Fine-grained Foreground Retrieval via Teacher-Student Learning

Zongze Wu
Hebrew University
zongze.wu@mail.huji.ac.il

Dani Lischinski
Hebrew University
danix@mail.huji.ac.il

Eli Shechtman
Adobe Research
elishe@adobe.com

Abstract

Foreground image retrieval is a challenging computer vision task. Given a background scene image with a bounding box indicating a target location, the goal is to retrieve a set of images of foreground objects from a given category, which are semantically compatible with the background. We formulate foreground retrieval as a self-supervised domain adaptation task, where the source domain consists of foreground images and the target domain of background images. Specifically, given pretrained object feature extraction networks that serve as teachers, we train a student network to infer compatible foreground features from background images. Thus, foregrounds and backgrounds are effectively mapped into a common feature space, enabling retrieval of the foregrounds that are closest to the target background in that space. A notable feature of our approach is that our training strategy does not require instance segmentation, unlike current state-of-the-art methods. Thus, our method may be applied to diverse foreground categories and background scene types and enables us to retrieve the foreground in a fine-grained manner, which is closer to the requirements of real world applications.

1. Introduction

Foreground retrieval is the task of finding an image of a suitable foreground object, for a given background scene. Specifically, given a background image, a bounding box that indicates the location of a foreground object, and its category, the goal is to retrieve a set of images of foreground objects, compatible with the background. In this task, the focus is on retrieving a suitable foreground, rather than on seamless compositing. Foreground retrieval can greatly benefit a number of real-world applications. For graphics design, manually searching for a suitable foreground object is time-consuming, while an automatic foreground retrieval system could enlarge the search space and suggest a gallery of suitable candidates in real time. For interior design, foreground retrieval could suggest suitable pieces of furniture that best fit a user-provided image of a room.

Existing literature focuses on the general, *coarse-grained* foreground retrieval setting, where the searching space is very broad (different types of background scenes, and foreground objects). In this setting, the focus is on retrieval of foreground objects that are semantically compatible with the background, but not necessarily on matching finer level features. For example, given a dining room scene, existing methods [28, 29] are more likely to retrieve a chair than a bed, but might not distinguish between different chair poses or colors. Simply restricting the searching space during training and inference fails to achieve good results (Section 2.1). This limits the ability of these methods to cope with the foreground retrieval task within a specific domain, which is the typical real-world scenario. For example, when a user wants to see what his/her living room would look like with a new sofa, the retrieved foreground images should be of sofas with a compatible style and pose. Therefore, we focus instead on a *fine-grained* foreground retrieval task, meaning that the background scenes come from a well-defined category, such as room interiors, and the library of foreground images consists of semantically relevant objects, such as various kinds of furniture. The goal is then to decide which of the foreground images constitute a good fit for the input background scene at the designated location, and rank the retrieved candidates.

In the fine-grained setting described above, all of the candidate foregrounds are semantically relevant, and thus their fitness level for a given background mainly depends on a variety of fine-level features, rather than high-level semantics. Furthermore, the importance of each feature varies depending on the purpose of the retrieval. For creating a realistic composite, a compatible viewpoint might be more important than for a shopping application. Thus, we are interested in designing an approach capable of considering these finer-level features, and adjusting their relative weights, depending on the target application.

To the best of our knowledge, there is no large scale annotated dataset that can be used to train a fine-grained foreground retrieval network directly. Constructing such a dataset is a formidable task, as it would require assigning a fitness annotation to a quadratically growing number of background and foreground pairs. Furthermore, fitness is subjective, and as pointed out earlier, may be judged differently with a different application in mind.

On the other hand, extraction of features for classical computer vision tasks has been studied for many years, resulting in several very large scale datasets and highly successful trained models. For example, CNNs, such as VGG [22] and Resnet [7] excel at image classification, while Faster-RCNN [19] and Yolo [18] excel at object detection. MarrNet [27, 24] can extract 3D structure and viewpoint from images of furniture, and DeepFashion [15, 4] is able to identify the category, key points and style of clothes.

Our premise in this work is that given sufficient background context, it is possible to extract the features necessary for predicting the compatibility of a candidate foreground object. Consider, for example, an image of a bedroom, from which the bed has been removed. A human observer is easily able to predict the category and the pose of the missing item, and in many cases its color or style, as well. Similarly, a model trained to extract a set of relevant foreground features from a background should prove effective for foreground retrieval.

Therefore, we formulate foreground retrieval as a self-supervised domain adaptation task, where the source domain consists of images of foreground objects, while the target domain consists of background scene images, and solve it using teacher-student learning. Specifically, we use pretrained networks that extract features from foreground images as teachers, and train a student network to retrieve such features from the background images.

The above approach is generic and applicable to different fine-grained foreground retrieval scenarios, such as interior design, landscape architecture, and urban design. It offers users the flexibility to decide which features are relevant and assign them different weights. The approach only requires two types of pretrained networks: a detection network for preparing the training data and a task-related feature extraction teacher networks for the foreground. In our experiments, we focus on retrieval of furniture for indoor scenes and use Faster-RCNN [19] for detection. As the teacher networks, we use MarrNet [27, 24] to extract furniture category, viewpoint, and shape, and VGG [22] to extract style features. Using a new fine-grained foreground retrieval training and evaluation dataset, we show that our method achieves much better results compared to the current state-of-the-art.

