What Object Should I Use? - Task Driven Object Detection

Johann Sawatzky* Yaser Souri* Christian Grund Juergen Gall
University of Bonn
{jsawatzk, ysouri, grund, jgall} @ uni-bonn.de

Abstract

When humans have to solve everyday tasks, they simply pick the objects that are most suitable. While the question which object should one use for a specific task sounds trivial for humans, it is very difficult to answer for robots or other autonomous systems. This issue, however, is not addressed by current benchmarks for object detection that focus on detecting object categories. We therefore introduce the COCO-Tasks dataset which comprises about 40,000 images where the most suitable objects for 14 tasks have been annotated. We furthermore propose an approach that detects the most suitable objects for a given task. The approach builds on a Gated Graph Neural Network to exploit the appearance of each object as well as the global context of all present objects in the scene. In our experiments, we show that the proposed approach outperforms other approaches that are evaluated on the dataset like classification or ranking approaches.

1. Introduction

The task of object detection in images has been widely studied and the community achieved impressive progress on datasets like COCO [24] or Pascal VOC [11]. For many applications like assistive or autonomous systems, however, it is insufficient to detect all instances of a set of object categories. Similar to humans, the systems interact with the environment to solve certain tasks. For instance, if a service robot is asked to serve a glass of wine, detecting all glasses in an image does not answer the question which of them it should use. Taking a beer glass is definitely the wrong choice if a wine glass is available, but if no other glasses are available it might be the best option for the task. Even if there are several wine glasses, not all of them are necessary suitable since some of the glasses might be already used by someone else or need to be cleaned. If no glasses are available, some alternatives have to be considered. For instance, wine can be drunk from a cup or jug as well. This shows that

Figure 1. What object in the scene would a human choose to serve wine? In the left image, the wine glass is preferred to other drinking glasses. In the right image, neither a wine glass nor other drinking glasses are present. The cup is therefore chosen by the human.

answering the question, which object should be used for a task is very difficult since it depends on the present object categories in an image and the properties of the objects.

In this work, we address the problem of *task driven object detection*. It requires to detect all objects in an image which serve a given task best. To this end, we propose the *task driven object detection* (COCO-Tasks) dataset, which is based on the images and annotated objects of the COCO dataset [24]. For evaluation, we define 14 tasks and asked humans to mark all objects in an image which they favor to solve a given task. If none of the objects in an image is suitable, the annotators were allowed to select none of the objects. The dataset comprises about 40,000 annotated images and for each task between 1,100 and 9,900 objects have been marked by the annotators, where the number of different object categories varies between 6 and 30 for the different tasks. Figures 1 and 2 show a few examples.

In our experimental evaluation, we show that *task driven object detection* cannot be treated as a standard object detection task. If a standard object detector is trained for each task using the human annotations as ground-truth, the predictions are not very accurate since the favored objects strongly depend on the presence of other objects and their properties. We therefore propose a method based on Gated Graph Neural Networks (GGNN) [22] that explicitly incorporates all detection hypotheses in an image to infer which objects are preferred for a task. Our experimental results

---

*contributed equally, alphabetically ordered

show that our proposed method outperforms various ranking and classification based baselines and a thorough ablation study analyzes the design choices of our proposed approach. COCO-Tasks dataset and the code for reproducing our experiments are available online[1].

2. Related Work

Due to public benchmarks like Pascal VOC [11] and COCO [24], there was a tremendous advancement in the area of object detection. State-of-the-art object detectors [8, 3, 8, 5, 45, 28, 33, 53] rely exclusively on convolutional neural networks where in particular Faster R-CNN [38] has been widely used. For applications where runtime is critical, other detectors like [36, 25] provide a very good trade-off between efficiency and accuracy.

In contrast to standard object detection, task driven object detection requires an understanding of the entire scene. This relates it to the task of visual question answering which takes as input a question regarding the content of an image and returns an answer in text form, whereas for task driven object detection the input is a task and the output are bounding boxes around objects that are best suitable for solving the task. While [2, 13, 48, 37, 27] pioneered in visual question answering, [42, 43, 1, 32, 52] are examples of current state-of-the-art methods.

Choosing the best object among the available requires not only recognizing its class but judging its functional attributes, *i.e.* its affordances. Detecting and segmenting affordances in images has therefore received an increased interest [30, 31, 18, 39, 9]. In the work [54], learning functional and physical properties together with the handling of objects as tools is investigated. The model is learned from human demonstration and relies on 3d models of objects. The model is then used to recognize tools and affordance regions for 3D objects. Fang et al. [12] propose to learn to detect affordances from demo videos.

Applying deep neural network on graph structured data has seen a lot of attention from the community recently [15, 10, 17, 22]. Many computer vision problems including scene context can naturally be represented as a graph. Wang and Gupta [44] use a Graph Convolutional Network [17] to represent a video and achieve very good results on video classification. Qi et al. [35] have used graph neural networks for semantic segmentation. Chuang et al. [6] used Gated Graph Neural Networks [22] to model affordances in context. While our work compares the objects to each other, [6] focuses on the interaction of objects with their environment.

