CAT: LoCalization and IdentificAtion Cascade Detection Transformer for Open-World Object Detection

Shuailei Ma [1] Yuefeng Wang[1] Ying Wei[1 2] Jiaqi Fan[1]
Thomas H. Li[3] Hongli Liu[4] Fanbing Lv[4]

[1]Northeast University, Shenyang, China [2]Information Technology R&D Innovation Center of Peking University
[3]School of Electronic and Computer Engineering, Peking University Shenzhen Graduate School, Shenzhen, China
[4]Changsha Hisense Intelligent System Research Institute Co., Ltd

Abstract

*Open-world object detection (OWOD), as a more general and challenging goal, requires the model trained from data on known objects to detect both known and unknown objects and incrementally learn to identify these unknown objects. The existing works which employ standard detection framework and fixed pseudo-labelling mechanism (PLM) have the following problems: (i) The inclusion of detecting unknown objects substantially reduces the model's ability to detect known ones. (ii) The PLM does not adequately utilize the priori knowledge of inputs. (iii) The fixed selection manner of PLM cannot guarantee that the model is trained in the right direction. We observe that humans subconsciously prefer to focus on all foreground objects and then identify each one in detail, rather than localize and identify a single object simultaneously, for alleviating the confusion. This motivates us to propose a novel solution called CAT: Lo**C**alization and Identific**A**tion Cascade Detection **T**ransformer which decouples the detection process via **the shared decoder** in **the cascade decoding way**. In the meanwhile, we propose the **self-adaptive pseudo-labelling mechanism** which combines the model-driven with input-driven PLM and self-adaptively generates robust pseudo-labels for unknown objects, significantly improving the ability of CAT to retrieve unknown objects.*

1. Additional Experiments Material

1.1. Theory For Self-Adaptive Pseudo-labelling

For $0 < w_2 < w_1 < 1$, we find the potential relationship as follows:

$$\begin{cases} x^{w_1} > x^{w_2}, if \;\; x > 1 \\ x^{w_1} < x^{w_2}, if \;\; x < 1 \end{cases} \tag{1}$$

Thus, for $x^{w_1} \cdot y^{w_2}$ and $w_1 > w_2$, if $x > 1$ and $y > 1$, x weights more and y weights more if $x < 1$ and $y < 1$.

For the self-adaptive pseudo-labelling, we first normalize s_o to the range 0 to 1. Considering that the model itself has little knowledge in the early stages of model training, the model-driven pseudo-labelling should weigh less than the input-driven pseudo-labelling. As the training time of the model increases, the knowledge base of the model grows, and the weight of the model-driven pseudo-labelling gets bigger. Combining this with the patterns above, we update them as follows:

$$\begin{cases} \mathcal{W}_m{}^t = \mathcal{W}_m{}^{t-1} + \Delta w \times \mathcal{W}_m{}^{t-1}, \\ \mathcal{W}_I{}^t = \mathcal{W}_I{}^{t-1} - \Delta w \times \mathcal{W}_I{}^{t-1}, \\ \mathcal{W}_m^t, \mathcal{W}_I{}^t = norm\left(\mathcal{W}_m^t, \mathcal{W}_I{}^t\right), \end{cases} \tag{2}$$

1.2. Additional Illustration For Data Split

As shown in Table.1, the OWOD split proposed in ORE groups all VOC classes and data as $Task$ 1. The remaining 60 classes of MS-COCO are grouped into three successive tasks ($Task$ 2, 3, 4) with semantic drifts. However, it leads data leakage across tasks since different classes which belong to a super-categories are introduced in different tasks. The MS-COCO split proposed in OW-DETR is a stricter split, where all the classes of a super-categories are introduced at a time in a task. For OWOD split, Task 1 contains 16,551 training images and 4,952 testing images. Task 2 contains 45,520 images in training set and 1,914 images in testing set. For Task 3, there are 39,402 images in training set and 1,642 images in testing set. Task 4 consists of 40,260 training images and 1,738 testing images. For MS-COCO, there are 89,490 training images and 3,793 testing images in Task 1. For Task 2, there are 55,870 images in training set and 2,351 images in testing set. Task 3 contains 39,402 images in training set and 1,642 images in testing set. Task 4 contains 38,902 images in training set and 1,691 images in testing set.

