Unconstrained Foreground Object Search: Supplementary Materials

Yinan Zhao*, Brian Price+, Scott Cohen+, Danna Gurari*
* University of Texas at Austin, + Adobe Research
yinanzhao@utexas.edu, {bprice,scohen}@adobe.com, danna.gurari@ischool.utexas.edu

This document supplements our methods and results provided in the main paper. In Section 1, we describe the sampling module used to accelerate identifying plausible foreground objects for training the encoder (supplements Section **3.3**). In Section 2, we describe how we decompose an image into a background scene and foreground object when we create the dataset from MS-COCO [3] (supplements Section **4.2**). In Section 3, we show more quantitative and qualitative results in CAIS [10] (supplements Section **4.1**). In Section 4, we show more qualitative results in MS-COCO [3] (supplements Section **4.2**). In Section 5, we show a quantitative analysis of retrieval diversity in MS-COCO [3] (supplements Section **4.2**). In Section 6, we show two applications of our *UFO Search*: hole-filling and compositing (supplements Figure **1** in the main paper).

1. Sampling Module (supplements Section 3.3 in main paper)

To speed up the discriminator's role in generating training data, in the main paper we introduce a sampling module to accelerate sampling plausible foreground objects for training the encoder. Specifically, we put in the sampling set some random foreground objects, the top K_C most similar background scenes and top K_G most similar foreground objects. To measure the similarity between objects and background scenes, we use VGG-19 [6] pretrained for the ILSVRC-2014 competition [5] to extract features for foreground objects and background scenes respectively. We normalize the feature vectors to unit length and measure the similarities between objects and between background scenes by cosine similarity. To speed up similarity search, we build an index for the foreground object database and background scene database separately by Faiss [2], a library for efficient similarity search and clustering of dense vectors.

2. Decomposition of Foreground and Background (supplements Section 4.2 in main paper)

In Section **4.2** of the main paper, we employ MS-COCO [3] to create a diverse dataset of 79 foreground object categories for evaluation. We use the annotated object instance segmentation mask to decompose each image into foreground objects and a background scene. Specifically, to create a foreground image of input size 224x224, we segment out the foreground object using the instance segmentation mask, and then overlay it at the center of a square 224x224 image consisting of pixels set to the (ImageNet) image mean. As for the background image, we first crop it to the size 224x224 that is required by our network. We then create the hole by removing the original object in the annotated object instance segmentation mask as well as all pixels in a bounding rectangle around the object. We then set the pixels in the hole to the (ImageNet) image mean.

3. Quantitative and Qualitative Results in CAIS (supplements Section 4.1 in main paper)

In Section **4.1** of the main paper, we evaluate our method quantitatively in CAIS [3]. In this section, we show more quantitative and qualitative results in CAIS [3].

In Table 1, we show mAP results with respect to all the retrievals for two baselines, *UFO Search* and ablated variants of *UFO Search* in CAIS [10]. Note that we do not show results of *CFO* methods because they do not rank all objects in all categories. Overall, our *UFO Search* outperforms other baselines and variants; e.g. mAP is **29.03** for *UFO Search*, which is **3.34** percentage points improvement over the next best ablated variant. *UFO Search* outperforms other baselines and ablated variants in the following five object categories: boat, bottle, car, painting and plant. These findings align with and reinforce those in the main paper.

We also show qualitative results in Figure 1. In the top two examples, our *UFO Search* retrieves objects from the only compatible object type in the eight categories of CAIS [3] (boat and bottle, respectively). Note that the compatible foreground

objects depend on both hole shape and context in this dataset with various hole shapes. The bottom two examples demonstrate failure cases. For the second last example, our method retrieves chairs for a hole where there is supposed to be a plant. For the last example, our approach also retrieves chairs to fill the hole where bottles are compatible. Chairs are reasonable objects to be present in both scenes. However, they are not compatible in the given holes that are placed on top of tables. In other words, at times, our method fails to capture the spatial relationship between inserted objects and existing objects in both scenes. Our current model does not explicitly model spatial relationship between objects. It is interesting to explore in future work how to model the spatial relationship between objects more effectively.