2. Related Work

2.1. Foreground Image Search

Lalonde et al. [12] were the first to pose the problem of inserting a new object into a photograph as a context-sensitive object retrieval task. Given an image of the background and a location, a large library is searched for objects of the desired class that match the surrounding background in terms of camera pose, lighting, resolution, etc. Their approach makes use of a set of heuristic functions to estimate the desired attributes, and requires a rough 3D representation of the background scene, as well as relevant annotations for the objects in the library.

The power of deep neural networks has enabled approaches with more modest requirements. Tan et al. [25] focus on the human instance composition task. Their approach predicts suitable locations for adding humans into an image, and retrieves suitable images of humans to insert in these locations. The retrieval is based on matching the local context of the intended location to those of the candidate humans. The matching is done using deep feature representations extracted by a pre-trained network.

Zhao et al. [28] propose a self-supervised learning system that utilizes triplet loss [20] to select suitable candidates. They use instance segmentation to crop out foreground objects, and construct background images by masking the foreground objects' bounding boxes. Using background images and pairs of foreground images, they train a triplet network to map the backgrounds and the foregrounds to a common space, where the embedding of the "positive" (original) foreground is closer to that of the background than the embedding of the "negative" (non-original) foreground. Once trained, this network is used to rank candidate foregrounds by the distance from their embedding to that of the background.

The above approach has the advantage of using a self-supervised learning framework, where the positive and the negative examples are generated automatically. However, it requires accurate instance segmentation of the foreground objects. This is a crucial requirement, as without accurate foreground masks, the network might simply learn to match parts of the background surrounding the foreground. State-of-the-art instance segmentation methods, such as Mask-RCNN [6] are not able to produce sufficiently accurate masks for images in the wild, and tend to miss parts of the foreground, while including parts of the background. Although there are a few datasets with instance segmentation, for foreground categories such as furniture, many of the images have a narrow field-of-view, depicting foreground objects without sufficient surrounding background context.

Another disadvantage of their triplet-based training is that it only uses the original foregrounds as the positive examples, while all other foregrounds are considered negative. This is a very restrictive assumption, since in practice the original foreground can often be replaced with one from another background, without harming realism. Zhao et al. [28] thus must resort to heuristic methods for increasing the number of positive examples.

Rather than using handcrafted heuristics, Zhao et al. [29]

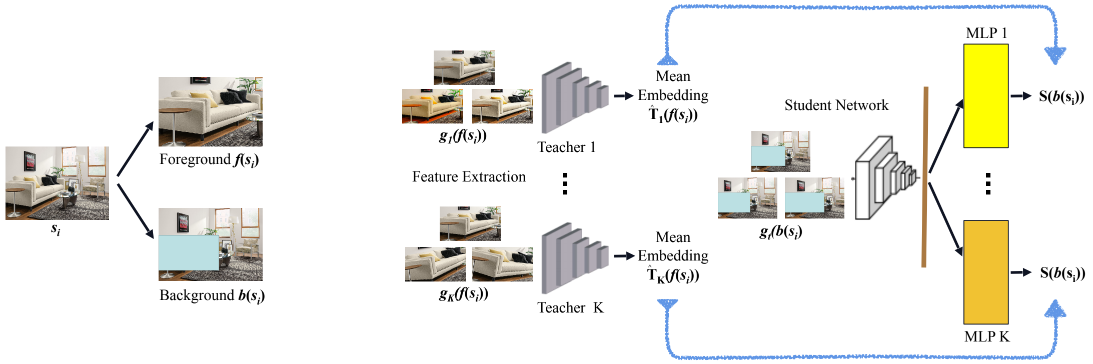

Figure 1: A high-level diagram depicting our approach. Left: A pretrained object detection network is used to identify foreground objects in scene images s_i, resulting in compatible background-foreground pairs $(f(s_i), b(s_i))$. Middle: Average task-related features $T_j(f(s_i))$, $j \in 1, \dots, K$, are extracted from an augmented set of foreground images $g_j(f(s_i))$ using pretrained teacher networks T_j. Right: A student network S is trained to predict the foreground features $T_j(f(s_i))$, conditioned on an augmented set of masked backgrounds $g_t(b(s_i))$.

first train a discriminator to assess the compatibility of background-foreground pairs, and then use this discriminator to generate positive and negative examples for training a two-branch embedding network using triple loss. Since the foreground embedding branch of their network is pretrained, their method may also be considered as a form of teacher-student learning. It should be noted that Zhao et al. [29] apply their approach to the unconstrained foreground object (UFO) search task, where the method selects a suitable object category on its own, rather than getting the category as input. However, both the constrained and the unconstrained foreground retrieval variants of their method still rely on instance segmentation to obtain accurate foreground masks for the training.

