Foreground Object Search by Distilling Composite Image Feature

Bo Zhang[1], Jiacheng Sui[2], and Li Niu*[1]

[1]Center for Machine Cognitive Computing of Artificial Intelligence Institute
Artificial Intelligence Institute, Shanghai Jiao Tong University
{bo-zhang, ustcnewly}@sjtu.edu.cn
[2]Xian Jiao Tong University
rookiecharles99@gmail.com

Abstract

*Foreground object search (FOS) aims to find compatible foreground objects for a given background image, producing realistic composite image. We observe that competitive retrieval performance could be achieved by using a discriminator to predict the compatibility of composite image, but this approach has unaffordable time cost. To this end, we propose a novel FOS method via **dis**tilling **co**mposite feature (DiscoFOS). Specifically, the abovementioned discriminator serves as teacher network. The student network employs two encoders to extract foreground feature and background feature. Their interaction output is enforced to match the composite image feature from the teacher network. Additionally, previous works did not release their datasets, so we contribute two datasets for FOS task: S-FOSD dataset with synthetic composite images and R-FOSD dataset with real composite images. Extensive experiments on our two datasets demonstrate the superiority of the proposed method over previous approaches. The dataset and code are available at https://github.com/bcmi/Foreground-Object-Search-Dataset-FOSD.*

1. Introduction

Foreground Object Search (FOS) aims to find compatible foregrounds from specified category for a given background image which has a query bounding box indicating the foreground location [42]. More precisely, an object is compatible with a background image if it can be realistically composited into the image [44], as illustrated in Figure 1. FOS is a core technique in many image composition applications [24]. For example, FOS technique can help users acquire suitable foregrounds from a foreground pool automatically and efficiently for object insertion in photo editing [19]. Moreover, FOS also can be used to fill a region comprising undesired objects using new foreground [44].

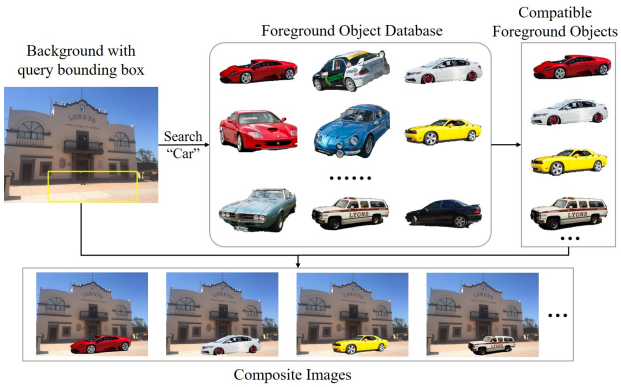

Figure 1. Illustration of foreground object search. Given a background image with query bounding box (yellow), foreground object search aims to find compatible foreground objects of a specified category from a database, which is composited with the background to produce a realistic composite image.

There exist many factors that affect the compatibility between background and foreground, including semantics, style (*e.g.*, color and texture), lighting, and geometry (*e.g.*, shape and viewpoint). Previous works on FOS may consider different factors. For example, early methods [42, 44] focused on the semantic compatibility. Recent works [19, 37, 47] considered geometry and other factors, including style [37] and lighting [47]. In this paper, we focus on semantics and geometry compatibility following [19], because incompatible color and lighting between the background and foreground can be tackled to some extent by image harmonization [6, 21, 4].

*Corresponding author

The general pipeline of most existing methods [42, 44, 47, 19, 37] is to learn an embedding space with two encoders respectively for background and foreground, so that compatible background and foreground are close to each other in this space. Alternatively, some approaches [46, 44] trained a discriminator to predict background-foreground compatibility by feeding the composite image. Based on preliminary experiments, we observe that the discriminator can achieve much better results than encoders when taking the cropped composite image as input (see teacher network in Figure 3). We conjecture that the forward pass in the discriminator allows thorough interaction between background and foreground, which could provide useful contextual cues for estimating background-foreground compatibility. However, given a background image, it is very time-consuming to composite with each foreground image and perform forward computation for each composite image. Motivated by this, we propose a novel FOS framework called DiscoFOS via knowledge distillation. Specifically, we distill the knowledge of composite image from the discriminator to two encoders, in which we enforce the interaction output of foreground feature and background feature to match with the composite image feature. How to design the interaction between two encoders is challenging, due to the trade-off between performance and computational cost. On the one hand, insufficient interaction between two encoders may be unable to mimic the rich knowledge in composite image feature. On the other hand, sufficient interaction would largely increase the computational burden. Considering the abovementioned trade-off, we perform interaction only on the last feature maps of two encoders, which achieves significant performance improvement with acceptable computational overhead.

