# Supplementary Material: 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

| Task # | Task | COCO supercategories |
|---|---|---|
| 1 | step on something | furniture |
| 2 | sit comfortably | furniture |
| 3 | place flowers | kitchen, outdoor |
| 4 | get potatoes out of fire | sports, kitchen, outdoor |
| 5 | water plant | kitchen, indoor |
| 6 | get lemon out of tea | kitchen |
| 7 | dig hole | sports, kitchen, indoor |
| 8 | open bottle of beer | furniture, kitchen, indoor |
| 9 | open parcel | kitchen, indoor |
| 10 | serve wine | kitchen |
| 11 | pour sugar | kitchen |
| 12 | smear butter | kitchen |
| 13 | extinguish fire | kitchen, indoor |
| 14 | pound carpet | sports |

Table 1. We ensured that for each task images with certain COCO supercategories are overrepresented in the dataset. This table shows the respective supercategories.

## 1. COCO-Tasks Dataset

As described in Section 3, we use the COCO supercategories to sample images. The list of supercategories used for each task is provided in Table 1.

### 1.1. Annotation Tool

A screen shot of the annotation tool is provided in Figure 1.

### 1.2. Dataset Statistics

The distributions of the chosen object categories for all tasks are provided in Figure 2. We can see that the COCO-Tasks dataset provides a wide range of tasks with respect to the distribution of chosen object categories. For example, Task 1 (*step on something*) shows a large bias towards the *chair* category. Humans, however, differentiate between instances of the same category for each task and the baseline *pick best class*, which exploits the category bias for each task, achieves only 22.9% mAP@0.5 for Task 1 (Table 2). In contrast, our approach learns to differentiate between categories as well as instances of the same category and achieves 36.6%. For ground-truth bounding boxes, the difference is even larger (47.3% vs. 81%). The distributions of the number of selected instances per image for all tasks are provided in Figure 3.

## 2. Proposed Method and Experiments

### 2.1. Implementation Details

We train our models using standard SGD for 3 epochs. We use an initial learning rate of $10^{-2}$ and decay it by a factor of 10 every epoch. During training, we shuffle the order of images before each epoch starts. After calculating the gradients, we clip them if the $\ell_2$ norm of the gradients are larger than 15. For regularization, we use Dropout with 0.25 probability. We place dropout layers before final layers in all our methods. We also apply dropout to the $x_i^t$ input to the GRU at each step. The input and hidden size of the GRU is 128. The source code to reproduce our results is available online[1].

We weight by 10 and 1 the cross entropy losses on top of $p_i$ and $\hat{p}_i$ respectively. To compute the ranking for mAP calculations, we first compute the final probability estimate of each object and then multiply it with the detection score.

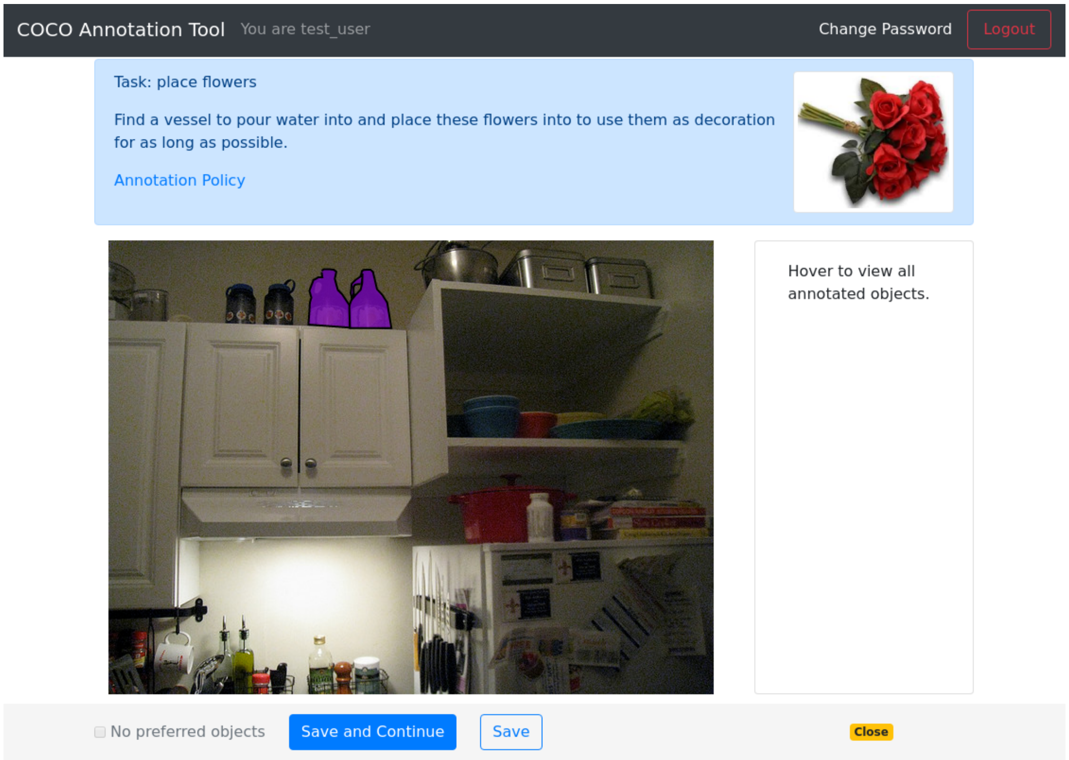
Our vanilla object detector is a derivative of Faster-RCNN [3] provided by [4]. As backbone model we take ResNet101 [1] and the weights are updated with SGD. We use 120 region proposals, a batch size of 2, start learning rate of 0.005 and train for 10 epochs. The learning rate decays by 0.1 every 5 epochs. We use the large scale option to upscale the shorter side of the image to 800 pixels. We train on all images of the COCO train2014 split and all images of val2014 split which do not belong to our test set. The detector achieves an object detection mAP of 25.0% and a mAP@0.5 of 39.4% on our test set.