2.2. Teacher-Student Learning for Unsupervised Domain Adaptation

Li et al. [13] propose the idea of teacher-student (T/S) learning for unsupervised domain adaptation in the context of speech recognition. Given a well-trained "teacher" model for a source domain, a "student" model can be trained for a target domain, provided that parallel (i.e., paired) unlabeled data is available. For example, a model pretrained to recognize the speech of an adult may be used to train a model for recognizing the speech of a child, provided synchronized untranscribed sequences of both speakers, by minimizing the KL-divergence between the outputs of the two models. The T/S learning approach was later improved by Meng et al. [16], who use adversarial training to ensure that the deep features extracted by the student network, i.e., the latent embedding, is domain invariant.

The applicability of T/S learning is restricted by the strong requirement for paired data. However, in our setting, a natural pairing of data arises by extracting the background and the foreground components from the same image. Thus, treating the background and the foreground images as two domains, T/S learning may be used to learn a joint embedding and use it to retrieve a suitable foreground for a given background image.

3. Proposed Approach

As explained earlier, our premise in this work is that the compatibility of a foreground object to a background scene may be assessed by inferring a set of task-related foreground features from the surrounding context in the background image. Thus, suitable foreground images may be retrieved from a library by comparing their features to those inferred from the input background. Furthermore, the models for inferring these features may be trained using teacher-student learning, where the teachers are models pretrained for extracting the desired features from images of foreground objects.

Our approach is depicted in Fig. 1. Formally, given an image of a scene s_i, let $f(s_i)$ denote the region of the image, occupied by a foreground object. For example, if s_i is an image of a living room, $f(s_i)$ might be a rectangular region containing a sofa. The remainder of the image contains the background context, which we denote as $b(s_i)$.

Let $\{e_{i1}, \dots, e_{iK}\}$ denote a set of features, relevant to the foreground retrieval task, which may be extracted from the foreground image using a set of pretrained teacher networks T_j, i.e.:

$$e_{ij} = T_j(f(s_i)) \quad \text{for } j = 1, \dots, K. \tag{1}$$

Our goal is to train a set of student networks S_j that are able to extract the same features from the corresponding background images, i.e.:

$$e_{ij} = S_j(b(s_i)) \quad \text{for } j = 1, \dots, K, \forall s_i. \tag{2}$$

In other words, the student networks S_j learn to compute the same feature embedding for background images $b(s_i)$ as the pretrained teacher networks do for the corresponding foreground objects $f(s_i)$.

Having trained the student networks, given a background $b(s_i)$, we can retrieve a suitable foreground $f(s_k)$ by matching the feature embeddings:

$$k^* = \arg\min_k \sum_j \vec{w}_j D_j(S_j(b(s_i)), T_j(f(s_k))). \tag{3}$$

Here, D_j is a distance metric for the j-th feature, whose exact form depends on the feature at hand. For example, distance between viewpoints may be defined using the Geometric Structure Aware Loss [23], while abstract deep features may be compared using the cosine distance between their latent space vectors. The distances are weighted using weights $\vec{w}_j$, which are determined based on the purpose of the retrieval task, or set automatically to default values, as described in Section 3.2.

In practice, we train a single student network S to predict the output of all teacher networks simultaneously, i.e.:

$$S(b(s_i)) = \{T_j(f(s_i)) | j = 1, \ldots, K\}, \forall s_i, \tag{4}$$

since multiple task learning usually boosts the performance of each sub task, as was indeed observed in our experiments.

3.1. Robust Feature Extraction

We use augmentation in order to make our approach less sensitive to small changes in the input. Specifically, in order to extract more consistent features, during the training process, we apply to each foreground $f(s_i)$ a set of augmentations that are not supposed to affect the feature that each teacher network is supposed to extract. For example, changing object color or lighting should not affect the extracted orientation, while rotation or translation of the full image should not affect the object style. Thus, to extract consistent features we take the mean feature extracted from a set of such augmentations:

$$\hat{T}_k(f(s_i)) = \frac{1}{n} \sum_{l=1}^{n} T_k(g_k^l(f(s_i))) \tag{5}$$

Here n is total number of augmentation per image ($n = 15$ in all our experiments). A similar set of augmentations is performed on each input to the student network.

3.2. Automatic Default Weights

While our method allows users to adjust the weights of different features according to the intended purpose of retrieval, a good set of default weights enables the method to function even without requiring the user to fine-tune them.

Let $d_{i,k}^j = D_j(S_j(b(s_i)), T_j(f(s_k)))$ denote the distance between the j-th feature of the background $b(s_i)$ and the foreground $f(s_k)$. The total distance between $b(s_i)$ and $f(s_k)$ is a linear combination of the feature distances:

$$d_{i,k} = \sum_j \vec{w}_j d_{i,k}^j \tag{6}$$

where $\vec{w}_j$ is the weight of feature j with $\vec{w}_j > 0, \sum_j \vec{w}_j = 1$. We seek a set of weights $\vec{w}_j$ that make the original foreground $f(s_i)$ to be closer to $b(s_i)$ than any other foreground ($d_{i,i} < d_{i,k}$), by minimizing a hinge loss

$$L(i,k) = \max(0, \sum_j \vec{w}_j (d_{i,i}^j - d_{i,k}^j) + m), \tag{7}$$

where m is a positive margin to encourage a gap between the positive and negative sample. This optimization problem can be solved by a single fully-connected layer, with a softmax activation applied to its weights $\vec{w}$, to ensure that $\vec{w}$ is a transition vector ($\vec{w}_j > 0, \sum_j \vec{w}_j = 1$), as in [5]. Since a single layer network may suffer from bad initialization [2], in practice we use a linear network with e layers and a softmax activation on its weights, and the weights $\vec{w}$ are obtained as the product of the learned layer weights

$$\vec{w} = w M_{e-1} M_{e-2} \ldots M_2 M_1 \tag{8}$$

where M_i are the transition matrices of the first $e-1$ layers, and w is the transition vector of the last layer.