Table 1. The table shows task composition in the OWOD and MS-COCO split for Open-world evaluation protocol. The semantics of each task and the number of images and instances(objects) across splits are shown.

Task ID	**Task 1**	**Task 2**	**Task 3**	**Task 4**
		OWOD split		
Semantic split	VOC Classes	Outdoor, Accessories, Appliances, Truck	Sports, Food	Electronic, Indoor, Kitchen, Furniture
# training images	16551	45520	39402	40260
# test images	4952	1914	1642	1738
# train instances	47223	113741	114452	138996
# test instances	14976	4966	4826	6039
		MS-COCO split		
Semantic split	Animals,Person, Vehicles	Appliances, Accessories, Outdoor, Furniture	Sports, Food	Electronic, Indoor, Kitchen
# training images	89490	55870	39402	38903
# test images	3793	2351	1642	1691
# train instances	421243	163512	114452	160794
# test instances	17786	7159	4826	7010

1.3. Additional Implementation Details

For selective search, we use the *selective_search* function in Selectivesearch library and the scale, sigma, min_size of parameter is set to 500, 0.9 and 200, respectively. In addition, we eliminate candidate boxes with less than 2000 pixel points. The multi-scale feature maps extracted from the backbone are projected to feature maps with 256-channels using 1×1 convolution filters and used as multi-scale input to deformable transformer encoder. The PyTorch library and eight NVIDIA RTX 3090 GPUs are used to train our CAT framework with a batch size of 3 images per GPUs. In each task, the CAT framework is trained for 50 epochs and finetuned for 20 epochs during the incremental learning step. We train our CAT using the Adam optimizer with a base learning rate of 2×10^{-4}, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and weight decay of 10^{-4}. For finetuning during incremental step, the learning rate is reduced by a factor of 10 and trained using a set of 50 stored exemplars per known class.