Since previous works (CAIS [42], UFO [44], and GALA [47]) did not release their datasets, we build our own datasets based on an existing large-scale real-world dataset, *i.e.*, Open Images [15], as illustrated in Figure 2. We construct two FOS Datasets respectively containing Synthetic composite images and Real composite images, abbreviated as S-FOSD and R-FOSD respectively. We first introduce the S-FOSD dataset. Given a real image with instance segmentation mask, we choose one object and fill its bounding box with image mean values to get the background. Meanwhile, we crop out this object as foreground. After removing unsuitable categories and occluded foregrounds, the resultant dataset contains 57,859 backgrounds and 63,619 foregrounds. Following [42, 47], for each background image, we deem the foreground object from the same image as ground-truth. For R-FOSD dataset, we collect images from Internet as background images and draw a bounding box at the expected foreground location as query bounding box. R-FOSD dataset shares the same foregrounds with the test set of S-FOSD dataset. Then we employ multiple human annotators to label the compatibility of each pair of background and foreground. *In summary, S-FOSD dataset is lowcost and highly scalable, but has neither complete background nor ground-truth negative samples. Oppositely, R-FOSD dataset has complete background image with both positive and negative foregrounds annotated by human, yet is unscalable due to the high annotation cost.* In our experiments, S-FOSD dataset is used for both training and validating model, while R-FOSD dataset is only used for model evaluation. More details about dataset construction could be found in Section 3.

We evaluate our method on the proposed datasets, which validates the superiority of the proposed method over previous approaches. Our major contributions can be summarized as follows: 1) To facilitate the research on FOS task, we contribute two public datasets: S-FOSD dataset with synthetic composite images and R-FOSD dataset with real composite images. 2) We propose a novel method named DiscoFOS that improves foreground object search by distilling the knowledge of composite image feature into two encoders. 3) Extensive experiments demonstrate the superiority of the proposed method over previous baselines on our datasets.

2. Related Works

2.1. Foreground Object Search

Early works [16, 3] employed hand-crafted features to match background with foreground, yet their performance may be limited by the representation ability of hand-crafted features. Recent work applied deep learning based feature for foreground retrieval. For example, [32] utilized deep features to capture local context particularly for person compositing. [46] trained a discriminator to estimate the realism of composites, which is available for selecting compatible foregrounds by compositing each foreground with the background, but is computationally expensive.

More recent methods [42, 44, 47, 37, 19] typically trained two encoders to extract background feature and foreground feature, and then measured background-foreground compatibility by calculating feature similarity. These methods considered different factors that affect the compatibility between background and foreground. For example, the methods [42, 44] considered the semantic compatibility, while the approaches [19, 37, 47] considered geometry and other factors, including style [37] and lighting [47]. In this work, we focus on semantic compatibility and geometry compatibility following [19], and propose a novel foreground object search method that improves the encoders with knowledge distillation. Additionally, previous works [42, 44, 37, 47] did not release their datasets, we contribute two datasets to facilitate the research in this field.

2.2. Image Composition

As summarized in [24], existing image composition works attempted to solve one or some issues which affect the quality of composite image, such as illumination, shadow, and geometry. Image harmonization methods [34, 6, 7, 5] focused on eliminating the color and illumination discrepancy between background and foreground. Besides, some shadow generation works [14, 9, 40, 22, 12] aimed to generate plausible shadow for the inserted foreground. To blend the foreground into the background more naturally, image blending methods [26, 18, 35] paid attention to smoothing the boundary between background and foreground. The closest subtask to ours is object placement [45, 25, 23, 17, 33], which generates reasonable locations and sizes to place foreground over background. Given a background, object placement focuses on predicting bounding box for the inserted foreground, while our task targets at retrieving compatible foregrounds from a specified category for a given query bounding box.

2.3. Knowledge Distillation

Knowledge distillation methods usually improve the student network by forcing to mimic the behaviors of the teacher network, such as soft predictions [11], logits [1], intermediate feature maps [27], or attention maps [39]. Apart from the above works on image classification, some recent works extend knowledge distillation to more complex vision tasks, including semantic segmentation [41, 29], object detection [20, 2], face recognition [13], and so on.

Previous works [19, 37] have also considered introducing knowledge distillation to foreground object search, in which foreground (*resp.*, background) encoder plays the role of the teacher (*resp.*, student) network and the foreground information is distilled from foreground embedding to background embedding. However, these approaches rely on fine-grained annotations of foreground attributes [19] or multiple pretrained models [37], which may hinder the generalization to unseen foreground categories. Unlike the above methods, our method adopts a composite image discriminator as teacher network and two encoders as student network, in which we distill composite image feature from discriminator to the interaction output of foreground feature and background feature, which proves to be effective and computationally affordable.

