Unconstrained Foreground Object Search

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

Abstract

Many people search for foreground objects to use when editing images. While existing methods can retrieve candidates to aid in this, they are constrained to returning objects that belong to a pre-specified semantic class. We instead propose a novel problem of unconstrained foreground object (UFO) search and introduce a solution that supports efficient search by encoding the background image in the same latent space as the candidate foreground objects. A key contribution of our work is a cost-free, scalable approach for creating a large-scale training dataset with a variety of foreground objects of differing semantic categories per image location. Quantitative and human-perception experiments with two diverse datasets demonstrate the advantage of our UFO search solution over related baselines.

1. Introduction

Image-based search, the task of retrieving images based on an image query, is a popular research problem with many applications [16, 23, 1, 28, 7]. While it is often used to find visually or semantically similar images to the query image, a less explored subproblem in this domain is searching for content to edit the query image. Yet the importance of this subproblem is evidenced by the existence of many stock image websites, for example `shutterstock.com`, `www.istockphoto.com`, and `stock.adobe.com` to name a few, which contain tens of millions of images of objects on a white or plain background to make it easy to cut out just the foreground object to use it in another image. Whether a user is placing an object on top of a complete image (compositing) or using an object to partially fill a hole (created, for example, by removing another object or area), an important part of the creative process is to find a large variety of content that is compatible with the surrounding background in order to explore multiple possible outcomes.

The most relevant related work to this subproblem are compositing-aware methods which require a user to specify the desired object type to be pasted into a query image, and then search for suitable objects [11, 30].[1] While specifying the object type to be inserted can guide the search process, it also introduces a limitation that creatives cannot explore many possible image modifications representing a variety of objects that can be inserted into a query image believably.

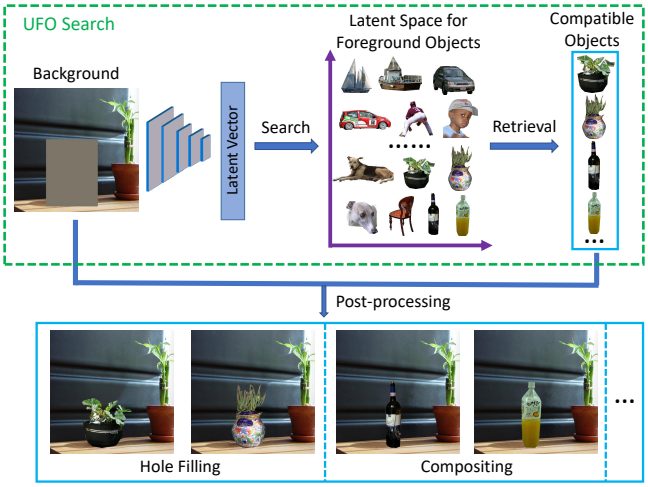

Figure 1. We propose a method to search for foreground objects that are semantically compatible with a background image. In this example, our approach takes the background image with a hole on the table, searches in a large object database consisting of multiple semantic classes, and returns compatible foreground objects. This example illustrates how UFO search can be used for hole filling (using [4] to fill in the gaps around the object) and compositing.

In this paper, we propose the problem of unconstrained foreground object search (UFO search). Specifically, the goal is to *search for foreground objects that are semantically compatible with a background image* without any constraint on what objects to retrieve. An object is compatible with a background image if it can be realistically composited into the image or used to aid hole filling, as illustrated in Figure 1. Here, we focus on *semantic* compatibility as other methods address correcting geometrical errors [12, 2] and low-level color and appearance differences [24, 31].

[1] Of note, search is a valuable approach since deep learning based methods that synthesize realistic-looking content are unable to do so for large holes with complex surrounding structures [15, 27, 8, 29].

We also introduce a novel solution for UFO search. Inspired by [30], our network projects background images and foreground objects into a high-level feature space, *without* requiring object labels, such that compatible objects and backgrounds are near each other. These high-level features are then used for efficient search. A key contribution of our work is a cost free, scalable approach for creating a large (noisy) dataset for training unconstrained foreground object search methods. Experiments demonstrate the effectiveness of our UFO search method over numerous related baselines.

2. Related Work

Constrained Foreground Object Search is the task of retrieving foreground objects that are compatible with the background image given the desired object type. Early works such as Photo Clip Art [11] retrieved foreground objects of a given class based on handcrafted features such as camera orientation, lighting, resolution and local context. More recently, Tan et al. [21] used off-the-shelf deep CNN features from the context to find suitable foreground persons particularly for person composition. Zhao et al. [30] used end-to-end feature learning to adapt to different object categories. In contrast, our approach has no constraint on what objects to retrieve and our experiments demonstrate it can retrieve compatible object candidates of different classes.

Predicting Compatibility. Prior work [31] has demonstrated it is possible to solve a related problem of predicting whether a composite and image are compatible. However, while [31] focuses on low-level compatibility (e.g., color, lighting, texture), we aim to stay largely agnostic to low-level properties (since properties such as lighting and color differences can be corrected in post-processing) and instead address semantic compatibility. Experiments show the advantage of our solution over [31] for the UFO search task.