Method	boat	bottle	car	chair	dog	painting	person	plant	overall
Shape [10]	6.53	3.69	9.90	4.09	10.89	2.56	4.74	5.55	5.99
RealismCNN [12]	5.84	6.10	3.55	1.50	5.68	2.66	3.27	7.58	4.52
Ours: No BG Training	34.19	4.61	3.98	4.84	10.69	8.70	8.81	16.48	11.54
Ours: Discriminator Only	32.99	8.68	17.67	13.13	**35.67**	16.29	17.58	13.62	19.45
Ours: Regression	49.31	11.82	16.98	12.09	28.07	21.35	**30.11**	10.44	22.52
Ours: No Discriminator	52.85	16.94	19.62	**16.13**	31.92	21.17	24.11	22.80	25.69
Ours: UFO Search	**56.64**	**23.62**	**31.63**	13.77	33.39	**24.33**	23.94	**24.93**	**29.03**

Table 1. *Mean Average Precision* with respect to all the retrievals. Results are shown as percentages.

4. Qualitative Results in MS-COCO (supplements Section 4.2 in main paper)

In the main paper, we conduct user evaluation and show qualitative results in MS-COCO [3]. We show more qualitative results in Figure 2 and Figure 3 to exemplify our performance in the more diverse dataset consisting of 79 categories. If there is only one object type that is compatible with the context, our method retrieves only those objects. Our approach also has the potential to retrieve compatible objects of different categories when many object types are appropriate for the scene.

5. Quantitative Analysis of Retrieval Diversity (supplements Section 4.2 in main paper)

In Figure **5** of the main paper, we qualitatively demonstrate that our *UFO Search* has the potential to retrieve compatible objects of different categories when many object types are appropriate for the scene. In this section, we measure retrieval diversity quantitatively. To do so, we compute the average number of categories compatible objects span in the top 25 retrievals of our user study in Section **4.2**. Results are shown in Table 2. Note that we measure diversity on compatible objects only instead of all the retrievals. Otherwise, a random guess would have a large diversity although most of its retrievals would be incompatible.

Method	Diversity ($\pm$ Std Dev)	P@25
UFO Search	1.90 ($\pm$ 1.54)	38.83
No Discriminator	1.82 ($\pm$ 1.30)	35.57
Discriminator Only	2.72 ($\pm$ 3.06)	35.77
Regression	2.56 ($\pm$ 2.34)	30.40
No BG Training	1.52 ($\pm$ 0.79)	12.50

Table 2. Quantitative analysis of diversity for our *UFO Search* method and its four ablated variants. We report the average number of categories compatible objects span in the top 25 retrievals along with the standard deviation. For completeness, we also report retrieval performance results (P@25) from Table **2** in the main paper.

The findings reveal that the increased diversity of *UFO Search* over *No Discriminator* may come from the discriminator, as evidenced by the fact that *Discriminator Only* has the largest diversity among the shown five methods. While *Regression* has the largest diversity in encoder-based methods (*UFO Search*, *No Discriminator*, *Regression* and *No BG Training*), the percentage of compatible objects it retrieves is significantly lower than *UFO Search* and *No Discriminator*.

6. Applications (supplements Figure 1 in main paper)

For completeness, we show two applications of *UFO Search*: hole-filling in Figure 4 and compositing in Figure 5. Specifically, we use our *UFO Search* to retrieve compatible foreground objects, and then use them to aid (1) hole-filling and (2)

compositing them directly into the background. For hole-filling, we overlay the object in the center of the hole, and fill the smaller gap around it using PatchMatch [1], as shown in Figure 4. For compositing, we insert the retrieved object at the center of the specified yellow rectangle, as shown in Figure 5. Note that the hole-filling and compositing results would look more natural with harmonization [7, 8, 4, 9, 11]. We are not applying such methods since that is not the main focus of this paper.