3. Dataset Construction

Our datasets are constructed on existing Open Images dataset [15] that contains 9.2 million images covering diverse scenes, making it very suitable for real-world evaluation. We build two datasets for foreground object search: S-FOSD with synthetic composite images and R-FOSD with real composite images, which have different ways to acquire backgrounds and foregrounds (see Figure 2).

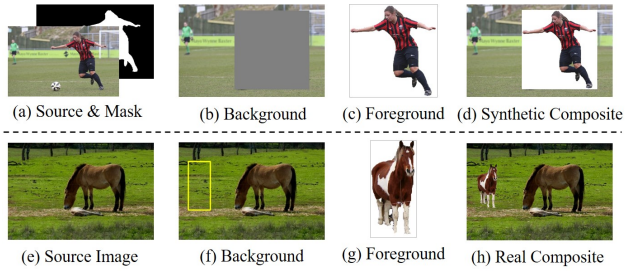

Figure 2. The illustration of building our foreground object search (FOS) datasets. Top: FOS Dataset with Synthetic composite images (S-FOSD). Bottom: FOS Dataset with Real composite images (R-FOSD). More details about dataset construction can be seen in Section 3.

3.1. S-FOSD Dataset

Foreground Object Selection. Open Images dataset [15] provides instance segmentation masks for 2.8 million object instances in 350 categories. To accommodate our task, we delete the categories and objects that are unsuitable for the task or beyond our focus (geometry and semantic compatibility), after which 32 categories remain. The detailed rules for object selection and the complete category list can be found in Supplementary.

Background and Foreground Generation. By using the segmentation masks, we generate background and foreground images in a similar way to previous works [42, 47, 44]. Specifically, as shown in Figure 2 (a) $\sim$ (c), given an image with instance segmentation masks, we choose one object and fill its bounding box with image mean values to get the background. Meanwhile, we crop out the object and paste it on white background, producing the foreground. With the background and foreground, we can obtain the synthetic composite image by resizing the foreground and placing it in the query bounding box on the background (see Figure 2 (d)). Additionally, we have also tried image inpainting [31, 43, 38] to fill the object bounding box, but got unsatisfactory results, probably due to large missing content and the residues of erased object (*e.g.*, shadow). After that, we obtain over 63,000 pairs of background and foreground covering 32 categories, with a maximum of 5,000 images and a minimum of 500 images in each category. For a given background, the foreground cropped from the same image is naturally compatible and thus deemed as ground-truth following [42, 47, 44].

Dataset Split. S-FOSD dataset is employed for both training and testing, as its construction is low-cost and highly scalable. We first select test samples to build the test set and the rest forms the training set. For reliable performance evaluation, we build test set mainly concerning its diversity and quality. Finally, we get 20 backgrounds and 200

foregrounds for each category, which contains 20 pairs of background and foreground. The remaining 57,219 pairs of background and foreground form the training set.

3.2. R-FOSD Dataset

Background and Foreground Generation. When building R-FOSD dataset, we directly adopt the foregrounds of the test set in S-FOSD dataset and collect images from Internet as backgrounds. It is very likely that a random image is unsuitable for compositing with any test foreground. Thus, we collect candidate backgrounds by searching similar images to the test backgrounds of S-FOSD dataset. After that, we draw a bounding box at the desired foreground location as query bounding box (see Figure 2 (f)). The size and location of the bounding box are decided by first mimicking the foreground in the similar background of S-FOSD dataset and then manual inspection. For each pair of background and foreground, we resize the foreground and place it in the query bounding box on the source image, generating a real composite image (see Figure 2 (h)). Finally, R-FOSD dataset contains 32 categories, each of which has 20 background images and 200 foreground images.

Compatibility Labelling. To acquire the binary compatibility label (1 for compatible background-foreground pair and 0 for incompatible pair), we employ three human annotators to label the compatibility for $32 \times 20 \times 200$ pairs of background and foreground. During annotation, we show real composite images and request annotators to assign binary labels by considering the semantics and geometry compatibility between background and foreground. Finally, for one background, we only consider the foregrounds for which all three human annotators label 1 as compatible and the others are treated as incompatible. Note that manually annotating the dataset is expensive, as it requires labelling a quadratically growing number of background and foreground pairs. Therefore, we only use R-FOSD dataset as a test set. To keep consistent with training data, we also fill the query bounding box of the background in R-FOSD dataset with image mean values at test time and the complete background image of R-FOSD dataset is only used to obtain the final composite image (Figure 2 (h)).