Context-based Reasoning has been used in object recognition and detection [6]. Some works model the interaction of existing content in the image. For example, early works [3, 19] incorporated context cues for object recognition and Bell et al. [5] recently proposed a recurrent neural network for object detection. Our method more closely aligns with methods that make predictions about missing content based on image context. For example, one work proposes solving object detection based on context cues only [22]. Another work trains a standalone object-centric context representation to detect missing objects [20]. While these methods focus on the binary decision of whether there should be an object of a semantic class at a specific location, our approach addresses a distinct problem of searching for foreground object instances that are compatible with the context. Moreover, the compatible foreground objects may be a subset of a semantic class or come from different classes.

Scene Completion methods [7, 26, 32], like our work, involve inserting foreign content into an image. However, such methods address a distinct problem from our proposed UFO search problem. The former assumes the goal is to find a *patch* to insert into a scene image. Consequently, it must find a patch that seamlessly matches every background element in the scene. In contrast, UFO Search only finds a *compatible object*. This distinction provides an advantage over Scene Completion methods since UFO search methods can work in a general-purpose pipeline that positions a foreground object over the majority of the hole, and then applies any downstream post-processing methods (exemplified in Figure 1) to fill the gaps.

3. Methods

We propose a method for retrieving foreground objects from a database that are semantically compatible with a given image at a specified location. Our approach learns how to represent both the background image and each candidate foreground object in a shared search space that supports efficiently ranking the compatibility of all foreground objects. The architecture and training scheme for our approach are summarized in Figure 2 and described below.

3.1. Deep Learning Architecture

We propose a deep neural network that consists of two encoders which characterize the background image and foreground objects respectively by projecting them into a high-level feature space where compatible objects and image are near each other spatially. The approach is inspired by [30], though our architecture is more straight-forward and does not require an object label. The input to the foreground encoder is a foreground object on the background of mean image value, and the input to the background encoder is the background image with a hole[2] (needed for masking out the original object at that location in the training set) at the desired object location. The high-level feature outputs from the foreground objects can be stored in an index so that the objects can be retrieved given the feature corresponding to a background image.

Both encoders are derived from the popular VGG-19 [18] architecture (up to *fc6* layer), that takes as input images of size 224×224 and outputs 4096 dimensional feature embeddings. For the foreground object encoder, our goal is to capture the semantics of foreground objects. Since that is already captured well in the VGG-19 [18] architecture, we keep the weights that were pretrained for the ILSVRC-2014 competition [17] fixed during training. In contrast, for the background encoder, we initialize the weights with those pretrained for the ILSVRC-2014 competition [17] and then modify them during training. The encodings of the background image and foreground objects are then converted to

[2]The hole is filled with the mean image value.

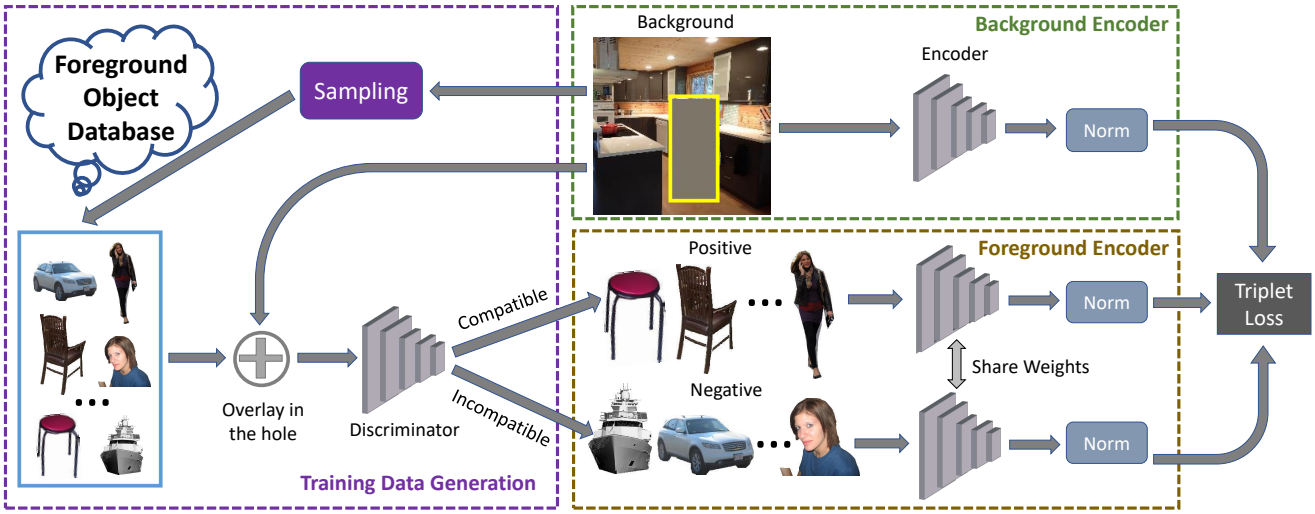

Figure 2. Architecture and training scheme of UFO search. Given a background image with a hole, we first sample foreground objects to overlay in the hole. Then the pretrained discriminator takes the overlaid image and identifies compatible and incompatible foreground objects. We use two encoders to encode the background image and foreground objects respectively. The triplet loss encourages the compatibility between the background and positive samples to be larger than its compatibility with negative samples.

unit feature vectors with ℓ_2 normalization and used to compute compatibility, by measuring their cosine similarity.