We use $e = 3$, $m = 0.1$ and train one network per object category. For each scene s_i, we randomly pick 1000 foregrounds $f(s_k)$ from the same category as negative examples and calculate $d_{i,k}^j$. To make sure the distance of each feature d^j are in similar scale, we normalize their population mean to 0 and variance to 1. We perform hard example mining by removing easy pairs, where the candidate k is worse than the original foreground i for all features j ($d_{i,i}^j - d_{i,k}^j < 0 \ \ \forall j$), or where the candidate is better than the original foreground for all features ($d_{i,i}^j - d_{i,k}^j > 0 \ \ \forall j$).

4. Dataset and Training

To the best of our knowledge, there is no dataset that contains enough semantically similar objects with bounding box annotations, while including a significant amount of surrounding background context (wide field of view). Thus, we constructed our own dataset. Below we describe our dataset construction procedure, feature extraction using the teacher networks, and the training of the student network.

4.1. Dataset construction

Scene images. We harvested a variety of indoor scene images from the web, conditioned on keywords, such as ‘bedroom’, ‘dining room’, ‘interior’, ‘living room’ and ‘office’. Low resolution images (under 300 × 300) were filtered out. To further filter out irrelevant images, we used VGG [22] pretrained on Place365 [30] to infer the scene type. Only images whose predicted scene category are among the top 30 categories were retained. The breakdown by category is reported in the supplementary material.

To extract foreground object images from these scene images, we use Faster-RCNN [19] trained on OpenImage data set [11] to generate bounding boxes for candidate foreground objects. We filter out boxes that do not belong to 'furniture' subcategory, as well as ones that are too large (width or height is larger than 0.6 of the image dimensions). Each remaining box is extended to a square shape, and used to crop a foreground image, which we resize to 256 × 256. Background images are resized to 256 × 192 by blur mirror padding [9]. In total, we obtain 16.7K unique scene images, and 31K foreground images. The dataset is split into training (85%), testing (10%) and validation (5%) sets. Horizontal flipping is used to augment the data.

Additional foreground images. To further enrich our foreground image set, we harvest additional images via Google image search using the keywords 'bed', 'bookcase', 'chair', 'desk', 'nightstand', 'sofa', 'table', and 'wardrobe'. Most of these images have a plain (white) background. We use the same object detection network as before to obtain the category label and filter out images where no furniture is detected, or containing multiple furniture items. Images are resized to 256 × 256 with zero padding. In total, we obtain 40K additional images. Horizontal flipping is used to augment the data. More statistics about the dataset can be found in the supplementary.

4.2. Foreground Feature Extraction

In our approach, the pretrained foreground feature extraction networks (teacher networks) may be chosen depending on the task. In this work we focus on retrieval of furniture for interior scenes, and we choose MarrNet [27, 24] to extract viewpoint (azimuth and elevation) and shape as geometric features, and VGG [22] to extract style features (color and texture).

Geometric Features. MarrNet [27, 24] is composed of two sub-networks. Net1 attempts to estimate the surface normal, depth map and silhouette of an input object. Based on this output, Net2 uses an encoder-decoder structure to recover the 3D shape, azimuth and elevation of the object. We use the Net2 encoder output (latent space) to represent shape, and the final azimuth and elevation vector from the Net2 decoder to represent azimuth and elevation. Standard color augmentation [10] is used to obtain robust features, as explained in Section 3.1.

Style Features. Although a dataset with furniture style annotations exists [1], the annotations there are noisy, since they are based on metadata, and many of the images appear to have a synthetic look. Thus, we chose to make use of generic style feature extraction, rather than training a model specifically to extract furniture style.

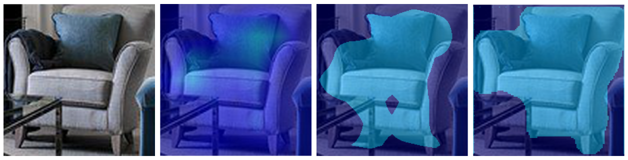
(a) image (b) sal. map (c) sal. mask (d) inst. mask

Figure 2: Given a foreground object image (a), we use Grad-GAM [21] to compute the saliency map (b) for the highest activation in the last layer of VGG (within the furniture category of WordNet [17], in this case 'sofa'). A binary mask (c) is obtained by thresholding with the saliency map's median. Note that the mask in (c) covers the foreground object almost as effectively as an instance segmentation mask produced by Mask-RCNN (d). Additional saliency mask examples may be found in the supplementary.