The task of scene graph generation proposed by Johnson et al. [16] requires the detection of objects and relationships between pairs of them. These relationships are
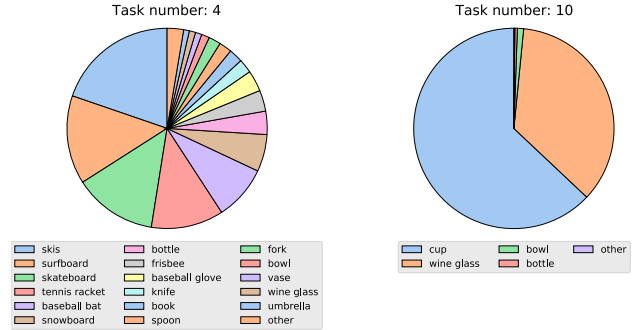
Figure 3. Distribution of chosen objects for task 4 and task 10 across COCO categories. These are the tasks with the highest and lowest number of selected categories, respectively.

typically prepositions indicating relative geometric position and physical interactions. While earlier approaches [26, 55, 34, 51, 50, 46, 19, 23, 7, 21, 29, 49] avoid the search over the exhaustive number of relations by heuristics, more recently [47] propose a method which learns to prune unlikely object relationships. While Li et al. [20] rely on modeling subgraphs for scene graph generation, Zellers et al. [49] focus on correlations between objects and higher order graph structure statistics.

3. COCO-Tasks Dataset

Detecting the objects, which are favored for a given task, is very difficult. It requires localizing objects as for a standard object detection task, but the preferred objects in an image vary among image and task. Figure 2 shows a few examples for the first task (step on something to reach top of a shelf) that requires to move an object to a shelf in order to step on it and take something from the top of the shelf, which cannot be reached otherwise. The first image shows a table which is selected by the annotator since it serves the task. In the second image, however, the table is not selected, since a chair which is much handier is also present. This constitutes the additional difficulty of task driven object detection compared to object detection: the validity of a detection also depends on the presence of better options which need to be detected and assessed. One needs to understand the scene in order to judge a particular object. The third image shows a task specific preference of instances within an object category: The neglected bed on the left hand side looks heavier than the bed on the right hand side. The height of the bed on the right hand side is also sufficient to reach the top of the shelf. In this case, the choice is not anymore at the object category level, but on a finer level where attributes of the instances need to be compared. In summary, task driven object detection requires a detailed understanding of an image, *i.e.*, it needs to be known what objects are in the image and what are the at-

	Task	Selected object categories	Objects of all selected categories	Objects chosen by humans	Intra class differentiations	Annotation consistency
1	step on something	12	30214	5783	964	0.927
2	sit comfortably	12	31392	9870	1004	0.938
3	place flowers	10	14732	3737	734	0.925
4	get potatoes out of fire	30	32775	6889	525	0.921
5	water plant	13	19050	4043	760	0.918
6	get lemon out of tea	15	22386	4707	661	0.873
7	dig hole	29	34015	6857	402	0.922
8	open bottle of beer	12	18177	1105	373	0.921
9	open parcel	7	7172	1759	160	0.921
10	serve wine	6	19209	3778	566	0.963
11	pour sugar	11	20596	5739	944	0.863
12	smear butter	9	17489	1819	270	0.896
13	extinguish fire	8	14821	2535	272	0.940
14	pound carpet	14	34160	7176	432	0.941

Table 1. List of the 14 tasks in the COCO-Tasks dataset and some statistics. Selected object categories (column 3) are COCO object categories for which there exists at least one instance chosen by the majority of the annotators for a given task. Column 4 reports how many instances of each of the selected categories are in the images. Column 5 provides the numbers of object instances that are chosen for each task. Column 6 counts the number of instances of categories in an image where at least one instance but not all instances of the same category are selected. Examples of such cases are shown in the last column of Figure 2. The last column reports the probability that two annotators agree if an object is preferred or not. Overall, we have a very high annotation consistency.

Figure 2. Whether an object should be chosen for a task depends on the object properties as well as the presence of better alternatives. The image in the first row shows the objects (green bounding boxes) that have been chosen by the majority of the annotators for the task `step on something to reach top of a shelf`. While in the first image the table is chosen, the chair is preferred instead of the table in the second image. In the last image, one of the two beds is chosen. The second row shows examples for the task `sit comfortably`. In the first image, both beds are selected. In the second image, the comfortable chair is preferred over the bed and and the stool. In the third image, the real toilet is selected. Note that the reflection of the toilet in the mirror is annotated as object in COCO as object, but it is not selected since one cannot sit on it.

tributes or properties of an instance relative to other objects in an image.