### 2.2. Task Wise Comparison to Baselines

In this section we provide per task comparison of our proposed methods result to baselines provided in the paper. The results are provided in Table 2.

*contributed equally, alphabetically ordered

[1] coco-tasks.github.io


COCO Annotation Tool You are test_user

Task: place flowers

Find a vessel to pour water into and place these flowers into to use them as decoration for as long as possible.

Hover to view all annotated objects.

□ No preferred objects

Figure 1. Overview of the annotation tool used by the annotators. At the top (blue section) the user can see the description of the task with an image which clarifies the intention of the task. Below is the image with already selected preferred objects (as purple). The annotator should select any object that is suitable by clicking on it. If the annotator hovers on an COCO annotated instance in the image, it will become highlighted with yellow color. No COCO class information is provided to the annotator. The user is able to click on an already selected instance to deselect it. On the right hand side of the image is an area over that the annotator can hover with the mouse to view all COCO annotated instances in the image (all with the same color). If there are no preferred instances for the task in the image, the user is able to mark a check box at the bottom and save the annotation result for the image.

| Comparison to Baselines on Faster-RCNN detections, mAP@0.5 | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| task # | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | **avg.** |
| object detector | 0.281 | 0.258 | 0.301 | 0.220 | 0.305 | 0.117 | 0.308 | 0.00 | 0.051 | 0.334 | 0.097 | 0.061 | 0.246 | 0.309 | 0.206 |
| pick best class | 0.229 | 0.181 | 0.198 | 0.150 | 0.213 | 0.058 | 0.204 | 0.039 | 0.033 | 0.220 | 0.111 | 0.05 | 0.125 | 0.156 | 0.141 |
| *detection stats* | 0.246 | 0.195 | 0.196 | 0.142 | 0.150 | 0.066 | 0.162 | 0.040 | 0.077 | 0.218 | 0.112 | 0.132 | 0.093 | 0.142 | 0.141 |
| ranker | 0.107 | 0.104 | 0.115 | 0.116 | 0.118 | 0.033 | 0.150 | 0.024 | 0.046 | 0.105 | 0.052 | 0.050 | 0.083 | 0.172 | 0.091 |
| classification | 0.331 | 0.267 | 0.368 | 0.329 | 0.354 | 0.146 | 0.403 | 0.144 | 0.176 | 0.384 | 0.171 | 0.245 | 0.332 | 0.381 | 0.288 |
| proposed + fusion | **0.366** | **0.298** | **0.405** | **0.376** | **0.410** | **0.172** | **0.436** | **0.179** | **0.210** | **0.406** | **0.223** | **0.284** | **0.391** | **0.407** | **0.326** |

| Comparison to Baselines on ground truth bounding boxes, mAP@0.5 | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| task # | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | **avg.** |
| pick best class | 0.473 | 0.637 | 0.411 | 0.524 | 0.392 | 0.338 | 0.517 | 0.098 | 0.343 | 0.340 | 0.453 | 0.151 | 0.138 | 0.585 | 0.386 |
| *detection stats* | 0.691 | 0.850 | 0.638 | 0.860 | 0.594 | 0.660 | 0.906 | 0.243 | 0.679 | 0.743 | 0.608 | 0.575 | 0.498 | 0.906 | 0.675 |
| ranker | 0.502 | 0.687 | 0.554 | 0.706 | 0.604 | 0.334 | 0.784 | 0.215 | 0.629 | 0.473 | 0.404 | 0.617 | 0.581 | 0.812 | 0.564 |
| classification | 0.676 | 0.762 | 0.610 | 0.800 | 0.549 | 0.497 | 0.871 | 0.265 | 0.458 | 0.728 | 0.435 | 0.562 | 0.539 | 0.870 | 0.616 |
| proposed + fusion | **0.810** | **0.847** | **0.702** | **0.914** | **0.668** | **0.640** | **0.951** | **0.385** | **0.727** | **0.790** | **0.590** | **0.747** | **0.672** | **0.945** | **0.742** |

Table 2. Comparison of the proposed method to several baselines on ground truth bounding boxes as well as Faster-RCNN [3] detections. This is an extension of Table 2 in the paper. *Detection stats* baseline is not included in the paper

We also report results on an additional baseline.

**Detection Stats Baseline.** We want to test if visual input is necessary or if the statistics of the present objects alone is sufficient to solve the task driven object detection problem. To this end, we trained a smaller 2-layer MLP to predict from the normalized histogram of present object classes as well as the average normalized bounding box coordinates for each class the preferred object class. As for the proposed method, we obtained the final confidence by multiplying the MLP output and the detector confidence. As for the *pick best class* baseline, we prefilter the detections by a detection confidence of 0.1. While this baseline performs well for ground-truth bounding boxes, it is still worse than the proposed approach. For detected bounding boxes, this