Style transfer between images has been extensively studied [3, 8, 31]. Although the notion of style in images mostly captures texture and color, and only partially coincides with the notion of style in furniture, our experiments show that these features are nevertheless useful for furniture retrieval. Specifically, as suggested by Huang and Belongie [8], we use the channel-wise means and the variances of deep feature maps to encode style. The feature maps are extracted by VGG [22], as is commonly done in image style transfer. When extracting the style features of a foreground object, we aim to avoid any influence of the local surrounding background. This is achieved by extracting a salience map for the predicted category [21], obtaining a binary foreground mask by thresholding, and extracting the style features only within the mask area, as demonstrated in Fig. 2. This method for obtaining the mask is applicable to more categories than instance segmentation, without requiring extra supervision. Two style vectors are extracted, corresponding to the channel-wise means and variances of the last convolutional layer of VGG. Augmentation for robustness (Section 3.1) is performed via rotations from -15 to +15 degrees, where empty space is filled with gray color.

Abstract feature normalization. Since shape, style mean, and standard deviation representation are vectors extracted from hidden layers of different networks, each feature may have different statistics. Thus, we estimate the population mean and variance of each feature in our training set, and use them to shift and scale the features.

4.3. Student Network

Our goal is to train a student network capable of predicting the foreground features described above from a background image with a foreground bounding box. We initialize the student network to VGG [22] pretrained on Place365 [30], and replace its fully connected layers with a collection of modules, each producing one kind of foreground features (see Fig. 1). Thus, feeding a background image through this network produces a background embedding. The intended foreground location is specified via a binary mask of the same dimensions as the background image, and this mask is fed into a separate branch, whose results are concatenated with those of the VGG branch and then merged. The merged feature map is then given to several different modules that predict each foreground feature separately. Each module consists of two fully connected layers. Additional architectural details may be found in the supplementary. The entire network is trained end-to-end.

Loss Function. Geometric structure-aware loss [23] is used for azimuth and elevation, while cosine distance is used for the shape and style features. The full loss is a weighted combination with weights [azimuth, elevation, shape, style mean, style std] = [1, 1, 0.6, 1, 1]. The weights were chosen to scale the loss terms into a similar scale.

5. Experiments

To evaluate the effectiveness of our approach for fine-grained foreground retrieval, we compare our model with the current state-of-the-art in foreground retrieval (given a background, retrieve suitable foregrounds) [29], referred to as UFO (Unconstrained Foreground Object search), and a standard baseline in fine-grained retrieval (given an object, retrieve similar objects) [26], referred to as SD (Selective Descriptor).

Neither UFO, nor SD, were designed for fine-grained foreground retrieval (given a background, retrieve foregrounds based on fine-grained compatibility). Thus, for a fair comparison, we modify them as described in 5.1 and 5.2. To the best of our knowledge, there is no existing dataset that could be used for evaluation of fine-grained foreground retrieval. We thus construct an evaluation dataset for this task, and annotate it using Amazon Mechanical Turk, as described in Section 5.3.

5.1. Retraining UFO for fine-grained retrieval

The UFO model [29] is originally trained for *coarse-grained* foreground retrieval, and requires instance segmentation to be trained. Thus, we considered training it using the MS COCO dataset, which comes with instance segmentation, and has several furniture categories, namely ‘chair’, ‘couch’, and ‘bed’. However, as explained in Section 2.1, most images in MS COCO have a narrow field-of-view, and do not contain sufficient surrounding background around the foreground objects, making them ill-suited for foreground retrieval. Therefore, we retrain the UFO model [29] using our dataset (Section 4.1), where images have adequately wide field-of-view, using Mask-RCNN pretrained on MS COCO to provide instance segmentation of the foreground objects.

To ensure that the extracted foregrounds are of reasonable quality (e.g., not overly cropped or fragmented), we filter out those extracted foregrounds for which Mask-RCNN could not detect an object of the correct foreground category. Having resized the background images to 224×224, we also discard foregrounds whose bounding boxes are too large (> 150), or too small (< 30).

For a fair comparison, we also retrain our model using the bounding boxes generated by Mask-RCNN. Thus, our model is trained using exactly the same training data as UFO, with the difference that we use easily obtained bounding boxes, rather than expensive instance segmentation. Example retrieval results by both methods are shown in Fig. 3. These results qualitatively show that the foreground objects retrieved by our method exhibit poses and styles that appear more compatible with the background. For example, nearly all chairs retrieved by our method are office chairs facing to the left, while UFO retrieved many right-facing and non-office chairs. Our quantitative comparison in Section 5.4 confirms these observations.

5.2. Adapting SD for foreground retrieval

SD [26] is a standard baseline for fine-grained image retrieval, however, it does not support our foreground retrieval setting (given background, retrieve foreground). Therefore, we compare with SD in two different alternative settings: (i) SD-FG: given the original foreground object, selective descriptor aggregation [26] is used to retrieve similar foregrounds. This task is much easier than our foreground retrieval setting, which does not have the benefit of access to the original foreground. Thus, the SD-FG performance may be considered as an upper bound for the foreground retrieval task. (ii) SD-BG: given only the background as input (as in our setting), selective descriptor aggregation is used to retrieve foreground objects whose background is similar to the input. Notice that this method can only search among foreground candidates with known background. Quantitative comparisons for SD-FG and SD-BG are shown in Section 5.4 and the supplementary, respectively.