References

[1] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. PatchMatch: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 28(3), Aug. 2009. 3, 7

[2] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. *arXiv preprint arXiv:1702.08734*, 2017. 1

[3] 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 *European conference on computer vision*, pages 740–755. Springer, 2014. 1, 2, 5, 6

[4] Patrick Pérez, Michel Gangnet, and Andrew Blake. Poisson image editing. *ACM Transactions on graphics (TOG)*, 22(3):313–318, 2003. 3

[5] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. 1

[6] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1

[7] Kalyan Sunkavalli, Micah K Johnson, Wojciech Matusik, and Hanspeter Pfister. Multi-scale image harmonization. In *ACM Transactions on Graphics (TOG)*, volume 29, page 125. ACM, 2010. 3

[8] Michael W Tao, Micah K Johnson, and Sylvain Paris. Error-tolerant image compositing. In *European Conference on Computer Vision*, pages 31–44. Springer, 2010. 3

[9] Yi-Hsuan Tsai, Xiaohui Shen, Zhe Lin, Kalyan Sunkavalli, Xin Lu, and Ming-Hsuan Yang. Deep image harmonization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, 2017. 3

[10] Hengshuang Zhao, Xiaohui Shen, Zhe Lin, Kalyan Sunkavalli, Brian Price, and Jiaya Jia. Compositing-aware image search. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 502–516, 2018. 1, 2, 4, 7, 8

[11] Yinan Zhao, Brian Price, Scott Cohen, and Danna Gurari. Guided image inpainting: Replacing an image region by pulling content from another image. *arXiv preprint arXiv:1803.08435*, 2018. 3

[12] Jun-Yan Zhu, Philipp Krahenbuhl, Eli Shechtman, and Alexei A Efros. Learning a discriminative model for the perception of realism in composite images. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3943–3951, 2015. 2