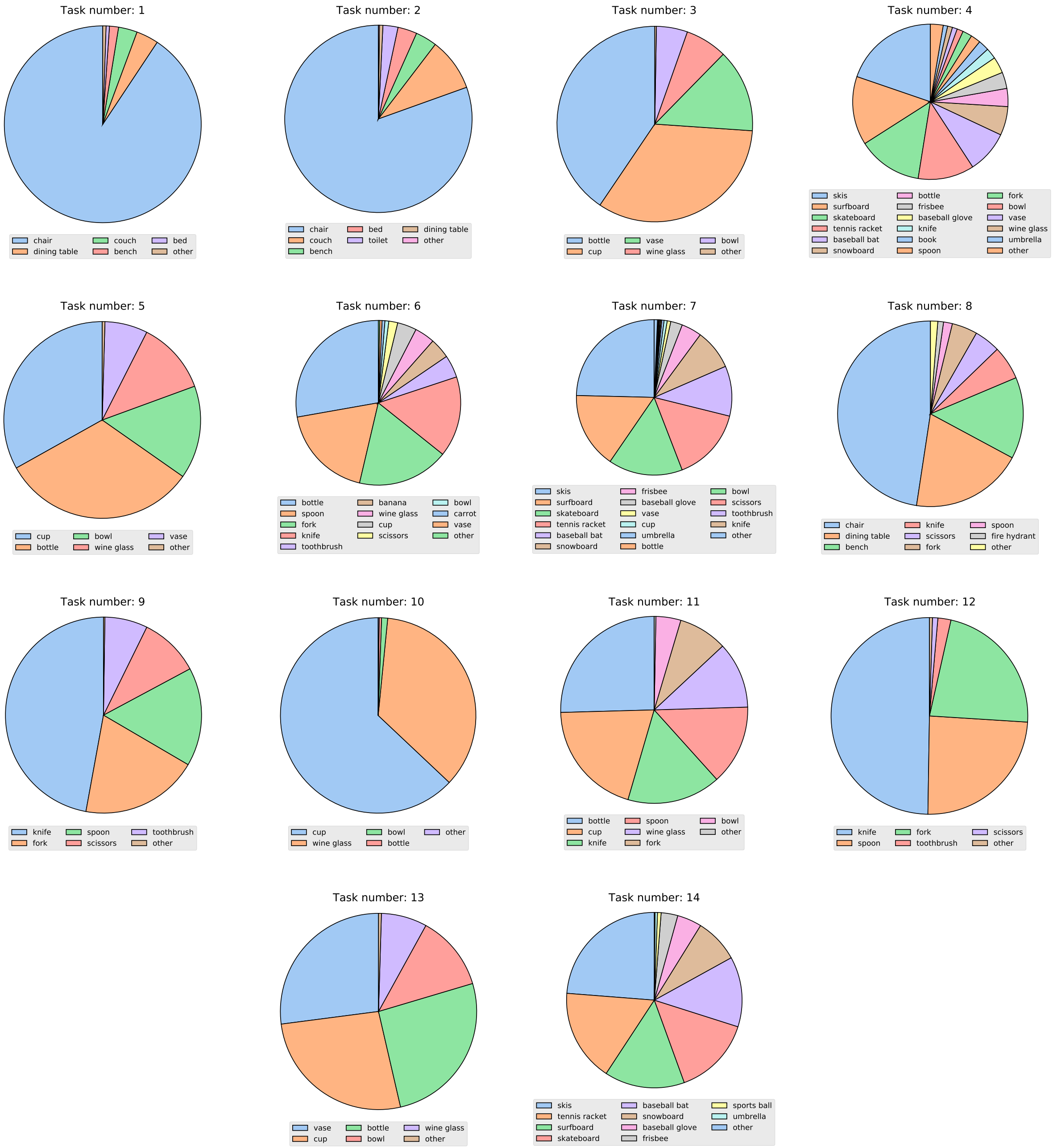


Figure 2. Distribution of chosen objects for all tasks across COCO categories.

baseline performs poorly.

### 2.3. Task Wise Ablation Experiment Results

Per task ablation experiment results are provided in Table 3.

### 2.4. Class Bias Analysis

In Table 4, we report the results only for the subset of the test images where an instance of the most suitable class is present, but an instance of another suitable class has been selected by the users. We conducted the experiment for our proposed method as well as the classification baseline which is the strongest baseline on Faster-RCNN detections and the detection stats baseline which is the strongest on