3.2. Loss Function

We adopt as our loss function a triplet loss [25] that takes as input a background image, positive sample, and negative sample. This function encourages the compatibility between a background image and a good foreground object (i.e., positive sample) to be larger than its compatibility with a bad foreground object (i.e., negative sample). Formally, given a background image I_b, positive sample I_f^p, and negative sample I_f^n, we want to enforce $C(I_b, I_f^p) > C(I_b, I_f^n)$. The triplet loss is a hinge loss $L(I_b, I_f^p, I_f^n) = \max(0, C(I_b, I_f^n) + M - C(I_b, I_f^p))$ where M is a positive margin to encourage a gap between the positive and negative sample. The training objective is to minimize the loss over all the sampled triplets.

3.3. Training Data Generation

We generate a training dataset that consists of triplets that contain a (1) background image, (2) compatible foreground object (positive), and (3) incompatible foreground object (negative). Exemplar triplets are shown in Figure 2.

Our key challenge lies in how to generate a sufficient number of positive samples per background image. That is because, for each background image, we only have one known positive sample: the foreground object that originally was there. Yet, for many scenes, numerous other foreground objects are plausible. We introduce two mechanisms for identifying a diversity of compatible foreground objects per background image: a discriminator to identify a noisy set of compatible foreground objects for each background image and a sampling module to accelerate identifying plausible foreground objects for training the encoder.

Training Data Filtering. We propose a discriminator to help filter the training data for effective training samples. We design it to take as input a given background image with the foreground object overlaid in the hole and output a prediction of whether they are compatible. Note that this discriminator is distinct from that employed for our UFO search encoder (described in Section 3.1). While our UFO search encoder learns how to represent the foreground objects and background image *de-coupled* in a complex, high-level feature space, the discriminator instead takes them *coupled* as input, with the foreground object overlaid on the background image. Consequently, while our UFO search encoder returns an efficient representation for search where objects that are compatible are close and objects that are not compatible are far away, the discriminator outputs a "yes"or "no"answer for a single pair of a foreground object and background image. We will show in Section 4 that the discriminator alone is unsuitable for solving our compatibility problem (in terms of accuracy and speed) but is valuable for boosting the performance of our UFO search encoder by generating noisy yet richer training triplets.

For the discriminator's architecture, we adapt VGG-19 [18] by replacing the last fully connected layer to produce a scalar value that indicates the compatibility score. To encourage the network to utilize high-level features so it focuses on semantic compatibility, we initialize with the weights pretrained for the ILSVRC-2014 competition [17] and freeze all the convolutional layers. We train all the

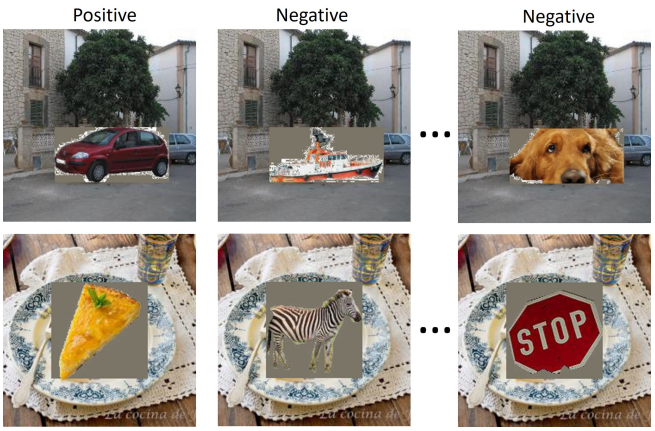

Figure 3. Examples of positive and negative samples used to train the discriminator for compatibility prediction. The positive samples (the left column) are created by overlaying the original object in the hole. The foreground objects in negative samples (the middle and right column) are randomly sampled from other images.

fully connected layers from scratch using a sigmoid cross-entropy loss. For training data, we generate compatible training examples by overlaying the original foreground object in the hole, and generate incompatible examples by selecting a random foreground object from another background image, resizing the object to fit in the hole, and then positioning it at the center of the hole. Examples of compatible and incompatible samples that we feed to train our discriminator are shown in Figure 3. Note that in the hole we overlay the object alone rather than the original patch containing the object. Otherwise the discriminator will simply learn to use low-level cues such as boundary continuity rather than semantics for classification.

We restrict training triplets to only include foreground objects that the discriminator confidently deems are (in)compatible when training the encoder. A foreground is deemed compatible with a given background if the discriminator predicts the compatibility score to be higher than a threshold t_{high} and incompatible if the score is lower than a threshold t_{low}. Despite training with a single ground truth object per background image, we show in the experiments that the discriminator can sufficiently rank the compatibility of diverse foreground objects. The success of training a classifier to rank has similarly been observed in prior work, e.g. Zhu. et.al [31] for the task of ranking the realism of image composites by low-level appearance.