2. Additional Results

2.1. Additional Qualitative Results

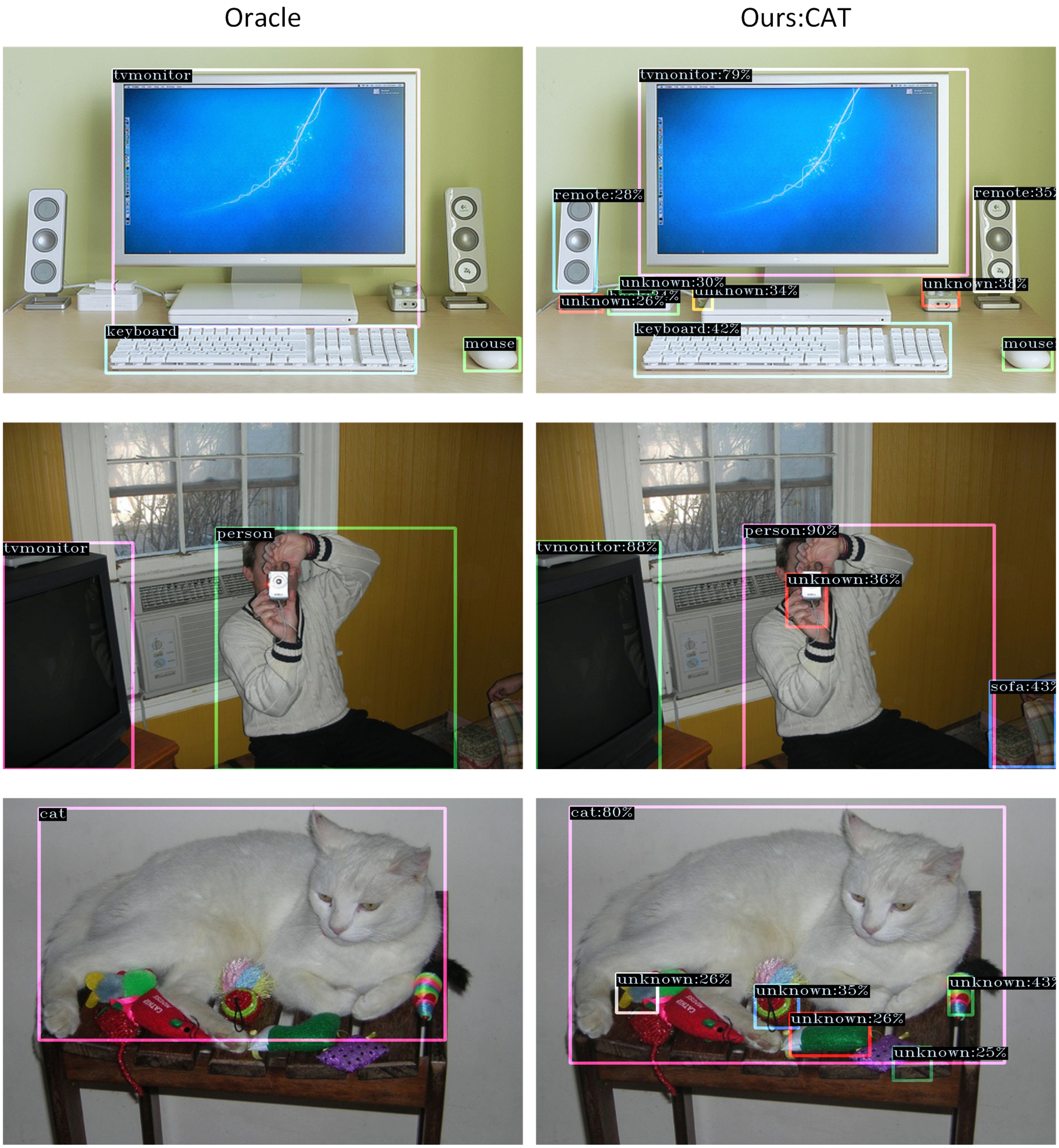
Figure.1 describes the visualization results comparison between CAT and Oracle. We visualize the detection results of our model for known and unknown objects, as well as the ground truth on the tasks corresponding to the weights, including the labels of known and unknown categories, where the objects of unknown categories are the objects of other categories that have not yet appeared in the total categories of the dataset. Our model can accurately detect known objects and unknown objects outside the total class of the dataset, such as the *electric plug* and *sound switch* in the first row, the *camera* in the second row and the *kitten toy* in the third row. It is also worth noting that although our model detects the *audio*, it does not identify it as an unknown object, but as a *remote*, showing the limitations of our model.

Figure.2 exhibits the visualization performance on incremental object detection. We visualize the detection results of the weights corresponding to different tasks for the same scenario. The results show that our CAT can identify unknown kinds of objects as the unknown class and accurately identify their classes after incrementally learning the unknown classes, such as *sports ball* and *tennis racket* in the first row, *surfboard* in the second row and *traffic light* in the third row.

3. Societal Impact and Limitations

Open-world object detection makes artificial intelligence smarter to face more problems in real life. It takes object detection to a cognitive level, as the model requires more than simply remembering the objects learned, it requires deeper thinking about the scene.

Although our results demonstrate significant improvements over ORE and OW-DETR in terms of WI, A-OSE, U-Recall and mAP, the performances are still on the lower side due to the challenging nature of the open-world detection problem. In this paper, we are mainly committed to enhance the model's ability to explore unknown classes. However, the confidence level of our model for the detection of unknown objects still needs to be improved, and this is what we will strive for in the future.


Oracle

Ours:CAT

Figure 1. Visualization results comparison between CAT and Oracle. We visualize the detection results of our model for known and unknown objects, as well as the ground truth on the tasks corresponding to the weights, including the labels of known categories and the labels of unknown categories, where the objects of unknown categories are the objects of other categories that have not yet appeared in the total categories of the dataset. Our model can accurately detect known objects and unknown objects outside the total class of the dataset, such as the *electric plug* and *sound switch* in the first row, the *camera* in the second row and the *kitten toy* in the third row. It is also worth noting that although our model detects the *audio*, it does not identify it as an unknown object, but as a *remote*, showing the limitations of our model.