In order to address the problem of task driven object detection, we introduce the COCO-Tasks dataset and we propose a first approach for task driven object detection which will be described in Section 4. The COCO-

Tasks dataset is based on the COCO dataset [24], which is the standard benchmark for object detection. We have defined 14 tasks which are listed in Table 1 together with some statistics. The tasks are quite diverse and include tasks that prefer a specific object shape and material like `serve wine` or `place flowers` and tasks
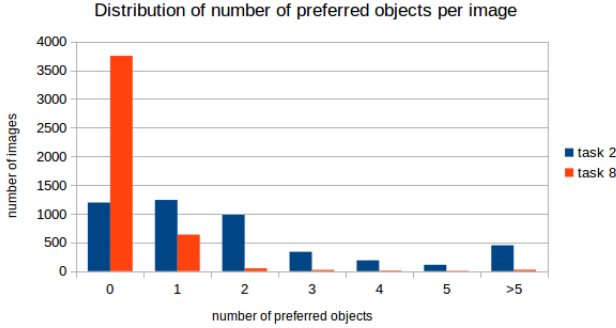
Figure 4. Distribution of the number of preferred objects per image for tasks 2 and 8, which are the tasks with highest and lowest number of selected instances per image, respectively.

that are related but require different attributes of the objects like `step on something to reach top of the shelf` or `sit comfortably`. For each of these tasks, we sample 3600 train images from the COCO train2014 split and 900 test images from the COCO val2014 split. To focus on more complex scenes with multiple objects to choose from, we bias the sampling procedure. For each task, we define which COCO supercategories are most useful. The list of supercategories per task is provided in the supplementary materials. Then we make sure that 40% of the images contain multiple categories from these supercategories, 40% contain exactly one category from these supercategories but multiple instances of it, and 10% of the images contain exactly one instance from one category. The remaining 10% are randomly sampled. In total, our train set contains 30,229 images and our test set contains 9,495 images.

In order to annotate the preferred objects in each of the 4,500 images for each task, we use the available COCO segmentation masks. We highlight the segmentation masks of all objects annotated in the COCO dataset for the annotator. To specify the requirements for the task on a more intuitive level, we visualize all tasks besides of providing a textual description of the task. For instance, we show an image of a shelf for the task `step on something to reach top of the shelf`. The annotators could choose any object, multiple objects or none of them if none of the objects is considered as suitable for this task. The annotators neither knew the procedure of sampling the images nor the supercategories, i.e., they could choose from all 80 COCO categories for each task. Each task was annotated by 5 trained annotators. An object is considered to be preferred if it was chosen by the majority of the annotators. Some example annotations are shown in Figure 2. More information about the annotation tool is provided in the supplementary material.

Table 1 provides some statistics of our dataset. We measured how diverse the selected objects with respect to COCO categories are by counting all categories where at least one instance was selected by the majority of the annotators. The datasets shows a high variation in terms of categories per tasks and the number of selected object categories varies between 6 and 30 depending on the task. From the 80 COCO object class categories of the object detection challenge 2014, instances of 49 classes have been selected for at least one of the 14 tasks. Note that COCO classes also include animals, which are not relevant for the tasks in our dataset. We then measured how many instances of all selected categories for each task are present in our datasets, which also largely varies between 7,172 and 34,160 instances. This shows that just reducing the number of categories to a small set that could be relevant for a task would still leave many instances to choose from. The number of instances that have been selected for each task varies between 1,105 and 9,870. We finally provide the number of instances where the annotators differentiate between instances of the same category as it is shown in the last column of Figure 2. In such cases, the properties or attributes of the instances are relevant to make the decision which object should be used. In Figure 3, we show the distribution of selected object categories for the tasks with the lowest and highest number of selected object categories. While for `serving wine` instances from the categories wine glass and cup are mostly selected, there is a large diversity of categories that have been selected for `getting potatoes out of the fire`. Additionally, we report the distribution of the number of selected instances per image in Figure 4. While for `open bottle of beer` the number of suitable objects is low, there is large diversity in the number of selected instances per image for `sitting comfortable`. The examples show the large variety of category and instance distributions among the tasks. Additional plots are provided in the supplementary material. Furthermore, we evaluated the consistency of the annotations. For each task and each object, we calculated the probability that two annotators agree if this object is preferred or not. As can be seen from Table 1, the consistency between annotators is very high.

As evaluation metric, we use the AP@0.5 object detection evaluation metric of the COCO detection challenge [24] where the preferred objects for a particular task are the ground truth instances to calculate average precision on. Taking the mean over the tasks yields mAP@0.5.

4. Task Driven Object Detection

In order to identify the most suitable objects in an image for a task, it is required to understand what objects are in the scene and why is an object preferred to other present objects. While the objects in an image can be detected by an off-the-shelf object detector, we have to model the relations
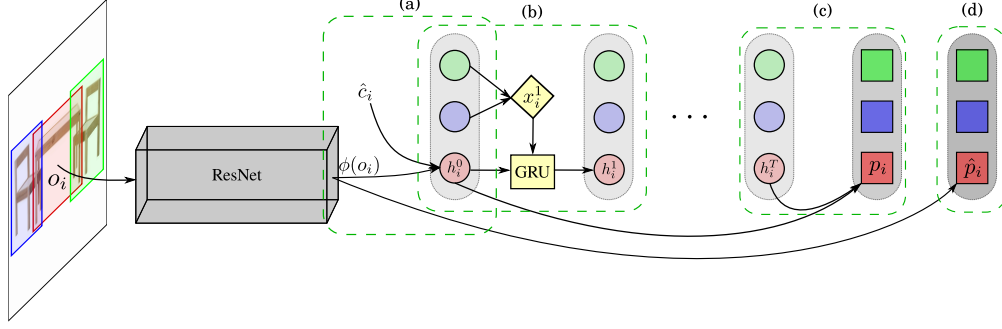
Figure 5. Overview of the proposed method. Given an image containing a number of objects (3 in this example shown with colors green, blue and red), our method first extracts ResNet features from each bounding box containing the object. (a) Using the extracted features and one-hot encoding of the detected class of the object ($\hat{c}_i$), we compute the initial hidden state of the graph node corresponding to that object using (1). (b) Using the hidden states of all of the other graph nodes, we aggregate the scene information using (2) and update the node's hidden state as in (3). (c) After T iterations of the GGNN, we combine each node's initial and final hidden states using (4) to compute the probability of that object being favored for the task. (d) Finally to make the features learned by the ResNet discriminative we also force the network to estimate suitability scores only from visual features of a single object. At test time, we average the two estimated probabilities.