Collecting Candidate Positive Examples Faster. While the discriminator solves an easier task than our UFO search method by solving a "yes" or "no" problem for a coupled input, it does so at the expense of efficiency. That is because naively applying the pretrained discriminator can require comparing each background image against almost every foreground object in a database before locating a sufficient number of high scoring compatible examples.

To speed up the discriminator's role in generating training data, we introduce two heuristics for sampling plausible foreground objects. First, we retrieve the top K_C most similar background scenes, and put the objects within those scenes into the sample set. The assumption is that similar backgrounds are likely to offer (possibly a diversity of) compatible objects. For example, for a given grass scene, we can find similar scenes such as a picnic on a lawn. The sitting persons or folding chairs in the picnic scene are also likely to be compatible with the grass scene. Second, we sample the top K_G foreground objects that are most similar to the original object, motivated by the assumption they are more likely to be compatible with the given context. For example, if a dog is running on the grass in the original image, it is likely that dogs in other scenes will also be compatible. In a database of over 60,000 objects, we observe a more than 20x speed up from the two proposed heuristics (from 731 to 32 random samples on average) to find another compatible object other than the original object in the hole.

3.4. Implementation

At training time, we employ the Adam solver [10] with fixed parameters $\beta_1 = 0.5$ and $\beta_2 = 0.999$. The initial learning rate is set to $l_r = 0.00001$ to train the encoder and $l_r = 0.00002$ to train the discriminator. We set the positive margin M, which encourages a gap between the positive and negative sample, to 0.3, with the threshold t_{high} for identifying positive samples in compatibility prediction set to 0.8 and the threshold t_{low} for identifying negative samples in compatibility prediction set to 0.3. All the background and foreground input images are set to a size of 224×224. We train the discriminator beforehand and then fix the discriminator when training the encoder. Training with PyTorch [14] takes 63 hours for 142,300 iterations on a single NVIDIA GeForce GTX 1080 Ti card.

At test time, we apply the background encoder to retrieve the most compatible foreground objects for a given background image with a hole. Compatible objects are found using nearest neighbor search between features describing the background image and foreground object. We speed up nearest neighbor search by using Faiss [9] to build an index for the evaluation set of foreground objects. After the speedup, it takes <0.1 seconds to retrieve top 25 compatible objects from a database of over 10,000 objects.

4. Experiments

We now examine the power of our UFO search approach in finding compatible foreground objects for a given hole in a background image. We examine the following questions: (1) How often do related baselines re-purposed for UFO search retrieve compatible foreground objects?, (2) How often does our UFO search method retrieve compat-

ible foreground objects?, and (3) How do our different design decisions contribute to the performance of our method? We conduct experiments on two datasets, with a quantitative evaluation in Section 4.1 and user evaluation in Section 4.2.

4.1. Quantitative Evaluation

We first conduct experiments using background images with holes that have various positions and sizes.

Dataset: Since large-scale datasets identifying all compatible foreground objects for a background image with a hole do not exist, we use as a proxy the image compositing dataset CAIS [30]. CAIS contains background images with a hole, an assigned category for the type of object that should fill the hole, and at least one compatible foreground object in that category. Although designed for constrained foreground object search, CAIS is also valuable for the more general problem of UFO search since most background images with holes unambiguously match only one object category from the eight foreground object categories represented[3]. The training set contains one compatible object for each of the 86,800 background images, using the original object in each image. The test set contains ∼16-140 compatible objects per image for 80 background images, with 10 background images for each object category.

Baselines: We compare our approach to four baselines:

Shape [30]: This adopts a naive strategy of ranking compatibility based on the extent to which the foreground object's aspect ratio (i.e., width/height) matches the hole's aspect ratio. For example, a tall hole would match a tree or pedestrian better than a car.

RealismCNN [31]: It uses a discriminator to predict the realism of image composites in terms of low-level cues such as color, lighting, and texture compatibility. After overlaying each foreground object into the hole (as in Figure 3), the pretrained model ranks the compatibility of all objects.

Two constrained search methods: Since *constrained* search methods require a category as input and so are not directly useful, we examine two ways to adapt them for an *unconstrained* setting. First, we train a classifier to decide which category to fill in the hole[4] and then apply a constrained search method to retrieve suitable instances within that category. We call this *Constrained Foreground Object Search - Classifier (CFO-C Search)*. Note that it has the limitation that it requires collecting class labels to train the classifier and so would not recover from the errors of the classifier. The second approach retrieves the top 100 objects for each of the eight categories using the constrained search method and then applies our trained discriminator to rank the retrieved 800 (100x8) objects. We call this *Constrained Foreground Object Search - Discriminator (CFO-D Search)*. Note that *CFO-D Search* becomes less practical with more categories and more retrievals, because it requires expensively traversing every retrieval with the discriminator and ranking all the retrievals. We evaluate both approaches using the constrained search algorithm [30].