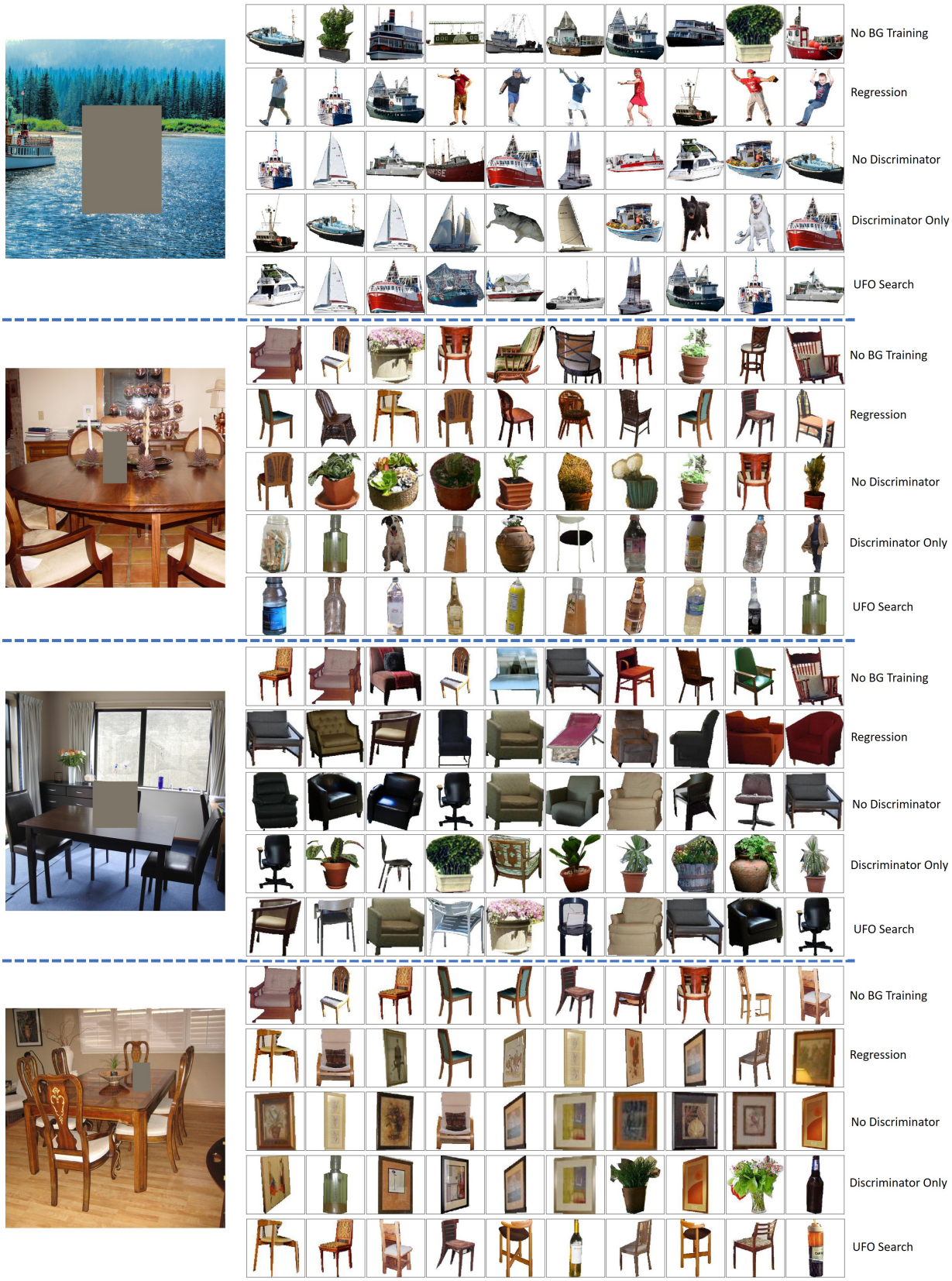

Figure 1. Qualitative results in CAIS [10]. In the top two examples, *UFO Search* retrieves objects from the only compatible object type in the eight categories (boat and bottle, respectively). The last two examples demonstrate failure cases where our method fails to capture the spatial relationship between inserted objects and existing objects in the scene.