5.3. Evaluation dataset

An existing foreground retrieval evaluation dataset, from Zhao et al. [28], is composed of semantically different categories, such as ‘bottle’,‘dog’, and‘plant’. For each background, there is a set of foreground objects, annotated as either good or bad (binary label). The only furniture category represented in this dataset is ‘chair’, the labeling appears to consider viewpoint, but not style, and the foreground resolution is low. Thus, this dataset is not suitable for fine-grained

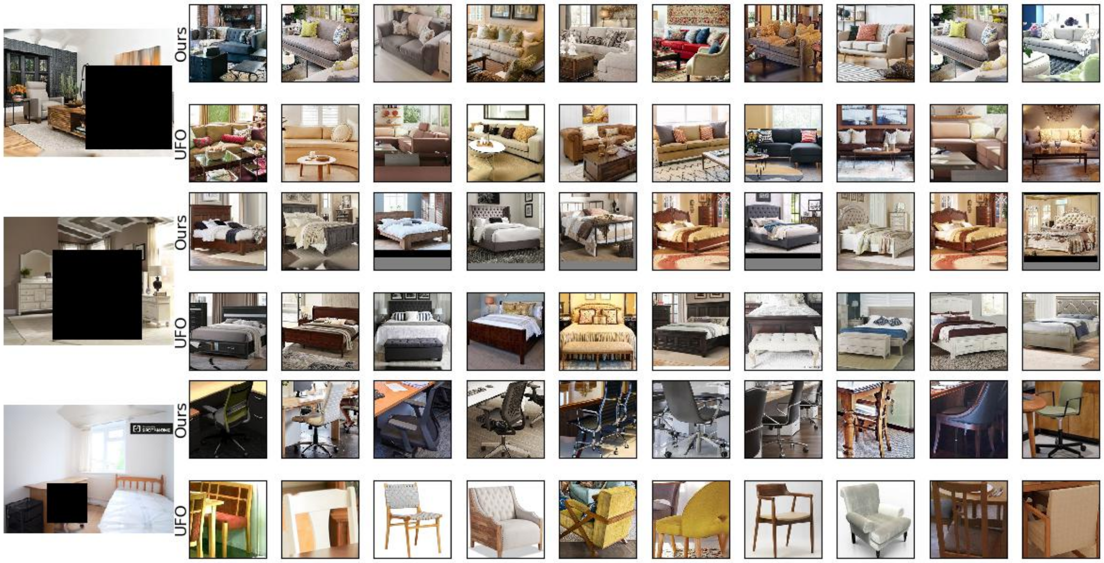

Figure 3: Foreground retrieval results for our method and for UFO [29], after training on our dataset, annotated using Mask-RCNN, as described in Section 5.1. The images on the left are query backgrounds with the location of the foreground indicated by the black square. To the right of each query we show the top 10 results retrieved by each of the two methods.

retrieval, necessitating the creation of a new evaluation set.

For each of the three furniture categories that were used to train the UFO model (bed, chair, and couch), we randomly select five background images from our validation set, and ask AMT workers to annotate a number of candidate foreground object images according to their compatibility to the background.

In order to ensure a manageable number of foregrounds to be annotated by each worker, while at the same time ensuring the set of foregrounds contains enough suitable candidates, we first perform a pre-screening stage. We first *randomly* select 100 foregrounds for each of the 15 query backgrounds, and recruit a small number of expert AMT workers to annotate them, such that each foreground is annotated at least three times. Based on these pre-screening annotations, the top 30 foregrounds are chosen for each background. In our experience, the foregrounds selected in this manner exhibit a mixture of fitness levels.

In the next, main stage, for each background, workers are first shown a preview screen with thumbnails of the 30 candidates, so that they can preview them and calibrate their expectations. Next, they are presented with 30 screens, each showing the background alongside a single foreground candidate. The workers are asked to assign each candidate one out of four fitness scores:

3 – “can crop out foreground and paste it directly”
2 – “foreground fits after minor adjustment”
1 – “foreground fits after major adjustment”
0 – “foreground does not fit at all”

We also include the original foreground as a positive indicator, i.e., the annotations of workers that assign the original foreground a low score are discarded. Note that the original foregrounds are only used to evaluate workers, they are not part of our evaluation set, and are not used for evaluating different algorithms.

Each foreground is annotated at least 8 times (without counting the annotations from the pre-screening stage). The mean score is used to represent how well each foreground is perceived to fit the query background. The rankings of the foregrounds for all 15 backgrounds are shown in supplementary. It may be seen that the ranking induced by these scores is plausible, with the top few candidates matching the background well in terms of pose and style, while clearly unrelated candidates receive much lower scores.