Ablated Variants: We evaluate ablations of our *UFO Search* to assess the influence of different design decisions:

- *No BG Training*: It uses the pretrained weights for the ILSVRC-2014 competition [17] as the background encoder's weights. This is valuable for assessing the benefit of training the background encoder when training UFO search.
- *No Discriminator*: It does not use our training data generation scheme, described in Section 3.3. Instead, it uses one compatible foreground object per background image (i.e., the original one in the hole) and many incompatible samples (i.e., all foreground objects in other background images).
- *Discriminator Only*: The discriminator described in Section 3.3, which we use for training data generation, is instead used to predict compatibility at test time. Recall that a limitation of this approach is that it requires overlaying each foreground object in the hole of each test background image, which is very computationally expensive.
- *Regression*: This approach matches the *No Discriminator* approach except that it trains for the regression problem (i.e., using *Mean Square Error (MSE)*) instead of the ranking problem (i.e., using the triplet loss). To do so, it regresses to the feature of the original foreground object in the hole from the background image using the MSE loss function. We evaluate on a simplified situation (without the discriminator) to assess the training approach on its own.

Evaluation Metrics: We use *mean Average Precision* (mAP) for evaluation, which is a common metric in image retrieval. We report mAP for each category as well as overall, by averaging over all category mAPs. To make our findings compatible with the constrained foreground object search methods (*CFO-C Search* and *CFO-D Search*), we evaluate the mAP for the top 100 retrievals. This is because CFO methods do not rank all objects in all categories. We share the mAP results with respect to all the retrievals for all other methods in the Supplementary Materials.

Overall Results: Results are shown in Table 1.

Overall, our *UFO Search* method outperforms the four related baselines: *Shape*, *RealismCNN* [31], *CFO-C*, and *CFO-C*. For example, mAP is **32.17%** for *UFO Search*, which is over 24 percentage points better than for *Shape* and *RealismCNN*. These results reveal that relying on hole shape alone or low-level compatibility alone is not very informative, and demonstrates the advantage of addressing semantic compatibility directly. *UFO Search* also results in a 1.49 percentage point improvement over the next best constrained search baseline. This shows that *UFO Search* not only offers a scalable end-to-end solution that avoids requiring a separate class predictor (required by *CFO-C*) or large

[3] boat, bottle, car, chair, dog, wall painting, person, and plant

[4] The classifier employs the VGG architecture with weights pretrained on ImageNet, and achieves overall accuracy of 63.75%.

Method	boat	bottle	car	chair	dog	painting	person	plant	overall
Shape [30]	7.47	1.16	10.40	12.25	12.22	3.89	6.37	8.82	7.82
RealismCNN [31]	12.33	7.19	7.55	1.81	7.58	6.45	1.47	12.74	7.14
CFO-C Search [30]	57.48	14.24	18.85	21.61	38.01	**27.72**	**47.33**	20.20	30.68
CFO-D Search [30]	55.48	8.93	24.10	18.16	**57.82**	21.59	27.66	23.13	29.61
Ours: UFO Search	**59.73**	**21.12**	**36.63**	19.27	36.51	25.84	27.11	**31.19**	**32.17**
Ours: No BG Training	49.09	0.62	3.23	9.01	7.37	11.66	7.30	22.02	13.79
Ours: No Discriminator	58.07	17.22	20.71	**21.93**	37.05	24.57	27.11	25.05	28.97
Ours: Discriminator Only	48.71	8.35	21.42	17.32	50.61	20.28	22.14	17.35	25.77
Ours: Regression	55.33	9.90	18.31	17.42	27.79	23.76	35.66	10.83	24.87

Table 1. *Mean Average Precision* for the top 100 retrievals of four baselines, our UFO search method, and its four ablated variants.

computational costs (required by *CFO-D* as more categories and retrievals are considered), but also yields improved prediction accuracy. This highlights a benefit of directly learning to solve the *unconstrained* search problem rather than modifying *constrained* search methods.

Our analysis also shows how our *UFO Search* compares to the baselines for different object categories. As shown in Table 1, our *UFO Search* outperforms all baselines on the following four object categories: boat, bottle, car, and plant. The top-performer for the other four categories is shared between three baselines. One reason our *UFO Search* performs poorer at times is that for the *person* category it can mistakenly retrieve boats for surfing scenes and dogs for park scenes. Our *UFO Search* also at times mistakenly retrieve boats and chairs for the *painting* category. Additionally, for the *dog* category, it at times mistakenly retrieve cars and persons for street scenes. These findings suggest that our approach understands the context semantically, but does not always capture well the potential interaction between the inserted object and the context for specific categories such as *person*, *painting*, and *dog*. We hypothesize discriminator based methods (*CFO-D Search* and *Ours: Discriminator Only*) can perform better than our *UFO Search* method for the *dog* category because it can be easier to recognize which foreground objects are incompatible for the hole's size and shape when overlaying the foreground object directly in the hole (as the discriminator methods do).

UFO Design Analysis Results: Results in Table 1 also illustrate the benefit of design choices for our *UFO Search*.

The poor performance from the *Discriminator Only* demonstrates that the triplets sampled by the discriminator are imperfect; i.e., mAP score is 25.77%. Moreover, it performs worse than our *UFO Search* both in terms of accuracy (i.e., mAP score is 6.4 percentage points worse) and speed (i.e., we observe over a 3000x slow down from 0.1 to 365.6 seconds when relying on the discriminator instead of *UFO Search* to perform retrieval from a database of 60,000 objects). These findings highlight a strong advantage of learning how to represent the background image and foreground objects de-coupled in a complex, high-level feature space that supports efficient search, as our *UFO Search* does, rather than coupling the background image and foreground object as input, as the discriminator requires.