of all present objects in an image to select the preferred objects among all detected objects. To this end, we will use a Gated Graph Neural Network (GGNN) [22] to model the global information of all objects in an image.

4.1. Proposed Method

Our model consists of a ResNet101 [14] network without the final fully connected layer with the weights initialized from ILSVRC. On top of the ResNet features, we construct a Gated Graph Neural Network [22] where each node is an object in the image and each node is connected to all of the other nodes to gather the information from all of the objects present in the scene. On top of the GGNN, we have a fully connected layer which predicts the probability of each object being suitable for each task. We train the whole network end-to-end using binary cross entropy loss. Below we will describe the model in more detail.

An overview of our method is shown in Figure 5. Given an input image I and a collection of N detected objects in that image $o_i, i = 1, ..., N$ specified with their corresponding bounding boxes b_i, detection scores d_i and predicted category c_i, our method predicts p_i the probability of the object o_i being selected for a task.

We first preprocess the bounding boxes by making them square and 10% larger in each dimension, and then crop the image with the preprocessed bounding boxes. We then extract the features from each cropped bounding box arriving at $\phi(o_i)$.

We create a GGNN with one node for each object in the image. We set the initial hidden value of each node based on the one-hot encoding of the category of that object $\hat{c}_i$ and the ResNet features $\phi(o_i)$ such that

$$h_i^0 = g(W_c\hat{c}_i) \odot g(W_\phi\phi(o_i)) \tag{1}$$

where $g(.)$ is the ReLU activation, $\odot$ is the element-wise multiplication and W_c and W_ϕ are parameters of the model.

At each step of the GGNN, we first aggregate the information from all other nodes in the graph:

$$x_i^t = \sum_{j,j\neq i} W_p d_j h_j^{t-1} + b_p \tag{2}$$

where W_p and b_p are the parameters of the learned linear mapping in the aggregation step. This corresponds to a graph where each node is connected to all other nodes. We call the multiplication of d_j in (2) *weighted aggregation*. It gives the possibility to our method to account for misinformation in bad detections with low detection scores. Using the aggregated x_i^t and the previous hidden state of the node h_i^{t-1} we arrive at the new hidden state of each node in the graph using the GRU [4] update rule

$$
\begin{aligned}
z_i^t &= \sigma(W_z x_i^t + U_z h_i^{t-1} + b_z) \\
r_i^t &= \sigma(W_r x_i^t + U_r h_i^{t-1} + b_r) \\
\hat{h}_i^t &= tanh(W_h x_i^t + U_h(r_i^t \odot h_i^{t-1}) + b_h) \\
h_i^t &= (1 - z_i^t) \odot h_i^{t-1} + z_i^t \odot \hat{h}_i^t
\end{aligned}
\tag{3}
$$

where σ is the sigmoid activation and the GRU weights (W_z, W_r, W_h, U_z, U_r, U_h, b_z, b_r, b_h) are learned end-to-end and are shared between all tasks just like the ResNet backbone network. This update rule is applied T times. In our experiments T is set to 3. We observed that increasing T does not improve our results.

At the end of the T iterations the model calculates the probability estimate from the concatenation of the initial and final hidden state of each node

$$p_i = \sigma(f([h_i^0; h_i^T])) \tag{4}$$

while learning the weights. $f(.)$ corresponds to a 2 layer fully connected MLP with ReLU activations for the hidden layer where the final layer has a single output. We can modify this output model to generate one probability value for each task using a final layer with M outputs where M is the number of tasks and train a single model for all tasks jointly. In order to make the features learned by the ResNet discriminative, we also directly compute suitability estimates

$$\hat{p}_i = \sigma(\hat{f}(\phi(o_i))) \qquad (5)$$

from only ResNet features $\phi(o_i)$ as shown in Figure 5 (d). We use two binary cross entropy losses during training for p_i and $\hat{p}_i$. At test time, we use average fusion of p_i and $\hat{p}_i$ to estimate the final probability.

To train our model, we construct each minibatch from objects inside a single image from our training set. All COCO annotated objects are included in the batch, the ones which are specified by our dataset as being preferred for a task are considered as positive examples for that task and the others are considered as negative. Since we use the COCO annotated bounding boxes during training, we set all d_is to 1. During testing, we first perform standard object detection on the test image and get a set of object bounding boxes and their corresponding detection scores and categories. We then perform testing by constructing a batch from all of the detected objects and estimate the probability of each object being preferred for each task as described above. The final confidence for mAP evaluation is obtained by multiplying the detection score with the estimated probability. Implementation details are provided in the supplementary material.