5.4. Quantitative Evaluation

Mean Average Precision (mAP) and Normalized Discounted Cumulative Gain (nDCG) are commonly used metrics for comparing rankings [14]. mAP requires binary labels (either good or bad), thus we set a threshold of 2 to convert mean fitness scores to binary labels. nDCG requires a label with several fitness levels, and we choose $k = 5$, i.e., consider only the top 5 scores in the ranking. The results using both metrics are reported in Tables 1 and 2.

Note that UFO [29] performs better when trained on our dataset, compared to training on MS COCO, indicating that our training set is better suited for fine-grained foreground retrieval. Nevertheless, our method outperforms UFO by a large margin. Furthermore, the results are comparable or

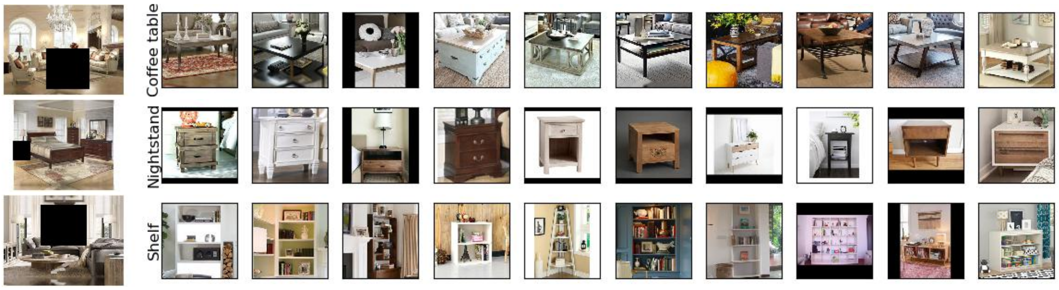

Figure 4: Foreground retrieval results for other furniture categories using our model trained on our dataset annotated by Faster-RCNN [19] pretrained on OpenImage dataset [11], as described in Section 4.1.

	Bed	Chair	Couch	Overall
UFO: trained on MS-COCO	0.373	0.471	0.262	0.369
UFO: trained on our dataset	0.395	0.508	0.288	0.397
SD: SD-FG	0.457	**0.641**	0.340	0.479
Ours: no augmentation	0.354	0.407	0.299	0.353
Ours: with uniform weights	0.529	0.567	**0.398**	0.498
Ours: full	**0.542**	0.574	0.384	**0.500**

Table 1: Comparison with UFO and SD-FG in terms of Mean Average Precision, using a score threshold of 2.

	Bed	Chair	Couch	Overall
UFO: trained on MS-COCO	0.76	0.683	0.744	0.729
UFO: trained on our dataset	0.744	0.722	0.763	0.743
SD: SD-FG	0.795	**0.810**	**0.785**	0.797
Ours: no augmentation	0.740	0.671	0.739	0.716
Ours: with uniform weights	0.816	0.768	0.765	0.783
Ours: full	**0.848**	0.777	0.764	**0.797**

Table 2: Comparison with UFO and SD-FG in terms of Normalized Discounted Cumulative Gain, $k = 5$.

better than those achieved using SD-FG, despite the fact that SD-FG is provided with the original foreground, while our method is only provided with the target background. The comparison with SD-BG is not included in these tables, since it is only applicable to a subset of the evaluation set: 13 foregrounds with known background (instead of 30) for each background image. The results of a comparison with SD-BG on this subset is included in the supplementary, where it may be seen that our method outperforms SD-BG by a large margin.

We also include in the comparison two ablated variants of our method: the “no augmentation” variant does not perform the augmentation described in Section 3.1, and the “uniform weights” variant assigns equal weights to all features, instead of using the weights obtained as described in Section 3.2. In most categories (and overall) the full method performs better than its two ablated variants. More quantitative results, as well as additional ablations, can be found in the supplementary.

5.5. Additional Results

Since training our model only requires easily obtained bounding box annotations, we are able to leverage large existing object detection datasets, such as the OpenImage dataset [11] with 27 furniture categories. This enables training our model to retrieve objects from many different categories. As an example, Fig. 4 shows retrieval results for the categories ‘coffee table’, ‘nightstand’, and ‘shelf’. Our method may also be used without requiring the foreground category as input. In this scenario, the retrieval process is unchanged, but the set of candidate objects is no longer filtered to contain only objects of the specified category. The supplementary video demonstrates that our method performs in a satisfactory manner in this scenario as well.

6. Conclusions

In contrast to existing foreground retrieval methods that focus on matching the high-level semantics between a background and a foreground object, we address the problem of fine-grained foreground retrieval that aims to select the most compatible foreground from a set of semantically similar objects. Our method may be applied to diverse foreground categories and background scene types, since it does not require instance segmentation, unlike current state-of-the-art methods. Instead, we leverage large scale datasets and pretrained models for object detection to construct a large training dataset for the fine-grained foreground retrieval task. Casting the problem as one of domain adaptation, we propose to apply teacher-student learning to train a model to predict foreground features relevant to the retrieval task from images of background scenes. Our approach allows selecting a set of meaningful features to be used for retrieval, and assigning these features task-dependent weights. Experiments demonstrate that our method is able to perform better fine-grained furniture retrieval for indoor scenes than the existing state-of-the-art.