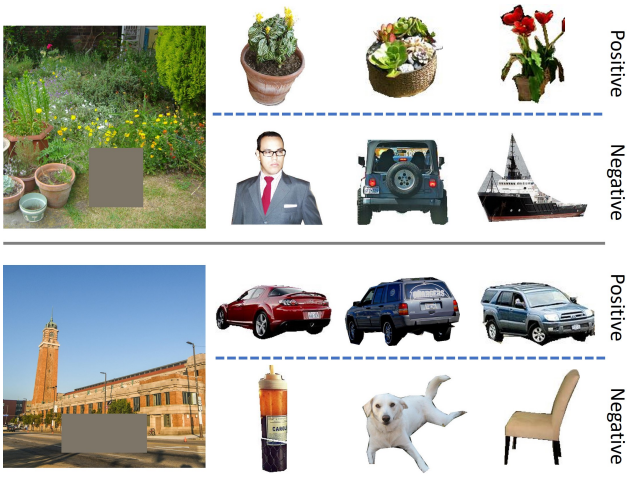

Figure 4. Positive and negative samples that are deemed compatible and incompatible with the background image by our discriminator. As shown, it can identify multiple compatible objects.

The gain of *UFO Search* over *No Discriminator* demonstrates the advantage of employing our training data generation method; i.e., we observe more than a three percentage point boost. The encoder in *UFO Search* benefits from learning using the noisy training data sampled by the discriminator. Figure 4 exemplifies that the discriminator can identify multiple compatible objects, despite having trained with a single positive ground truth per background image.

The weaker performance of *Regression* versus *No Discriminator* illustrates the advantage of training for the ranking problem (by using the triplet loss); i.e., *No Discriminator* yields more than a four percentage point boost over *Regression*. We attribute this performance gain to a benefit of training with both positive and negative samples in the ranking problem, rather than only positive samples when training for regression. Training with both positive and negative samples better shapes the feature space by pushing compatible objects closer to the background encoding and incompatible objects farther away from the background encoding.

Finally, the poor performance of *No BG Training* compared to the three methods that train the background encoder (i.e., *UFO Search*, *No Discriminator*, *Regression*) demonstrates the benefit of feature learning.

4.2. User Evaluation

We next conduct user evaluation on a more diverse dataset consisting of 79 object categories.

Dataset: We employ MS-COCO [13] to create a diverse dataset of 79 foreground object categories. We use the object segmentation mask annotations to decompose each image into a background scene and foreground objects (see Supplementary Materials for more details). This yields 14,350 background images and 61,069 foreground objects. We use 14,230 background images for training and the remaining 120 for evaluation. To provide a large foreground object database at test time, we use all 61,069 foreground objects in both training and testing. This is acceptable since we do not learn the feature space for foreground objects. In order to evaluate the effectiveness in encoding the context exclusively, we fix the hole size and position for all background images. Specifically, we create the holes for each background scene by removing a square that bounds each foreground object. Then we resize each background image to 224×224 with a hole of size 112×112 in the center.

Ablated Variants: We compare with our *UFO Search* the four ablated variants described in Section 4.1: *No BG Training*, *Regression*, *No Discriminator*, and *Discriminator Only*.

Evaluation Metrics: Since this dataset does not identify multiple compatible foreground objects per background image, we conduct a user study to measure the Precision@K (P@K), which is the percentage of compatible foreground objects in the top K retrievals. We show users a background image and K candidate foreground objects retrieved by an image search approach. Users are asked to select the foreground objects that are not compatible with the background image. Each background image is evaluated by 3 different users. If any user labels a foreground object as incompatible, the foreground object is considered to be incompatible.

Method	P@5	P@10	P@15	P@20	P@25
No Training	12.67	13.33	13.28	12.50	12.50
Regression	30.33	30.75	30.39	30.50	30.40
No D	38.50	36.58	36.11	35.54	35.57
D Only	36.33	37.25	36.00	35.46	35.77
UFO Search	**41.83**	**40.33**	**39.39**	**38.96**	**38.83**

Table 2. User study results showing the percentage of retrieved foreground objects in the top K retrievals that are deemed compatible by users. *No D = No Discriminator*, *No Training = No BG Training*, *D Only = Discriminator Only*.

Overall Results: Quantitative results are shown in Table 2 for using our *UFO Search* for the top 5, 10, 15, 20, and 25 retrievals respectively. Qualitative results are shown in Figure 5. The top two examples illustrate that our *UFO Search* can retrieve only one type of object when only one object type is compatible; specifically, it retrieves only frisbees and catchers for the dog and baseball field respectively. Also shown is that our *UFO Search* can retrieve compatible objects that are from different categories when numerous object types are appropriate for the scene. Specifically, our approach retrieves carrots, oranges, bananas, cakes, sandwiches and hot dogs for a hole on a plate on a table (second to bottom example) and retrieve horses, motorbikes, cars and cows for the hole in the grass (bottom example).

