.771**	0.769	0.752	0.756	0.760	**0.781**
sofa	0.700	**0.756**	0.732	**0.734**	0.732	0.720	0.722
train	0.821	**0.842**	0.839	0.818	**0.842**	0.821	0.825
tvmonitor	0.746	**0.768**	0.762	**0.766**	0.762	0.765	0.758
average	0.747	**0.765**	**0.756**	0.753	0.753	0.754	0.750
animals	0.818	**0.825**	0.823	0.821	**0.825**	**0.829**	0.821

3) ATTR1 and ATTR2, which model semantic similarity of regions through two mid-level attribute representations (Sec. III-C)
4) GEOVIS and GEOVIS-S, which replace the belief propagation using context, from GRU-based to GCN-based (Sec. III-D)
5) GEOVIS-LING, our custom class name similarity loss (Sec. III-E)
6) SIN, proposed in [9], which uses the expensive GRU framework
7) Faster R-CNN (FRCNN) [17]

We include the number of trainable parameters for a subset of the methods in Tables III, for the PASCAL dataset. Since all methods are identical up to the FC6 layer, and their final classification and regression heads operate on $\mathcal{R}^{4096}$, we also report the number of trainable parameters in between FC6 and R-CNN head for each model in Table IV so that we can make both a fair and dataset-agnostic comparison. Note that our method and Faster R-CNN have comparable number of parameters, while ours achieves better results. In contrast, SIN has 200%-600% more parameters, while only being about 1.5% more accurate (see Table I).

In Table I, we show the performance of the methods highlighting our contributions. We show the top two (or three, in case of ties) best methods for each PASCAL category. We also show the average mAP, as well as the average mAP over animal categories only. We see that our SCENE, ATTR1 and ATTR2 methods, and our GEOVIS-S and GEOVIS-LING outperform FRCNN on most categories and on average. Importantly, **our proposed method, GEOVIS-S, outperforms SIN in terms of the average over animal categories**, and it uses $6\times$ fewer parameters compared to SIN (between FC6 and R-CNN head). This makes our model to be (1) more feasible to deploy on resource-constrained devices, and (2) more suitable for data-parallel distributed training as it will require less bandwidth for communication between servers. Further, all of our methods outperform SIN on many individual categories (5 for SCENE, 7 for ATTR1, 6 for ATTR2, 4 for GEOVIS-S, and 4 for GEOVIS-LING). Each of our methods achieves the best performance on 1-3 categories and up to second-best on 4-8, compared to 1 win for FRCNN. For reference, the results for BASE and GEOVIS are 0.744 and 0.749 (average) and 0.820 and 0.815 (average over the animal classes only). This shows the contribution of the scene and attribute models which improve over BASE, and the contribution of scene and the custom linguistic loss which improve over GEOVIS.

In Table II, we show the results of the two baseline methods, as well as our most promising method (in terms of the animals average), on the COCO dataset. In both cases, we train on COCO 2014. At the top, we show results on 2014 minival, and at the bottom, 2019 test-dev. In the top part, we also show averages over the 11 COCO supercategories [49]. We see that while SIN achieves the best results overall, our method is quite competitive. In particular, it achieves the best result on 4 of the 11 subcategory averages we computed. In terms of the first result, AP@50 (second row of the table), our method achieves 0.408 vs 0.415 for SIN and 0.401 for FRCNN. The gain over FRCNN is thus 3.5% for SIN (at $12\times$ the cost compared to FRCNN) vs 2% for our method (at only $2\times$ the cost). On the other hand, **our method achieves the same performance as SIN** on COCO 2019 test-dev benchmark when required IoU threshold is 0.75.

TABLE II
RESULTS ON THE COCO DATASET. THE BEST METHOD IS UNDERLINED.