5. Experiments

In this section, we first evaluate the performance of several baselines as well as our proposed method on COCO-Tasks. After that, we demonstrate both qualitatively and quantitatively that our proposed method learns useful information about the scene context. Furthermore with ablation experiments, we show the benefits of each component of our proposed method. For all of our experiments except the object detection baseline we train and test the models three times and report the average performance numbers.

5.1. Comparison to Baselines

For the object detection baseline, we train a separate object detector for each task on our train set and infer on the test set. For all other baselines as well as the proposed method, we train the respective method on ground truth bounding boxes of all COCO objects in the train set. We then evaluate all algorithms on (a) ground truth bounding boxes of COCO objects and (b) COCO object detections of a Faster-RCNN object detector [38]. While the latter

Comparison to Baselines mAP@0.5			
	gt bbox	Faster-RCNN detections	Yolo detections
object detector	-	0.206	-
pick best class	0.386	0.141	-
ranker	0.564	0.091	-
classification	0.616	0.288	0.291
proposed + fusion	**0.742**	**0.326**	**0.332**

Table 2. Comparison of the proposed method to several baselines on ground truth bounding boxes as well as Faster-RCNN [38] detections. The classification baseline is the strongest one but achieves 12.6% lower mAP on ground truth bounding boxes and 3.8% lower mAP on detections compared for our proposed approach.

evaluates the performance in a realistic scenario, the former demonstrates the potential of our method that can be reached with a perfect object detector. As metric, we use mAP@0.5 for all experiments and report the numbers in Table 2.

Object Detector Baseline. The most straightforward approach for task driven object detection is to treat it as a standard object detection task. To this end, for each of the 14 tasks, we train a 1-class object detector. All objects preferred for the respective task constitute the object class to detect. As detector, we use the same Faster-RCNN implementation. Apart from changing the number of classes from 80 to 1, we reduce the learning rate from 0.005 to 0.0001, all other hyperparameters stay identical. As reported in Table 2, this yields an mAP@0.5 of 20.6%, which is more than 10% lower than the proposed approach. This verifies that task driven object detection can not be treated as a standard object detection task because of the necessity to look for scene context and all present objects.

Pick Best Class Baseline. COCO classes differ significantly in their suitability for household tasks. To analyse this effect, we first rank the classes for each task by the fraction of all instances of this class to be preferred on the train set. Then for each task and each image of the test set, we omit all detections with detection confidence lower than 0.1. Among the remaining detections, we determine the highest ranked class and only keep the detections belonging to this class with their detection confidence as final confidence. The result is 14.1% on detections which is significantly worse than the *object detector* baseline. On ground truth bounding boxes, this baseline yields only 38.6% mAP@0.5. This shows that the task driven object detection problem is not solvable by the category information alone, but visual information from the objects and image context are required.

Ranker Baseline. For the *ranker* baseline, we train a model similar to Deep Relative Attributes [41] to rank COCO objects in terms of their suitability for a task. We exchange the original VGG16 backbone [40] of the ranker for a ResNet101 [14] backbone to make the method compa-

Ablation experiment results, mAP@0.5		
	gt bbox	Faster-RCNN detections
classifier	0.616	0.288
(a) joint classifier	0.647	0.302
(b) joint classifier + class	0.719	0.301
(c) joint GGNN + class	0.763	0.293
(d) joint GGNN + class + w. aggreg.	-	0.303
(e) proposed	**0.771**	0.318
(f) proposed + fusion	0.742	**0.326**
(g) no visual input	0.589	0.237
(h) no visual input + bounding box	0.412	0.152

Table 3. Evaluation of the components of our proposed method. We start with a task wise classifier, (a) then add joint training, (b) add COCO classes as input, (c) introduce the GGNN, (d) add weighted aggregation, (e) add the discriminatory loss and (f) perform fusion. Further ablation experiments (g) and (h) reveal the impact of the visual information.

rable to other baselines. We train a model for each task separately using the Adam optimizer with 10^{-4} learning rate for 3 epochs to remain as close as possible to [41]. As for the *pick best class* baseline, we prefilter the detections by a detection confidence threshold of 0.1. Then for each image and each task, we rank all n detections and assign each detection i of rank r_i the confidence $c_i = 1 - \frac{r_i - 1}{n}$. Although on ground truth bounding boxes this method performs better than *pick best class*, it is the worst baseline on detections giving only 9.1% mAP@0.5 as can be seen from Table 2. The reason is that a single detection ranked erroneously highly affects all other detections.

Classification Baseline. To investigate if a global analysis of all objects present in the scene is necessary, we train a binary classifier on top of the ResNet features for each task and apply it on detections and ground truth bounding boxes. As for the proposed method, we obtained the final confidence by multiplying the classifier output and the detector confidence. This baseline model is equivalent to our method, without the class information input, the context modeling using a graph and joint training for all tasks simultaneously. This is the strongest baseline as can be seen from Table 2. It gives 61.6% on ground truth bounding boxes and 28.8% on detections. However, this is still substantially below our proposed method which takes the scene context into account.