UFO Design Analysis Results: Our *UFO Search* method outperforms all its ablated variants for every retrieval size, increasing the search precision by **3.33**, **3.08**, **3.28**, **3.42**, and **3.06** percentage points compared to the next best ablated variant in top 5, 10, 15, 20, 25 retrievals respectively. This aligns with and reinforces our findings in Section 4.1.

Qualitative comparisons in Figure 5 illustrate strengths of our design choices. For the first example, while the top retrievals of *UFO Search* are all compatible frisbees, the *No Discriminator* retrieves umbrellas which have similar shape to frisbees but are not compatible in the context. For the second baseball field example, all ablated variants accurately retrieve the person category, however only our *UFO Search* method recognizes that a catcher is the only suitable activity for the context. The last example shows the retrievals of *Discriminator Only* can be noisy, containing incompatible objects such as a toaster, but also is effective in retrieving compatible objects of multiple categories, such as horses, bikes and cars. We attribute this diversity of categories from the discriminator as a core reason why the encoder of our *UFO Search* method is able to learn to retrieve compatible objects from multiple categories, as shown in the *UFO* retrieval of the last example. The discriminator can effectively generate triplets for training the encoder. In contrast, *No Discriminator* only retrieves cars although multiple object types are appropriate for the scene. Further analysis of the retrieval diversity is in the Supplementary Materials.

5. Conclusion

We introduce a novel problem of searching for compatible foreground objects to edit images without constraints on object types. We also propose a solution with an efficient, scalable approach for generating a large training dataset. Experiments demonstrate advantages of our approach for efficiently and accurately retrieving compatible foreground objects from large-scale, diverse datasets. We offer this work to support people in efficiently editing their images.

Acknowledgments. We thank the anonymous reviewers for their feedback and crowd workers for their contributions. This work is supported in part by gifts from Adobe.