COCO 2014 minival			
Test setting / Method	FRCNN	SIN	GeoVis-S
AP @[IoU=0.50:0.95 — area= all]	0.207	0.213	0.209
AP @[IoU=0.50 — area= all]	0.401	0.415	0.408
AP @[IoU=0.75 — area= all]	0.196	0.197	0.193
AP @[IoU=0.50:0.95 — area= small]	0.050	0.055	0.051
AP @[IoU=0.50:0.95 — area=medium]	0.232	0.242	0.237
AP @[IoU=0.50:0.95 — area= large]	0.342	0.346	0.345
Accessories AP @ IoU=[0.50,0.95]	0.079	0.084	0.084
Food AP @ IoU=[0.50,0.95]	0.161	0.166	0.169
Kitchenware AP @ IoU=[0.50,0.95]	0.116	0.118	0.120
Furniture AP @ IoU=[0.50,0.95]	0.216	0.225	0.216
Electronics AP @ IoU=[0.50,0.95]	0.245	0.263	0.255
Appliance AP @ IoU=[0.50,0.95]	0.228	0.252	0.215
Indoor objects AP @ IoU=[0.50,0.95]	0.133	0.137	0.138
Animal AP @ IoU=[0.50,0.95]	0.374	0.371	0.373
Vehicle AP @ IoU=[0.50,0.95]	0.262	0.265	0.262
Sports AP @ IoU=[0.50,0.95]	0.145	0.149	0.146
Outdoor objects AP @ IoU=[0.50,0.95]	0.270	0.271	0.261
COCO 2019 test-dev			
Test setting / Method	FRCNN	SIN	GeoVis-S
AP @[IoU=0.50:0.95 — area= all]	0.207	0.215	0.211
AP @[IoU=0.50 — area= all]	0.403	0.423	0.411
AP @[IoU=0.75 — area= all]	0.194	0.198	0.198

TABLE III
NUMBER OF TRAINABLE PARAMETERS IN THE MOST COMPETITIVE METHODS, ON PASCAL.

Method	FRCNN	SIN	GeoVis-S (Ours)
# Params	136,818,079	321,396,139	153,607,596

TABLE IV
NUMBER OF TRAINABLE PARAMETERS BETWEEN FC6 AND THE R-CNN HEAD.

Method	FRCNN	SIN	GeoVis-S (Ours)
# Params	16,781,312	201,359,372	33,570,829

Fig. 2 shows a qualitative comparison between SIN and our GEOVIS-S at 0.8 confidence threshold. As SIN passes messages between regions based on a single graphical representation wherein edges encode joint spatio-visual relationships between regions, it fails in utilizing context for rare object placements. In the first image it fails to detect the man who rides the bus since VOC07+12 contains very few examples with that particular spatio-visual relation between bus and person. Similarly, in the last image, SIN fails to detect the chair outside as it is under different illumination. Our method detects these two objects perfectly as it utilizes two graphs for message passing, separate for visual and geometric relationships, hence relaxes SIN's constraint.

V. CONCLUSION

We proposed a suite of lightweight techniques for capturing context in object detection. For some of our methods, we only use annotations in terms of object and attribute categories, on a small dataset (PASCAL), without expensive subject-verb-object relations. Most of our methods incur only negligible cost compared to the baseline Structure Inference Net method since we only adjust edge weights. For others of our methods, we switch to a framework which is $6\times$ more efficient than SIN, with results that are comparable or even stronger than SIN. In the future, we will explore the promise of context-based representations for scenarios such as domain adaptation, weakly supervised detection, and video detection.

REFERENCES

[1] S. E. Palmer, "The effects of contextual scenes on the identification of objects," *Memory & Cognition*, vol. 3, pp. 519–526, 1975.

[2] I. Biederman, R. J. Mezzanotte, and J. C. Rabinowitz, "Scene perception: Detecting and judging objects undergoing relational violations," *Cognitive psychology*, vol. 14, no. 2, pp. 143–177, 1982.

[3] D. Navon, "Forest before trees: The precedence of global features in visual perception," *Cognitive Psychology*, 1977.

[4] A. Torralba and P. Sinha, "Statistical context priming for object detection," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, vol. 1, 2001, pp. 763–770.

[5] D. Hoiem, A. A. Efros, and M. Hebert, "Putting objects in perspective," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2006.