Proposed Method. The proposed method with fusion where we average p_i and $\hat{p}_i$ for final estimate, yields 32.6% on Faster-RCNN [38] detections and 74.2% on ground truth bounding boxes outperforming our baselines by a large margin. Various ablation experiments showing the effect of different components of our method will follow.

Other Detector Our method outperforms the strongest classifier baseline even if we use the Yolov2 detector [36] as can be seen from Table 2.

5.2. Ablation Experiments

We observed that the classification baseline was lacking in performance compared to our proposed method. This is due to the differences between the classification baseline and our proposed method. These differences are: (a) joint training of all tasks together, (b) direct class information input, and (c) GGNN for scene context modeling. We will add these 3 components one by one to the classification baseline and show the effect of each of them. Furthermore, in our GGNN we show the effect of (d) *weighted aggregation*, (e) the direct discriminatory loss on top of the ResNet features and (f) fusion of p_i and $\hat{p}_i$ for the final probability estimate. The results for these ablation experiments are reported in Table 3.

a) Joint Training. While for the classification baseline we train a separate classifier for each of the tasks, a first improvement can be easily obtained by training a classifier jointly for all tasks, *i.e.* using shared features. This is done by replacing the final single output fully connected layer that estimates p_i into a layer with M outputs, where M is the number of tasks. If a task is annotated for an image during training, we calculate the binary cross entropy loss and skip that task otherwise. Training the classifier jointly increases the performance on ground truth bounding boxes from 61.6% to 64.7% and on detections from 28.8% to 30.2%. We think this is due to the higher number of training images and better features that are learned by ResNet.

b) Direct Class Information Input. The object's class as a direct input provides additional valuable information that might be harder for the network to learn from ResNet features. Given this insight we use (1) to combine the ResNet features ($\phi(o_i)$) and the one-hot encoding of the classes ($\hat{c}_i$) as it is done in our proposed method. We then use the hidden representations h_i^0 as input to the final classification layer. During training we use the ground truth class, during inference we use the detected classes which might be noisy. On ground truth, the results get boosted from 64.7% to 71.9%. However, on detections, the performance stays almost the same. We reckon that this is due to the difference between reliable ground truth classes during training and erroneous classes as predicted by the detector during inference. In our proposed method, this problem is addressed by our *weighted aggregation* mechanism.

c) GGNN for Scene Context Modeling. We now add the GGNNs as described in Section 4.1 to see the effect of scene context modeling. For this ablation experiment, the *weighted aggregation* (by setting all d_is in (2) to 1) and the discriminator loss are not used. This is equivalent to a simplified GGNN. On ground truth bounding boxes we get an improvement of 4.4% arriving at 76.3% as a result of scene context modeling, but on detections the performance slightly drops to 29.3%. The detection confidence problems encountered by the classifier are amplified, since the GGNN

takes all detections into account when judging a single one. Thus the final result for each detection is affected by low confidence detections during inference. Typically these are wrong detections, thus the GGNN is confronted with visual input not seen during training. To solve this issue, we have incorporated the *weighted aggregation*.

d) Weighted Aggregation. By the *weighted aggregation*, we take the confidences of the detections d_is into account (2). We observe that addition of such weighting improves our results considerably on detections. This thwarts the propagation of visual features of low confidence detections through the GGNN resulting in an improvement from 29.3% to 30.3%. Note that the *weighted aggregation* does not change the result on ground truth bounding boxes since the d_is are equal to 1 in this case.

e) Direct Discriminator Loss. We also impose intermediate supervision on the visual features fed into the initializer. We add a fully connected layer mapping these features onto probabilities for each task and apply a task wise binary cross entropy loss to these probabilities. This loss makes the visual features more discriminative for the final goal. The features give a better backup in case the class information is not correct. In general such a loss improves the performance of our model to 77.1% and 31.8% on ground truth bounding boxes and detections, respectively.

f) Probability Fusion. Average fusion of the probabilities p_i from (4) and $\hat{p}_i$ from (5) further improves the results on detections. We observe that this does cause some performance decrease for the case of a perfect detector. The fusion is therefore only relevant if the detections are noisy.

g) Removing Visual Input $\phi(o_i)$. Since class information improves the results on ground truth bounding boxes significantly, the question comes to mind if visual information inside the bounding boxes is necessary at all. To test this, we do not use the visual features $\phi(o_i)$ for GGNN and only keep the class information as input. As a result, the mAP significantly drops, showing that the appearance of the objects is very important for the task and that GGNN takes it into account.

h) Removing Visual Input $\phi(o_i)$ and Adding Bounding Box Geometry. We then used the coordinates of the bounding boxes normalized by image width and height $b(o_i)$ instead of the visual features $\phi(o_i)$ for GGNN (proposed no vis. input + bbox). This leads to even worse results since the model overfits to the coordinates of the bounding boxes of the objects inside the training images.

In the supplementary material we provide the results for each task.