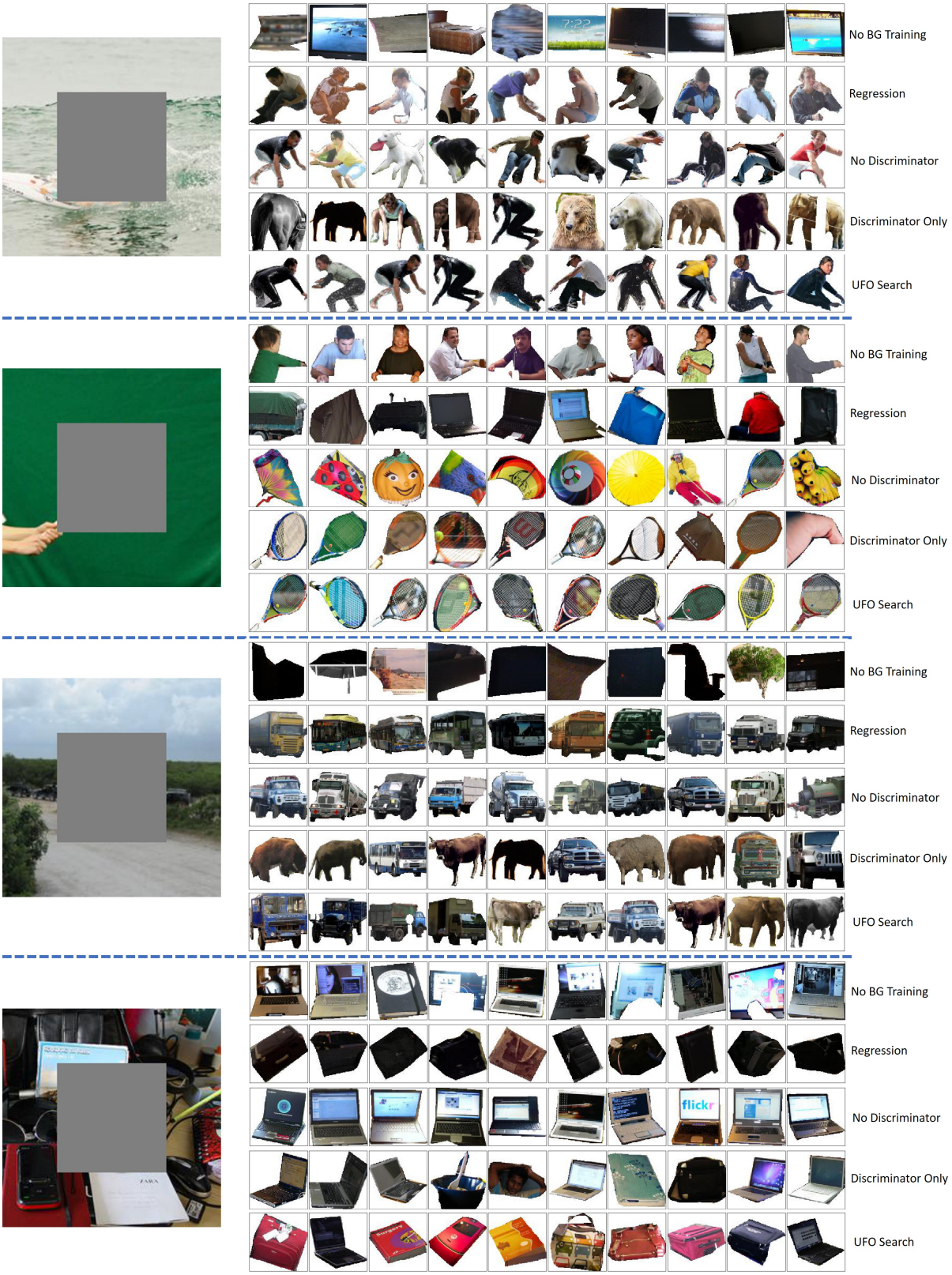

Figure 2. Qualitative results in MS-COCO [3]. In the first two examples, *UFO Search* retrieves objects from the only object type in MS-COCO (surfer and racket, respectively) that is really compatible with the context. The bottom two examples show that our method can retrieve compatible objects from differing categories when numerous object types are appropriate for the scene. For example, our method retrieves truck, jeep and cattle for a hole on a road (second to bottom example) and retrieves bag, book and laptop for a hole on a cluttered table (bottom example).

Figure 3. Qualitative results in MS-COCO [3]. In the top three examples, *UFO Search* retrieves objects from the best fitting category in MS-COCO (motorcycle, bag and cup, respectively) in the top-ranked retrievals. In the bottom example, *UFO Search* retrieves compatible objects from differing categories, e.g. horse, zebra and elephant.