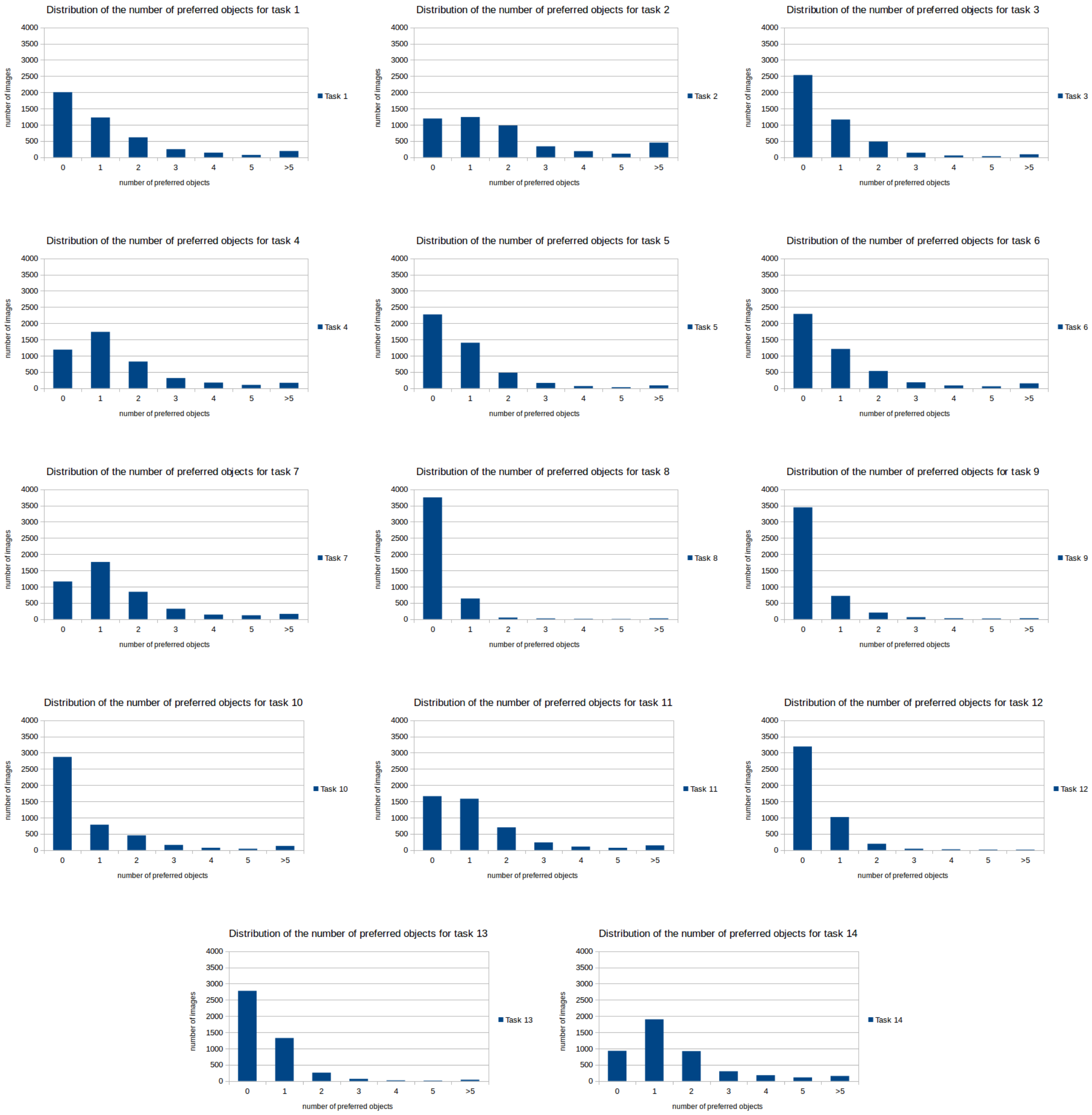


Figure 3. Distribution of the number of selected instances per image for all tasks.

ground truth bounding boxes. As expected, these cases are much more difficult and mAP@0.5 is much lower compared to Table 2. Nevertheless, our approach outperforms the baselines. This shows that the proposed model takes the appearance of the object and the scene context into account and is able to make decisions against the class bias, but there is a large room for improvement even for ground truth bounding boxes.

## 2.5. Task Wise Difficulty Analysis

In Table 5 we report the mAP@0.5 for each task of the proposed method on ground truth bounding boxes and the proposed+fusion method for Faster-RCNN [3] detections and Yolov2 [2] detections. As expected, our method generalizes well to another detector. The performance on the individual tasks ranges from 37.0% to 96.1% on ground truth bounding boxes. The different difficulty of the tasks can be explained by the typical class of the preferred ob-

| Ablation experiment results on Faster-RCNN detections, mAP@0.5 | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| task # | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | **avg.** |
| classifier | 0.331 | 0.267 | 0.368 | 0.329 | 0.354 | 0.146 | 0.403 | 0.144 | 0.176 | 0.384 | 0.171 | 0.245 | 0.332 | 0.381 | 0.288 |
| (a) j. clfr. | 0.343 | 0.274 | 0.370 | 0.347 | 0.379 | 0.158 | 0.418 | 0.170 | 0.196 | 0.386 | 0.182 | 0.254 | 0.363 | 0.388 | 0.302 |
| (b) j. clfr. + cls | 0.343 | 0.287 | 0.361 | 0.381 | 0.358 | 0.165 | 0.436 | 0.109 | 0.178 | 0.382 | 0.180 | 0.250 | 0.368 | 0.410 | 0.301 |
| (c) j. GGNN + cls | 0.349 | 0.283 | 0.342 | 0.372 | 0.361 | 0.158 | 0.434 | 0.125 | 0.162 | 0.383 | 0.166 | 0.231 | 0.333 | 0.409 | 0.293 |
| (d) j. GGNN + cls + w.a. | 0.350 | 0.287 | 0.362 | 0.382 | 0.370 | 0.163 | 0.437 | 0.132 | 0.169 | 0.387 | 0.198 | 0.249 | 0.345 | **0.413** | 0.303 |
| (e) proposed | **0.367** | **0.298** | **0.405** | **0.383** | 0.398 | 0.165 | **0.438** | 0.136 | 0.187 | 0.405 | 0.214 | 0.268 | 0.372 | 0.411 | 0.318 |
| (f) proposed + fusion | 0.366 | **0.298** | **0.405** | 0.376 | **0.410** | **0.172** | 0.436 | **0.179** | **0.210** | **0.406** | **0.223** | **0.284** | **0.391** | 0.407 | **0.326** |
| (g) no vis. input | 0.320 | 0.234 | 0.282 | 0.358 | 0.286 | 0.082 | 0.419 | 0.043 | 0.060 | 0.342 | 0.159 | 0.116 | 0.278 | 0.394 | 0.241 |
| (h) no vis. input + bbox | 0.164 | 0.140 | 0.211 | 0.300 | 0.194 | 0.043 | 0.354 | 0.033 | 0.029 | 0.187 | 0.106 | 0.035 | 0.135 | 0.327 | 0.161 |