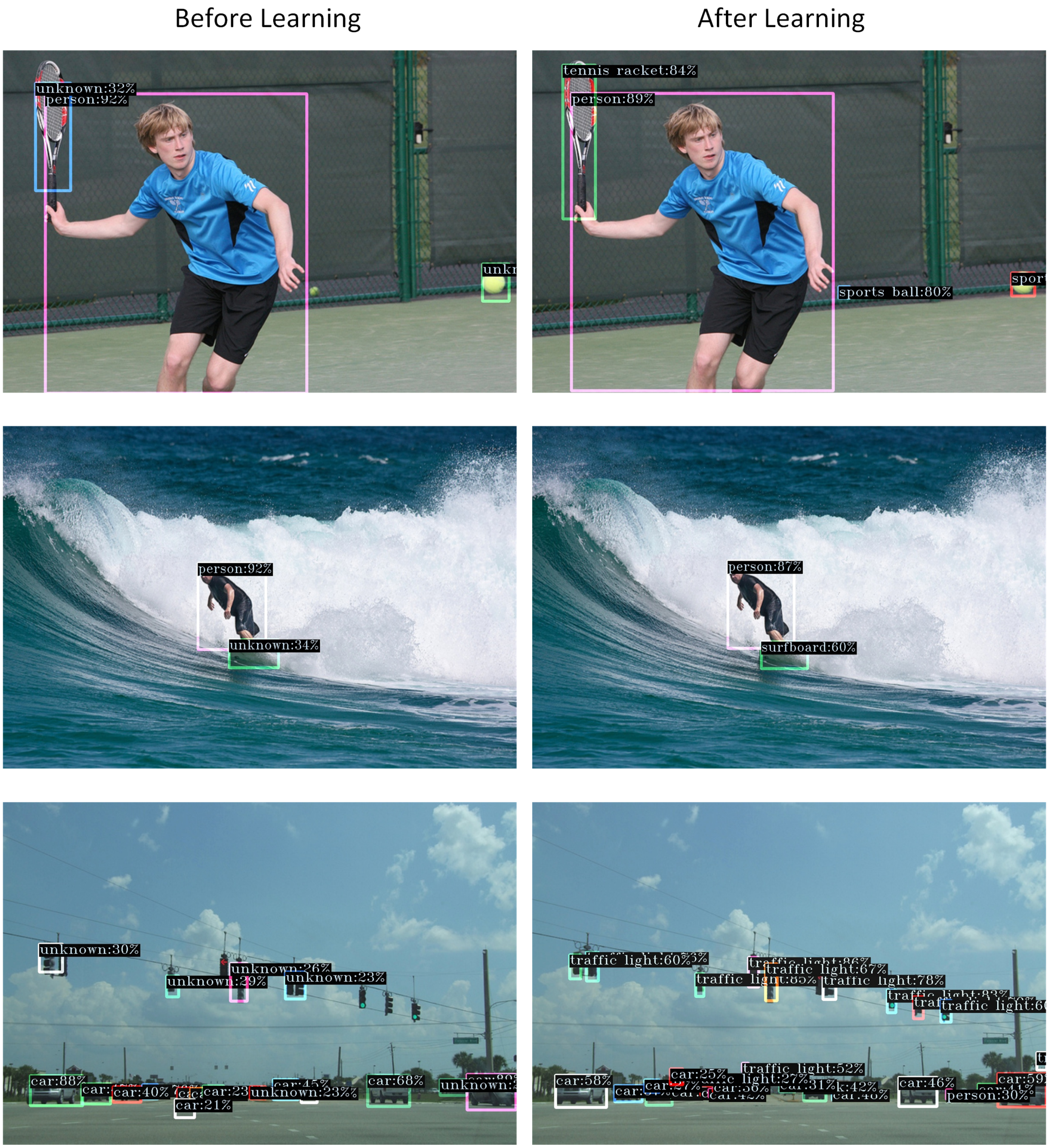


Figure 2. Visualization performance on incremental object detection. We visualize the detection results of the weights corresponding to different tasks for the same scenario. The results show that our CAT can identify unknown kinds of objects as the unknown class and accurately identify their classes after incrementally learning the unknown classes, such as *sports ball* and *tennis racket* in the first row, *surfboard* in the second row and *traffic light* in the third row.