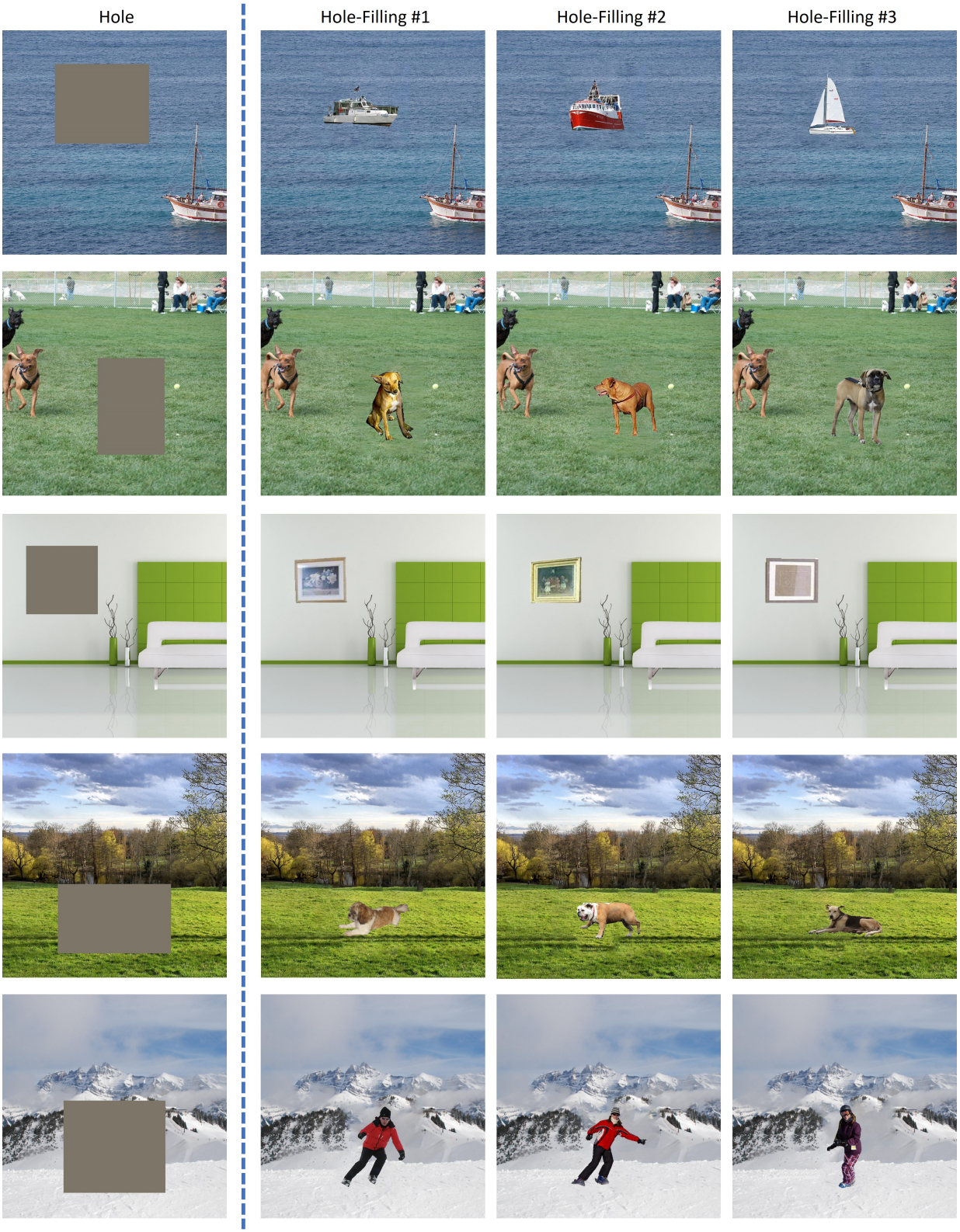

Figure 4. Application of *UFO Search* in hole-filling. We show the background image with a hole on the left, and three different hole-filling results on its right. To fill the hole, we pick a top-ranked foreground object retrieved by our *UFO Search*, overlay the object in the center of the hole, and fill the smaller gaps around it using PatchMatch [1]. The shown background image and foreground objects are in the test set of CAIS [10]. Note that there are not a diversity of object types being retrieved for the shown background images since most background images in CAIS [10] unambiguously match only one assigned foreground object category from the few candidate categories represented.