| Ablation experiment results on ground truth bounding boxes, mAP@0.5 | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| task # | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | **avg.** |
| classifier | 0.676 | 0.762 | 0.610 | 0.800 | 0.549 | 0.497 | 0.871 | 0.265 | 0.458 | 0.728 | 0.435 | 0.562 | 0.539 | 0.870 | 0.616 |
| (a) j. clfr. | 0.713 | 0.772 | 0.624 | 0.833 | 0.577 | 0.543 | 0.895 | 0.343 | 0.512 | 0.734 | 0.470 | 0.579 | 0.572 | 0.892 | 0.647 |
| (b) j. clfr. + cls | 0.755 | 0.822 | 0.663 | 0.918 | 0.569 | 0.664 | 0.959 | 0.328 | 0.774 | 0.809 | 0.537 | 0.736 | 0.578 | 0.953 | 0.719 |
| (c) j. GGNN + cls | 0.777 | 0.838 | 0.719 | **0.935** | 0.697 | **0.702** | 0.957 | 0.375 | **0.804** | 0.801 | **0.650** | 0.797 | 0.675 | **0.963** | 0.763 |
| (e) proposed | **0.831** | **0.853** | **0.732** | 0.934 | **0.699** | 0.690 | **0.961** | 0.370 | **0.804** | **0.810** | 0.638 | **0.803** | **0.706** | 0.960 | **0.771** |
| (f) proposed + fusion | 0.810 | 0.847 | 0.702 | 0.914 | 0.668 | 0.640 | 0.952 | **0.385** | 0.727 | 0.790 | 0.590 | 0.747 | 0.672 | 0.945 | 0.742 |
| (g) no vis. input | 0.645 | 0.804 | 0.478 | 0.882 | 0.443 | 0.456 | 0.913 | 0.148 | 0.473 | 0.586 | 0.527 | 0.493 | 0.479 | 0.921 | 0.589 |
| (h) no vis. input + bbox | 0.372 | 0.558 | 0.363 | 0.798 | 0.307 | 0.272 | 0.838 | 0.105 | 0.202 | 0.300 | 0.390 | 0.205 | 0.232 | 0.823 | 0.412 |

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. This is an extension of Table 3 in the paper.

| Task wise results on Faster-RCNN detections, mAP@0.5 | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| task # | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | **avg.** |
| detection stats | 0.011 | 0.256 | 0.020 | 0.007 | 0.039 | 0.004 | 0.006 | 0.007 | 0.015 | 0.017 | 0.008 | 0.036 | 0.118 | 0.000 | 0.039 |
| classification | 0.023 | 0.310 | 0.091 | 0.016 | 0.138 | 0.015 | 0.182 | 0.039 | 0.058 | 0.180 | 0.050 | 0.038 | 0.106 | 0.006 | 0.090 |
| proposed + fusion | 0.018 | 0.379 | 0.097 | 0.032 | 0.151 | 0.050 | 0.202 | 0.030 | 0.046 | 0.290 | 0.090 | 0.068 | 0.145 | 0.017 | **0.115** |

| Task wise results on ground truth bounding boxes, mAP@0.5 | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| task # | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | **avg.** |
| detection stats | 0.021 | 0.309 | 0.106 | 0.205 | 0.175 | 0.238 | 0.300 | 0.039 | 0.060 | 0.091 | 0.023 | 0.139 | 0.123 | 0.439 | 0.162 |
| classification | 0.032 | 0.320 | 0.208 | 0.129 | 0.199 | 0.082 | 0.292 | 0.104 | 0.261 | 0.223 | 0.113 | 0.403 | 0.161 | 0.225 | 0.197 |
| proposed + fusion | 0.033 | 0.412 | 0.257 | 0.205 | 0.230 | 0.171 | 0.445 | 0.064 | 0.181 | 0.373 | 0.151 | 0.350 | 0.222 | 0.368 | **0.247** |

Table 4. Task wise results on subset of the test images where an object belonging to the most frequently chosen category was neglected in favor of an object from a less frequently chosen category.

ject. Consider the pie charts in Figure 2. The three tasks with mAP@0.5 higher than 90% on ground truth detections can typically be solved with sports tools and these occur in sports images. In these images, only a small number of objects of one selected category occurs with little functional intra class differentiation. Therefore selecting all objects in this category is a good choice. Contrary to that, the five tasks where our proposed method gives less than 70% on ground truth detections, have to be solved with vessels or cutlery. These objects typically occur on kitchen or party images, showing a high number of selected categories and multiple small objects of each category. Especially for the task "open bottle of beer", different objectst of the same category might have a suitable ridge or not, i.e. exhibit a high functional intra class variance.

### 2.6. Further Investigation of Scene Context Learned by GGNN

In Section 5.3, we quantitatively showed that GGNN learns information about the scene context. We confirm this observation by another experiment. We measure the absolute difference between the number of instances belonging to the COCO category of the query object in the example image and the retrieved images. When using nearest neighbors according to the $\ell_2$ distance between $h_i^0$, the difference is far higher than when using $h_i^T$ (Figure 4). This implies that the GGNN also improves the ability of the model to implicitly count the objects.

We also provide some qualitative nearest neighbor comparisons. In Figures 5 to 18, we show the query object with a red bounding box on the left and the top 5 nearest neighbor objects from the test set of each task based on $h_i^0$ and $h_i^T$ on the top row and the bottom row, respectively. In general, we can see that the retrieved images based on the $h_i^T$ distance show a more similar scene configuration. We would like to mention that although the displayed examples show better scene configuration similarity for the retrieved images based on $h_i^T$, this is not always the case and qualitative examples are sometimes hard to interpret.

## References

[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CVPR*, 2016. 1

[2] Joseph Redmon and Ali Farhadi. YOLO 9000: Better, faster, stronger. *CVPR*, 2017. 4, 6

| Task wise results, mAP@0.5 | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| task # | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | **avg.** |
| gt bboxes | 0.831 | 0.853 | 0.732 | 0.934 | 0.699 | 0.690 | 0.961 | 0.370 | 0.804 | 0.810 | 0.638 | 0.803 | 0.706 | 0.960 | 0.771 |
| Faster-RCNN dets | 0.366 | 0.298 | 0.405 | 0.376 | 0.410 | 0.172 | 0.436 | 0.179 | 0.210 | 0.406 | 0.223 | 0.284 | 0.391 | 0.407 | 0.326 |
| Yolov2 dets | 0.368 | 0.319 | 0.391 | 0.380 | 0.416 | 0.165 | 0.444 | 0.187 | 0.230 | 0.390 | 0.223 | 0.269 | 0.440 | 0.420 | 0.332 |

Table 5. We present the task wise numbers of the proposed method on Faster-RCNN [3] detections, the Yolov2 [2] detection and on ground truth bounding boxes.