Acknowledgments: This work was supported in part by the Israel Science Foundation (grant 2366/16) and a gift from Adobe Systems.

References

[1] Divyansh Aggarwal, Elchin Valiyev, Fadime Sener, and Angela Yao. Learning style compatibility for furniture. In *German Conference on Pattern Recognition*, pages 552–566. Springer, 2018.

[2] Sefi Bell-Kligler, Assaf Shocher, and Michal Irani. Blind super-resolution kernel estimation using an internal-GAN. In *Advances in Neural Information Processing Systems*, pages 284–293, 2019.

[3] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proc. IEEE CVPR*, pages 2414–2423, 2016.

[4] Yuying Ge, Ruimao Zhang, Xiaogang Wang, Xiaoou Tang, and Ping Luo. DeepFashion2: A versatile benchmark for detection, pose estimation, segmentation and re-identification of clothing images. In *Proc. IEEE CVPR*, pages 5337–5345, 2019.

[5] Jacob Goldberger and Ehud Ben-Reuven. Training deep neural-networks using a noise adaptation layer. 2016.

[6] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *Proc. IEEE ICCV*, pages 2961–2969, 2017.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. IEEE CVPR*, pages 770–778, 2016.

[8] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proc. IEEE ICCV*, pages 1501–1510, 2017.

[9] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

[10] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[11] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Tom Duerig, et al. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *arXiv preprint arXiv:1811.00982*, 2018.

[12] Jean-François Lalonde, Derek Hoiem, Alexei A. Efros, Carsten Rother, John Winn, and Antonio Criminisi. Photo clip art. *ACM Transactions on Graphics (SIGGRAPH 2007)*, 26(3):3, August 2007.

[13] Jinyu Li, Michael L. Seltzer, Xi Wang, Rui Zhao, and Yifan Gong. Large-scale domain adaptation via teacher-student learning. In *Proc. Interspeech 2017*, pages 2386–2390, 2017.

[14] Tie-Yan Liu et al. Learning to rank for information retrieval. *Foundations and Trends® in Information Retrieval*, 3(3):225–331, 2009.

[15] Ziwei Liu, Ping Luo, Shi Qiu, Xiaogang Wang, and Xiaoou Tang. DeepFashion: Powering robust clothes recognition and retrieval with rich annotations. In *Proc. IEEE CVPR*, pages 1096–1104, 2016.

[16] Zhong Meng, Jinyu Li, Yifan Gong, and Biing-Hwang Juang. Adversarial teacher-student learning for unsupervised domain adaptation. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5949–5953. IEEE, 2018.

[17] George A Miller. *WordNet: An electronic lexical database*. MIT press, 1998.

[18] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proc. IEEE CVPR*, pages 779–788, 2016.

[19] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, pages 91–99, 2015.

[20] F. Schroff, D. Kalenichenko, and J. Philbin. FaceNet: a unified embedding for face recognition and clustering. In *Proc. IEEE CVPR*, pages 815–823, 2015.

[21] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual explanations from deep networks via gradient-based localization. In *Proc. IEEE ICCV*, pages 618–626, 2017.

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

[23] Hao Su, Charles R Qi, Yangyan Li, and Leonidas J Guibas. Render for CNN: Viewpoint estimation in images using CNNs trained with rendered 3d model views. In *Proc. IEEE CVPR*, pages 2686–2694, 2015.

[24] Xingyuan Sun, Jiajun Wu, Xiuming Zhang, Zhoutong Zhang, Chengkai Zhang, Tianfan Xue, Joshua B Tenenbaum, and William T Freeman. Pix3d: Dataset and methods for single-image 3d shape modeling. In *Proc. IEEE CVPR*, pages 2974–2983, 2018.

[25] Fuwen Tan, Crispin Bernier, Benjamin Cohen, Vicente Ordonez, and Connelly Barnes. Where and who? automatic semantic-aware person composition. In *Proc. IEEE Winter Conf. on Applications of Computer Vision (WACV)*, 2018.

[26] Xiu-Shen Wei, Jian-Hao Luo, Jianxin Wu, and Zhi-Hua Zhou. Selective convolutional descriptor aggregation for fine-grained image retrieval. *IEEE Transactions on Image Processing*, 26(6):2868–2881, 2017.

[27] Jiajun Wu, Yifan Wang, Tianfan Xue, Xingyuan Sun, Bill Freeman, and Josh Tenenbaum. MarrNet: 3d shape reconstruction via 2.5d sketches. In *Advances in Neural Information Processing Systems*, pages 540–550, 2017.

[28] Hengshuang Zhao, Xiaohui Shen, Zhe Lin, Kalyan Sunkavalli, Brian Price, and Jiaya Jia. Compositing-aware image search. In *Proc. ECCV*, 2018.

[29] Yinan Zhao, Brian Price, Scott Cohen, and Danna Gurari. Unconstrained foreground object search. In *Proc. IEEE ICCV*, pages 2030–2039, 2019.

[30] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2017.

[31] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proc. IEEE ICCV*, pages 2223–2232, 2017.