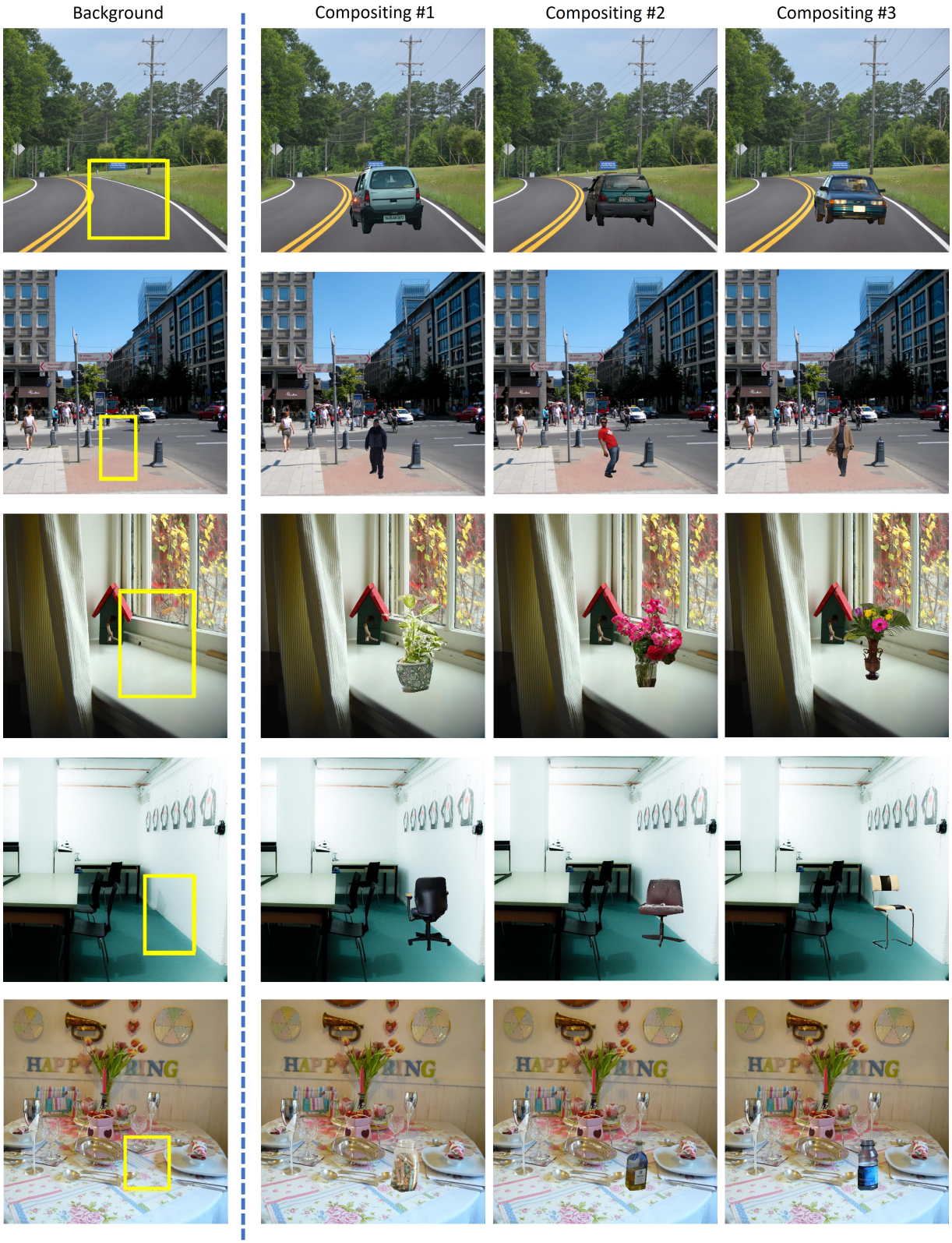

Figure 5. Application of *UFO Search* in compositing. We show on the left the background image with a yellow rectangle indicating the position to insert the object, and three different compositing results on its right. The shown background image and foreground objects are in the test set of CAIS [10]. Note that there are not a diversity of object types being retrieved for the shown background images since most background images in CAIS [10] unambiguously match only one assigned foreground object category from the few candidate categories represented.