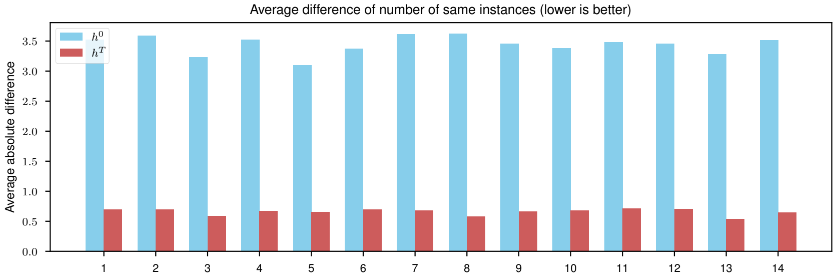


Figure 4. Average difference of the number of same instances between the query image and the nearest neighbor images based on $h_i^0$ vs. $h_i^T$.

[3] Shaoqing Ren, Kaiming He, Ross Girshik, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. *ICCV*, 2015. 1, 2, 4, 6

[4] Jianwei Yang, Jiasen Lu, Dhruv Batra, and Devi Parikh. A faster pytorch implementation of faster R-CNN. *https://github.com/jwyang/faster-rcnn.pytorch*, 2017. 1

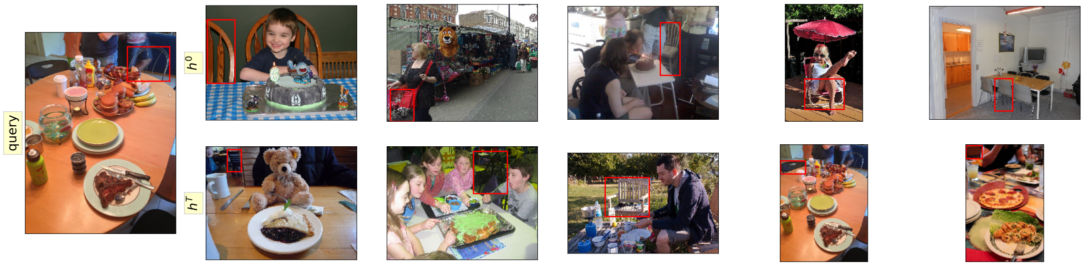


Figure 5. A qualitative example from Task 1 test set comparing nearest neighbors according to $h_i^0$ vs. $h_i^T$.

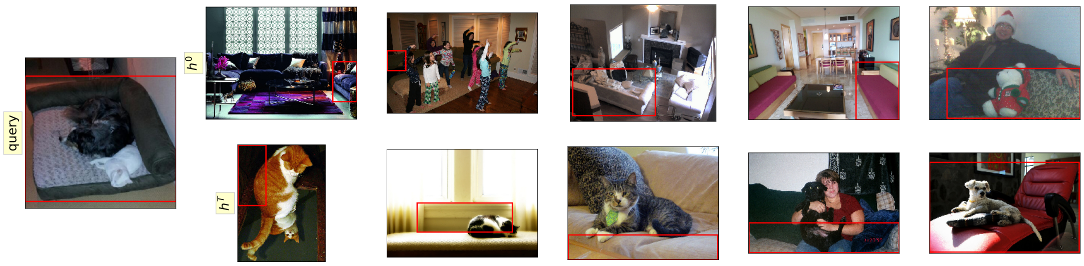


Figure 6. A qualitative example from Task 2 test set comparing nearest neighbors according to $h_i^0$ vs. $h_i^T$.

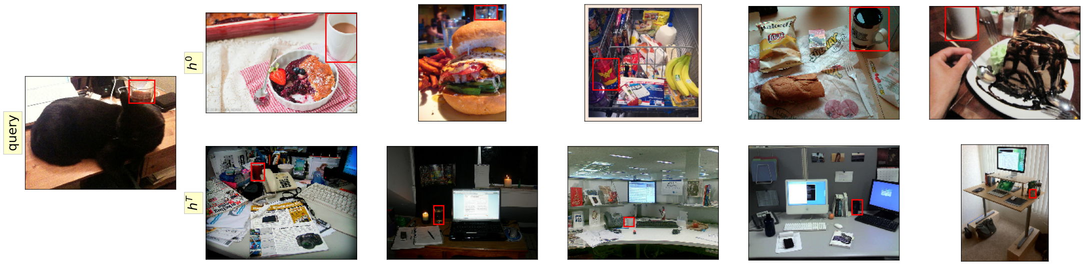


Figure 7. A qualitative example from Task 3 test set comparing nearest neighbors according to $h_i^0$ vs. $h_i^T$.

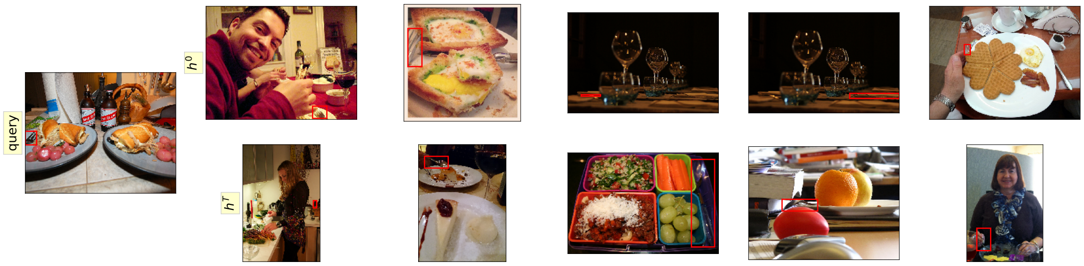


Figure 8. A qualitative example from Task 4 test set comparing nearest neighbors according to $h_i^0$ vs. $h_i^T$.