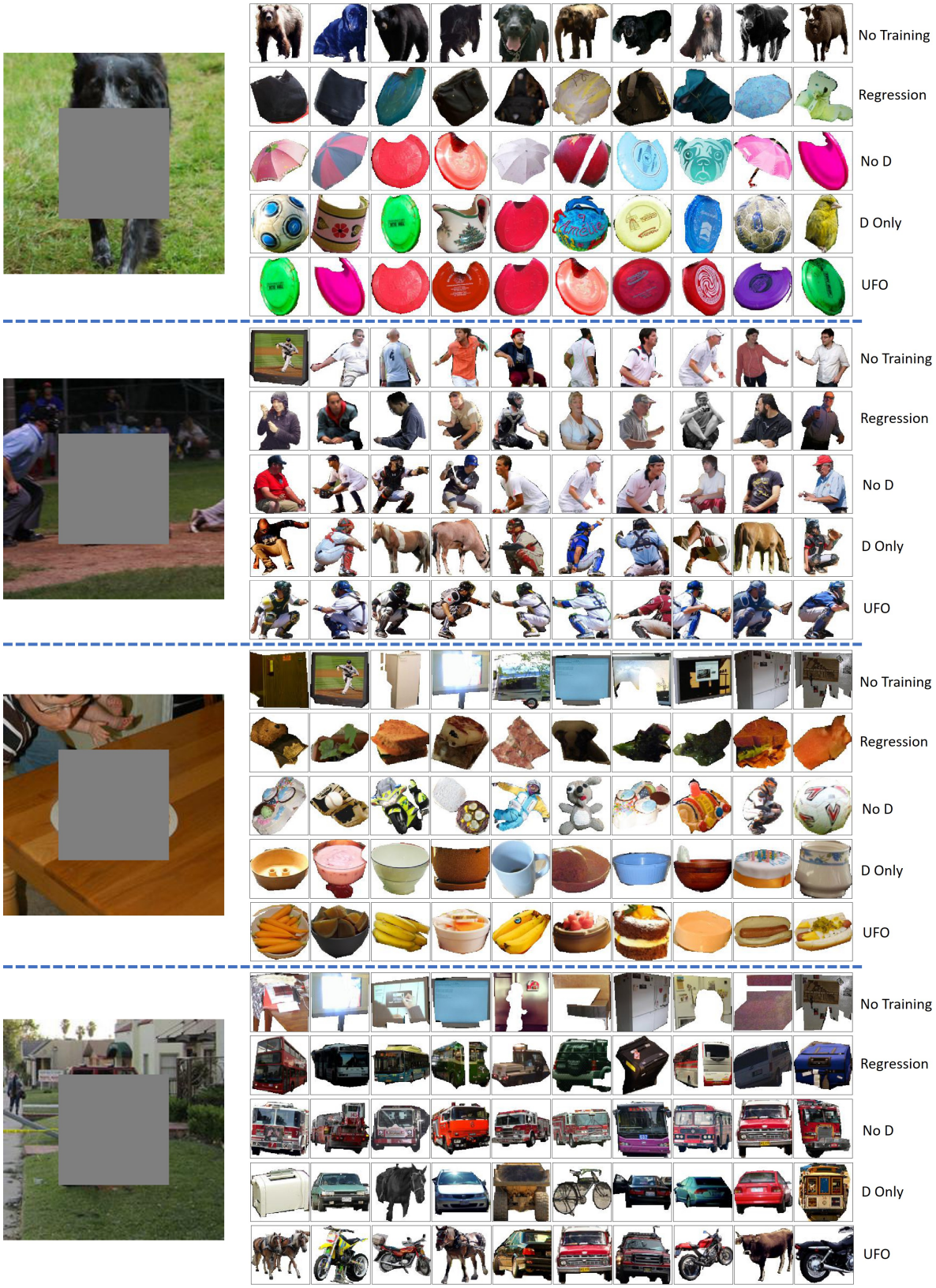

Figure 5. Qualitative results on MS-COCO [13]. For the top two examples, our approach retrieves objects from the only object type from MS-COCO (frisbee and catcher, respectively) that is really compatible with the context. The bottom two examples demonstrate that our approach has the potential to retrieve compatible objects of different categories when many object types are appropriate for the scene.

References

[1] Kenan E Ak, Ashraf A Kassim, Joo Hwee Lim, and Jo Yew Tham. Learning attribute representations with localization for flexible fashion search. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7708–7717, 2018. 1

[2] Samaneh Azadi, Deepak Pathak, Sayna Ebrahimi, and Trevor Darrell. Compositional gan: Learning conditional image composition. *arXiv preprint arXiv:1807.07560*, 2018. 1

[3] Moshe Bar and Shimon Ullman. Spatial context in recognition. *Perception*, 25(3):343–352, 1996. 2

[4] 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. 1

[5] Sean Bell, C Lawrence Zitnick, Kavita Bala, and Ross Girshick. Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2874–2883, 2016. 2

[6] Santosh K Divvala, Derek Hoiem, James H Hays, Alexei A Efros, and Martial Hebert. An empirical study of context in object detection. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1271–1278. IEEE, 2009. 2

[7] James Hays and Alexei A Efros. Scene completion using millions of photographs. In *ACM Transactions on Graphics (TOG)*, volume 26, page 4. ACM, 2007. 1, 2

[8] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Globally and locally consistent image completion. *ACM Transactions on Graphics (ToG)*, 36(4):107, 2017. 1

[9] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. *arXiv preprint arXiv:1702.08734*, 2017. 4

[10] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 4

[11] Jean-François Lalonde, Derek Hoiem, Alexei A Efros, Carsten Rother, John Winn, and Antonio Criminisi. Photo clip art. *ACM transactions on graphics (TOG)*, 26(3):3, 2007. 1, 2

[12] Chen-Hsuan Lin, Ersin Yumer, Oliver Wang, Eli Shechtman, and Simon Lucey. St-gan: Spatial transformer generative adversarial networks for image compositing. 1

[13] 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. 7, 8

[14] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017. 4

[15] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2536–2544, 2016. 1

[16] Jose A Rodriguez-Serrano, Diane Larlus, and Zhenwen Dai. Data-driven detection of prominent objects. *IEEE transactions on pattern analysis and machine intelligence*, 38(10):1969–1982, 2016. 1

[17] 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. 2, 3, 5

[18] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 2, 3

[19] Thomas M Strat and Martin A Fischler. Context-based vision: recognizing objects using information from both 2 d and 3 d imagery. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(10):1050–1065, 1991. 2

[20] Jin Sun and David W Jacobs. Seeing what is not there: Learning context to determine where objects are missing. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1234–1242. IEEE, 2017. 2

[21] Fuwen Tan, Crispin Bernier, Benjamin Cohen, Vicente Ordonez, and Connelly Barnes. Where and who? automatic semantic-aware person composition. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1519–1528. IEEE, 2018. 2

[22] Antonio Torralba. Contextual priming for object detection. *International journal of computer vision*, 53(2):169–191, 2003. 2

[23] Antonio Torralba, Rob Fergus, and William T Freeman. 80 million tiny images: A large data set for nonparametric object and scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 30(11):1958–1970, 2008. 1

[24] 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. 1

[25] Xiaolong Wang and Abhinav Gupta. Unsupervised learning of visual representations using videos. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2794–2802, 2015. 3

[26] Oliver Whyte, Josef Sivic, and Andrew Zisserman. Get out of my picture! internet-based inpainting. In *BMVC*, volume 2, page 5, 2009. 2

[27] Chao Yang, Xin Lu, Zhe Lin, Eli Shechtman, Oliver Wang, and Hao Li. High-resolution image inpainting using multi-scale neural patch synthesis. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, page 3, 2017. 1

[28] Jimei Yang, Brian Price, Scott Cohen, and Ming-Hsuan Yang. Context driven scene parsing with attention to rare classes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3294–3301, 2014. 1

[29] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. *arXiv preprint*, 2018. 1

[30] 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, 5, 6

[31] 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. 1, 2, 4, 5, 6

[32] Zhe Zhu, Hao-Zhi Huang, Zhi-Peng Tan, Kun Xu, and Shi-Min Hu. Faithful completion of images of scenic landmarks using internet images. *IEEE transactions on visualization and computer graphics*, 22(8):1945–1958, 2015. 2