[6] M. J. Choi, A. Torralba, and A. S. Willsky, "Context models and out-of-context objects," *Pattern Recognition Letters*, vol. 33, no. 7, pp. 853–862, 2012.

[7] A. Shrivastava and A. Gupta, "Contextual priming and feedback for faster r-cnn," in *European Conference on Computer Vision (ECCV)*. Springer, 2016, pp. 330–348.

[8] X. Chen and A. Gupta, "Spatial memory for context reasoning in object detection," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 4086–4096.

[9] Y. Liu, R. Wang, S. Shan, and X. Chen, "Structure inference net: Object detection using scene-level context and instance-level relationships," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 6985–6994.

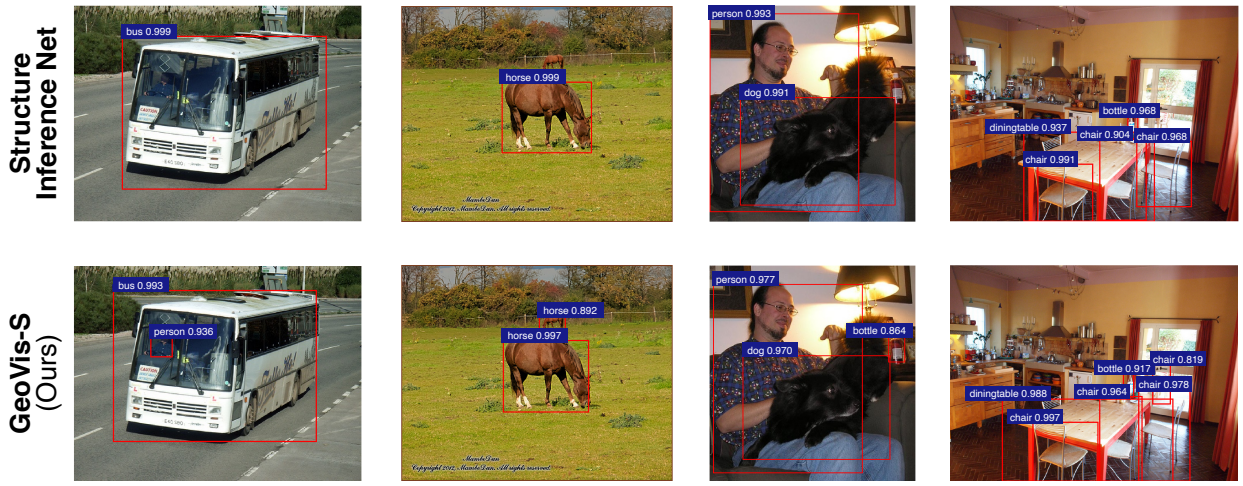

Fig. 2. Example detections from SIN and our method GeoVis-S (best viewed in color).

[10] X. Chen, L.-J. Li, L. Fei-Fei, and A. Gupta, "Iterative visual reasoning beyond convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 7239–7248.

[11] C. Jiang, H. Xu, X. Liang, and L. Lin, "Hybrid knowledge routed modules for large-scale object detection," in *Advances in Neural Information Processing Systems*, 2018, pp. 1552–1563.

[12] K. Kumar Singh, S. Divvala, A. Farhadi, and Y. Jae Lee, "Dock: Detecting objects by transferring common-sense knowledge," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 492–508.

[13] H. Xu, C. Jiang, X. Liang, L. Lin, and Z. Li, "Reasoning-rcnn: Unifying adaptive global reasoning into large-scale object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 6419–6428.

[14] X. Du, X. Shi, and R. Huang, "Repgn: Object detection with relational proposal graph network," *arXiv preprint arXiv:1904.08959*, 2019.

[15] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Oct. 2014, pp. 1724–1734.

[16] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *International Conference on Learning Representations (ICLR)*, 2017.

[17] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[18] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 580–587.

[19] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2117–2125.

[20] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 7132–7141.

[21] Y. Chen, M. Rohrbach, Z. Yan, Y. Shuicheng, J. Feng, and Y. Kalantidis, "Graph-based global reasoning networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 433–442.