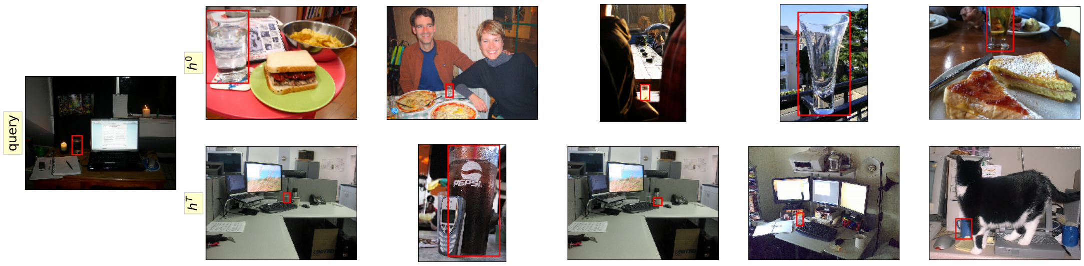


Figure 9. A qualitative example from Task 5 test set comparing nearest neighbors according to $h_i^0$ vs. $h_i^T$.

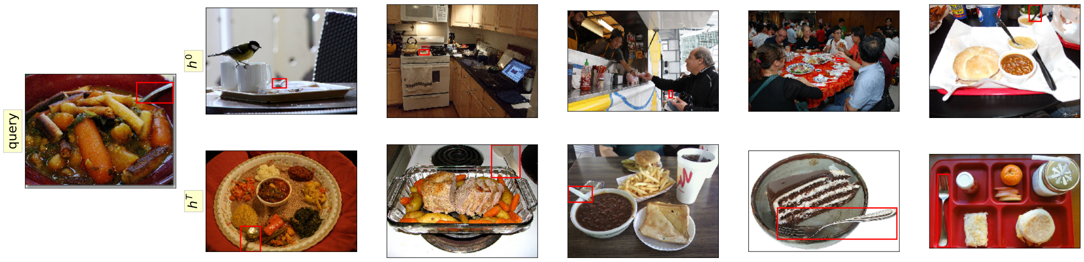


Figure 10. A qualitative example from Task 6 test set comparing nearest neighbors according to $h_i^0$ vs. $h_i^T$.

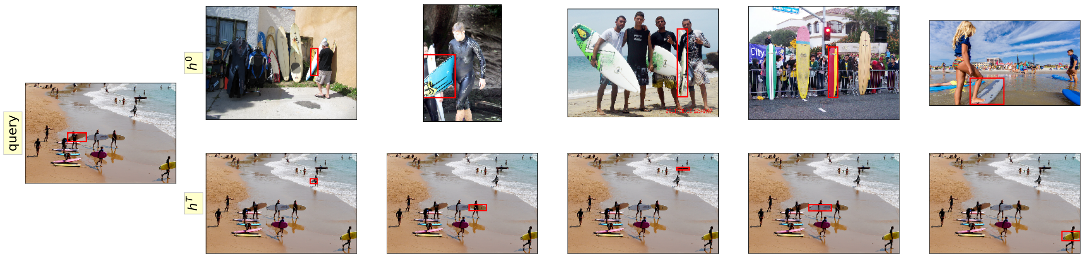


Figure 11. A qualitative example from Task 7 test set comparing nearest neighbors according to $h_i^0$ vs. $h_i^T$.

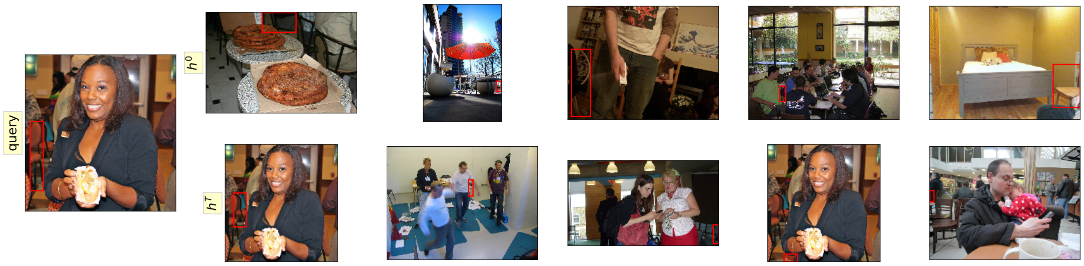


Figure 12. A qualitative example from Task 8 test set comparing nearest neighbors according to $h_i^0$ vs. $h_i^T$.

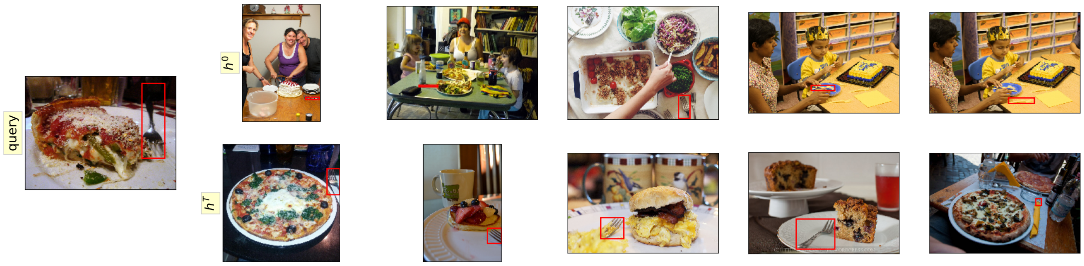


Figure 13. A qualitative example from Task 9 test set comparing nearest neighbors according to $h_i^0$ vs. $h_i^T$.