5.3. Scene Context Learned by GGNN

The aim of introducing the GGNN was to consider scene context in our model. Intuitively, the GGNN aggregates the information about all objects relevant for the task which are
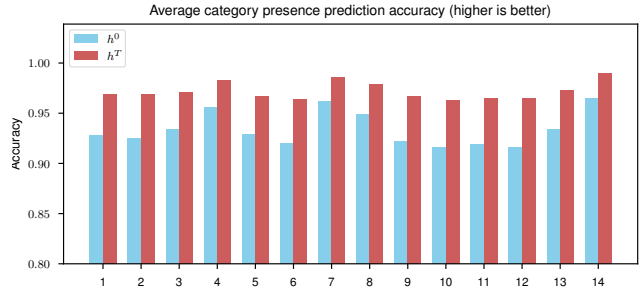

Figure 6. Accuracy of the predicted categories when retrieving nearest neighbors based on h_i^0 vs. retrieving based on h_i^T.

present in the image and stores them in the final hidden node representation h_i^T. To prove this intuition quantitatively, we retrieve the 5 most similar objects for each task and each object of the test set. Then we use the categories present in the scene of the query object as a prediction for the categories present in the scene of the retrieved objects and measure the prediction accuracy. We compute the similarity based on h_i^T, which should contain scene information and compare it to the similarity computed based on h_i^0.

The prediction accuracies are high in both cases, which is primarily due to the fact that most COCO categories are absent in any image. However as can be seen from Figure 6, when retrieving based on similarity of h_i^0 instead of h_i^T, the accuracy of this prediction is significantly lower for all tasks. This verifies our intuition. More analysis and qualitative examples are provided in the supplementary material.

6. Conclusion

In this work, we have addressed the problem of task driven object detection. In contrast to standard object detection, it requires to detect and select the best objects for solving a given task. To study this problem, we created a dataset based on the COCO dataset [24]. It comprises about 40k images with annotations for 14 tasks. We evaluated several baselines based on ranking or classification approaches on this dataset. We furthermore introduced a novel approach for this task that takes as input all detected objects in an image and uses a Gated Graph Neural Network to model the relations of the object hypotheses in order to infer the objects that are preferred for a given task.

References

[1] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. *CVPR*, 2018. 2

[2] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. VQA: Visual question answering. *ICCV*, 2015. 2

[3] Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S Davis. Soft-NMS – Improving object detection with one line of code. *ICCV*, 2017. 2

[4] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *SSST*, 2014. 5

[5] François Chollet. Xception: Deep learning with depthwise separable convolutions. *CVPR*, 2017. 2

[6] Ching-Yao Chuang, Jiaman Li, Antonio Torralba, and Sanja Fidler. Learning to act properly: Predicting and explaining affordances from images. *CVPR*, 2018. 2

[7] Bo Dai, Yuqi Zhang, and Dahua Lin. Detecting visual relationships with deep relational networks. *CVPR*, 2017. 2

[8] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. *ICCV*, 2017. 2

[9] Thanh-Toan Do, Anh Nguyen, Ian Reid, Darwin G. Caldwell, and Nikos G Tsagarakis. Affordancenet: An end-to-end deep learning approach for object affordance detection. *ICRA*, 2018. 2

[10] David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alan Aspuru-Guzik, and Ryan P Adams. Convolutional networks on graphs for learning molecular fingerprints. In *NIPS*. 2015. 2

[11] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher K.I. Williams, John Winn, and Andrew Zisserman. The PASCAL Visual Object Classes Challenge 2012: A Retrospective. *IJCV*, 2014. 1, 2

[12] Kuan Fang, Te-Lin Wu, Daniel Yang, Silvio Savarese, and Joseph J. Lim. Demo2Vec: Reasoning object affordances from online videos. In *CVPR*, 2018. 2

[13] Haoyuan Gao, Junhua Mao, Jie Zhou, Zhiheng Huang, Lei Wang, and Wei Xu. Are you talking to a machine? Dataset and methods for multilingual image question. *NIPS*, 2015. 2

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CVPR*, 2016. 5, 6

[15] Mikael Henaff, Joan Bruna, and Yann LeCun. Deep convolutional networks on graph-structured data. *arXiv preprint arXiv:1506.05163*, 2015. 2

[16] Justin Johnson, Ranjay Krishna, Michael Stark, Li-Jia Li, David Shamma, Michael Bernstein, and Li Fei-Fei. Image retrieval using scene graphs. *CVPR*, 2015. 2

[17] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *ICLR*, 2017. 2

[18] Sulabh Kumra and Christopher Kanan. Robotic grasp detection using deep convolutional neural networks. *IROS*, 2017. 2

[19] Yikang Li, Wanli Ouyang, and Xiaogang Wang. ViP-CNN: A visual phrase reasoning convolutional neural network for visual relationship detection. *CVPR*, 2017. 2

[20] Yikang Li, Wanli Ouyang, Bolei Zhou, Jianping Shi, Chao Zhang, and Xiaogang Wang. Factorizable net: An efficient subgraph-based framework for scene graph generation. *ECCV*, 2018. 2

[21] Yikang Li, Wanli Ouyang, Bolei Zhou, Kun Wang, and Xiaogang Wang. Scene graph generation from objects, phrases and region captions. *ICCV*, 2017. 2

[22] Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. Gated graph sequence neural networks. *ICLR*, 2016. 1, 2, 5