[22] Z. Qin, Z. Li, Z. Zhang, Y. Bao, G. Yu, Y. Peng, and J. Sun, "Thundernet: Towards real-time generic object detection on mobile devices," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[23] Z. Chen, S. Huang, and D. Tao, "Context refinement for object detection," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 71–86.

[24] H. Hu, J. Gu, Z. Zhang, J. Dai, and Y. Wei, "Relation networks for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 3588–3597.

[25] K. Marino, R. Salakhutdinov, and A. Gupta, "The more you know: Using knowledge graphs for image classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2673–2681.

[26] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma *et al.*, "Visual genome: Connecting language and vision using crowdsourced dense image annotations," *International Journal of Computer Vision*, vol. 123, no. 1, pp. 32–73, 2017.

[27] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1440–1448.

[28] S. K. Divvala, D. Hoiem, J. H. Hays, A. A. Efros, and M. Hebert, "An empirical study of context in object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2009, pp. 1271–1278.

[29] A. Torralba, K. P. Murphy, W. T. Freeman, M. A. Rubin *et al.*, "Context-based vision system for place and object recognition." in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, vol. 3, 2003, pp. 273–280.

[30] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 891–898.

[31] A. Oliva and A. Torralba, "The role of context in object recognition," *Trends in cognitive sciences*, vol. 11, no. 12, pp. 520–527, 2007.

[32] J. Yang, J. Lu, S. Lee, D. Batra, and D. Parikh, "Graph r-cnn for scene graph generation," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 670–685.

[33] S. Qi, W. Wang, B. Jia, J. Shen, and S.-C. Zhu, "Learning human-object interactions by graph parsing neural networks," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 401–417.

[34] K. Kato, Y. Li, and A. Gupta, "Compositional learning for human object interaction," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 234–251.

[35] W. Yang, X. Wang, A. Farhadi, A. Gupta, and R. Mottaghi, "Visual semantic navigation using scene priors," in *International Conference on Learning Representations (ICLR)*, 2019.

[36] Y. Zhu, A. Fathi, and L. Fei-Fei, "Reasoning about object affordances in a knowledge base representation," in *European Conference on Computer Vision (ECCV)*. Springer, 2014, pp. 408–424.

[37] X. Wang, Y. Ye, and A. Gupta, "Zero-shot recognition via semantic embeddings and knowledge graphs," in *Proceedings of the IEEE Con-*

ference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 6857–6866.

[38] C.-W. Lee, W. Fang, C.-K. Yeh, and Y.-C. Frank Wang, "Multi-label zero-shot learning with structured knowledge graphs," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 1576–1585.

[39] M. Kampffmeyer, Y. Chen, X. Liang, H. Wang, Y. Zhang, and E. P. Xing, "Rethinking knowledge graph propagation for zero-shot learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 11 487–11 496.

[40] Z. Ding, S. Li, M. Shao, and Y. Fu, "Graph adaptive knowledge transfer for unsupervised domain adaptation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 37–52.

[41] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth, "Describing objects by their attributes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2009, pp. 1778–1785.

[42] C. H. Lampert, H. Nickisch, and S. Harmeling, "Attribute-based classification for zero-shot visual object categorization," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 3, pp. 453–465, 2013.

[43] Z. Al-Halah and R. Stiefelhagen, "How to transfer? zero-shot object recognition via hierarchical transfer of semantic attributes," in *2015 IEEE Winter Conference on Applications of Computer Vision*. IEEE, 2015, pp. 837–843.

[44] O. Russakovsky and L. Fei-Fei, "Attribute learning in large-scale datasets," in *European Conference on Computer Vision (ECCV)*. Springer, 2010, pp. 1–14.

[45] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543.

[46] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[47] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Ieee, 2009, pp. 248–255.

[48] 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*, vol. 32, no. 9, pp. 1627–1645, 2009.

[49] 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 *European Conference on Computer Vision (ECCV)*. Springer, 2014, pp. 740–755.