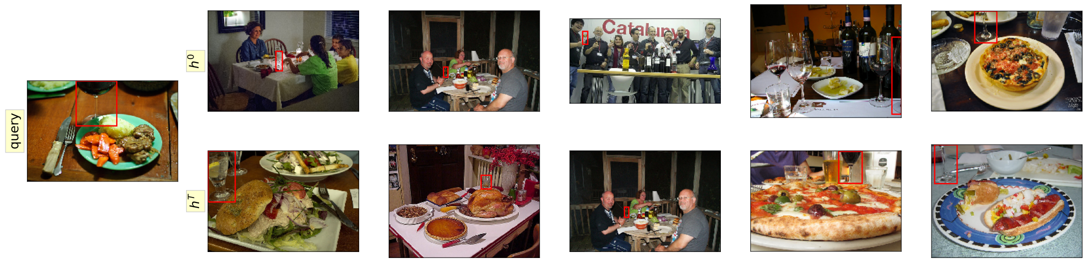


Figure 14. A qualitative example from Task 10 test set comparing nearest neighbors according to $h_i^0$ vs. $h_i^T$.

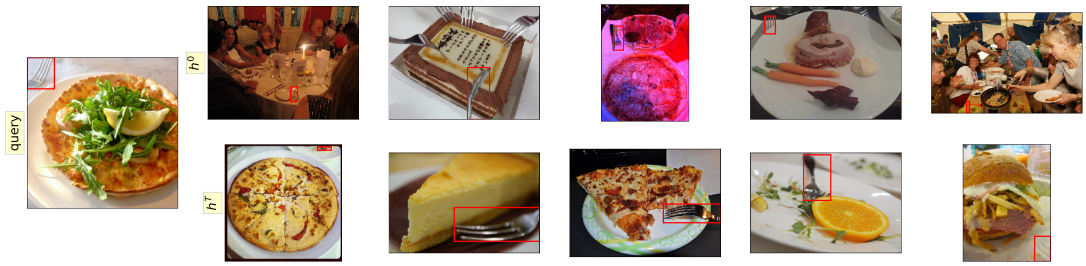


Figure 15. A qualitative example from Task 11 test set comparing nearest neighbors according to $h_i^0$ vs. $h_i^T$.

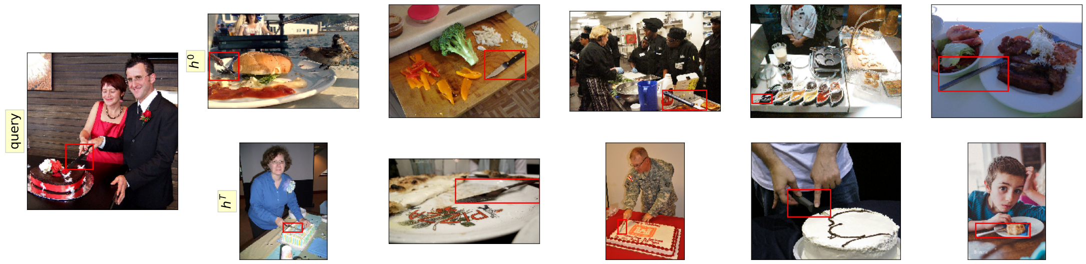


Figure 16. A qualitative example from Task 12 test set comparing nearest neighbors according to $h_i^0$ vs. $h_i^T$.

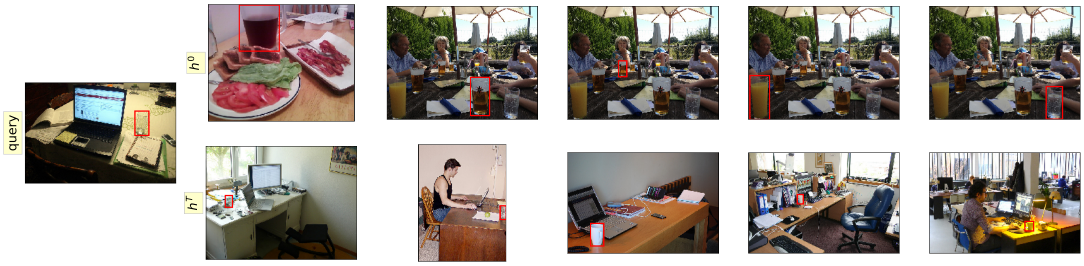


Figure 17. A qualitative example from Task 13 test set comparing nearest neighbors according to $h_i^0$ vs. $h_i^T$.

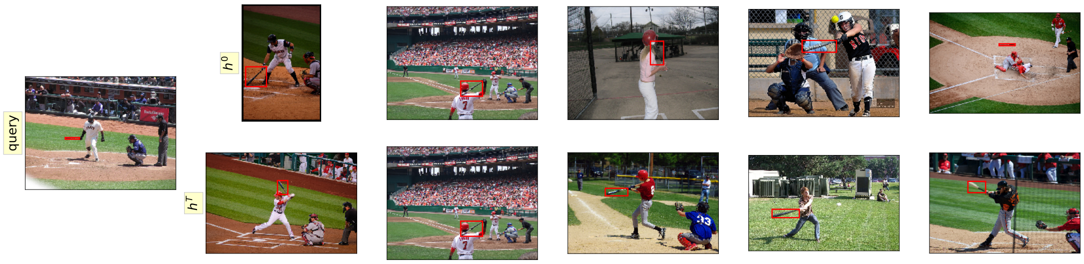


Figure 18. A qualitative example from Task 14 test set comparing nearest neighbors according to $h_i^0$ vs. $h_i^T$.