[23] Xiaodan Liang, Lisa Lee, and Eric P Xing. Deep variation-structured reinforcement learning for visual relationship and attribute detection. *CVPR*, 2017. 2

[24] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*. 2014. 1, 2, 3, 4, 8

[25] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. SSD: Single shot multibox detector. *ECCV*, 2016. 2

[26] Cewu Lu, Ranjay Krishna, Michael Bernstein, and Li Fei-Fei. Visual relationship detection with language priors. *ECCV*, 2016. 2

[27] Mateusz Malinowski and Mario Fritz. A multiworld approach to question answering about realworld scenes based on uncertain input. *NIPS*, 2014. 2

[28] Jiayuan Mao, Tete Xiao, Yuning Jiang, and Zhimin Cao. What can help pedestrian detection? *CVPR*, 2017. 2

[29] Alejandro Newell and Jia Deng. Pixels to graphs by associative embedding. *NIPS*, 2017. 2

[30] Anh Nguyen, Dimitrios Kanoulas, Darwin G. Caldwell, and Nikos G. Tsagarakis. Detecting object affordances with convolutional neural networks. *IROS*, 2016. 2

[31] Anh Nguyen, Dimitrios Kanoulas, Darwin G Caldwell, and Nikos G Tsagarakis. Object-based affordances detection with convolutional neural networks and dense conditional random fields. *IROS*, 2017. 2

[32] Duy-Kien Nguyen and Takayuki Okatani. Improved fusion of visual and language representations by dense symmetric co-attention for visual question answering. In *CVPR*, 2018. 2

[33] Chao Peng, Xuangyu Zhang, Gang Yu, Guiming Luo, and Jian Sun. Large kernel matters–Improve semantic segmentation by global convolutional network. *CVPR*, 2017. 2

[34] Julia Peyre, Ivan Laptev, Cordelia Schmid, and Josef Sivic. Weakly-supervised learning of visual relations. *ICCV*, 2017. 2

[35] Xiaojuan Qi, Renjie Liao, Jiaya Jia, Sanja Fidler, and Raquel Urtasun. 3D graph neural networks for RGBD semantic segmentation. In *CVPR*, 2017. 2

[36] Joseph Redmon and Ali Farhadi. YOLO 9000: Better, faster, stronger. *CVPR*, 2017. 2, 7

[37] Mengye Ren, Ryan Kiros, and Richard Zemel. Exploring models and data for image question answering. *NIPS*, 2015. 2

[38] Shaoqing Ren, Kaiming He, Ross Girshik, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. *ICCV*, 2015. 2, 6, 7

[39] Johann Sawatzky, Abhilash Srikantha, and Juergen Gall. Weakly supervised affordance detection. *CVPR*, 2017. 2

[40] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 6

[41] Yaser Souri, Erfan Nouri, and Ehsan Adeli. Deep relative attributes. *ACCV*, 2016. 6, 7

[42] Damien Teney, Peter Anderson, Xiaodong He, and Anton van den Hengel. Tips and tricks for visual question answering: Learnings from the 2017 challenge. *ECCV*, 2018. 2

[43] Damien Teney and Anton van den Hengel. Visual question answering as a meta learning task. *ECCV*, 2018. 2

[44] Xiaolong Wang and Abhinav Gupta. Videos as space-time region graphs. In *ECCV*, 2018. 2

[45] Saining Xie, Ross Girshick, Piotr Dollr, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. *CVPR*, 2017. 2

[46] Danfei Xu, Yuke Zhu, Christopher B. Choy, and Li Fei-Fei. Scene graph generation by iterative message passing. *CVPR*, 2017. 2

[47] Jianwei Yang, Jiasen Lu, Stefan Lee, Dhruv Batra, and Devi Parikh. Graph R-CNN for scene graph generation. *ECCV*, 2018. 2

[48] Licheng Yu, Eunbyung Park, Alexander C. Berg, and Tamara L. Berg. Visual madlibs: Fill in the blank description generation and question answering. *ICCV*, 2015. 2

[49] Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. Neural motifs: Scene graph parsing with global context. *CVPR*, 2018. 2

[50] Hanwang Zhang, Zawlin Kyaw, Shih-Fu Chang, and Tat-Seng Chua. Visual translation embedding network for visual relation detection. *CVPR*, 2017. 2

[51] Hanwang Zhang, Zawlin Kyaw, Jinyang Yu, and Shih-Fu Chang. PPR-FCN: Weakly supervised visual relation detection via parallel pairwise R-FCN. *ICCV*, 2017. 2

[52] Yan Zhang, Jonathon Hare, and Adam Prgel-Bennett. Learning to count objects in natural images for visual question answering. *ICLR*, 2018. 2

[53] Peng Zhou, Bingbing Ni, Cong Geng, Jianguo Hu, and Yi Xu. Scale-transferrable object detection. In *CVPR*, 2018. 2

[54] Yixin Zhu, Yibiao Zhao, and Song Chun Zhu. Understanding tools: Task-oriented object modeling, learning and recognition. In *CVPR*, 2015. 2

[55] Bohan Zhuang, Lingqiao Liu, Chunhua Shen, and Ian Reid. Towards context-aware interaction recognition for visual relationship detection. *ICCV*, 2017. 2