4. Methodology

In this section, we describe our proposed method for foreground object search. As illustrated in Figure 3, we train a discriminator D to predict the compatibility of composite image, which serves as the teacher network (see Section 4.1). We employ two encoders E^b and E^f (see Section 4.2) as well as a light-weight knowledge distillation module E^d (see Section 4.3) as the student network. The two encoders respectively extract background feature $\mathbf{F}^b$ and foreground feature $\mathbf{F}^f$, which are fed into distillation module to interact with each other. During training stage, we enforce the interaction output $\mathbf{F}^d$ to match with the composite image feature $\mathbf{F}^c$, in which the background-foreground compatibility information is distilled to $\mathbf{F}^d$. Finally, we predict compatibility scores for pairwise background and foreground based on the distilled feature $\mathbf{F}^d$.

4.1. Composite Image Discriminator

Network Architecture. Given a background image $\mathbf{I}^b$ with a query bounding box and a foreground image $\mathbf{I}^f$, we first generate a synthetic composite image $\mathbf{I}^c$ as in Figure 2 (d). Then, we employ a discriminator D that takes a composite image as input to predict whether background and foreground of the composite image are compatible. We implement the discriminator as a binary classifier that consists of backbone network and classification head. Moreover, based on preliminary experiments, we find that feeding a cropped composite image into the discriminator achieves significantly better performance than feeding the whole composite image. Specifically, we crop the composite image, ensuring that the foreground object is located at the center and the area of foreground bounding box is about 50 percent of the whole crop. Then, we resize the crop to the input size 224 $\times$ 224, which is denoted as $\widetilde{\mathbf{I}}^c$. The superiority of feeding cropped composite image can be attributed to that the contextual information near the foreground may be more helpful and the foreground is aligned with crop center. Here we refer to the crop box as B. With the cropped composite image, the discriminator first extracts the composite image feature map $\mathbf{F}^c$ by backbone network and then apply global average pooling (GAP) layer followed by a binary classifier. We train the discriminator using binary cross-entropy loss:

$$\mathcal{L}_D = -\log\left(p(\widetilde{\mathbf{I}}^c)_y\right), \tag{1}$$

in which $p(\cdot)_y$ means the predicted probability corresponding to the ground-truth compatibility label y.

Positive and Negative Samples Generation. During training phase, we consider compatible (*resp.*, incompatible) foreground objects as positive (*resp.*, negative) samples for a given background. The foreground cropped from the same image as the background is naturally viewed as positive sample. However, other foreground objects may also be compatible with the background. To guarantee the effectiveness of training samples, similar to [44], we train a binary classifier using VGG-19 [30] pretrained on ImageNet [8] as backbone network to help filter out training samples. Given a pair of background image and foreground image, the classifier takes their composite image as input and predicts the compatibility score. When training this classifier, we assume that only background-foreground pairs from same images are positive and others are negative. With the trained classifier, we restrict negative samples to only include those foreground objects which are confidently

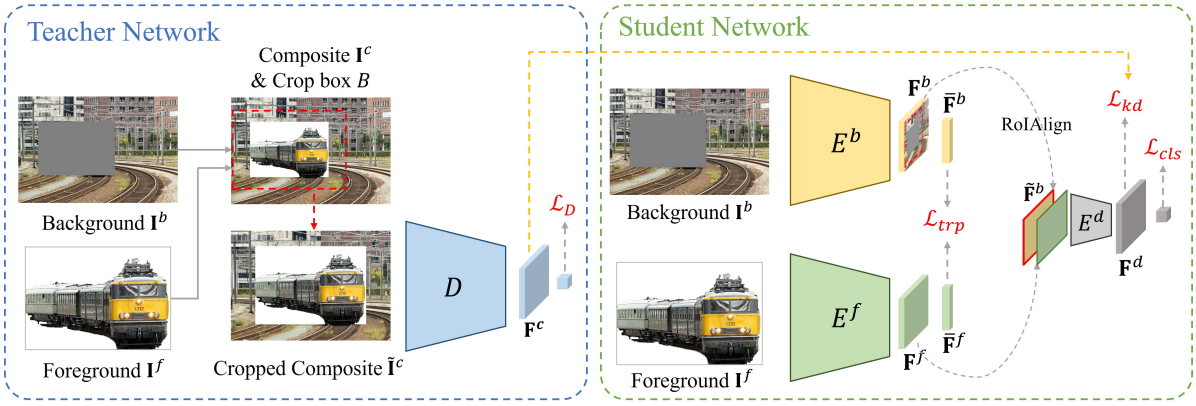

Figure 3. Illustration of the proposed DiscoFOS for foreground object search. The discriminator D is first trained to predict the background-foreground compatibility of input composite image $\mathbf{I}^c$, whose intermediate feature map $\mathbf{F}^c$ then servers as distillation target to train the student model. The student network first extracts background feature $\mathbf{F}^b$ and foreground feature $\mathbf{F}^f$ by two encoders $\{E^f, E^b\}$, then applies the features to generate distilled feature $\mathbf{F}^d$ and compatibility prediction by knowledge distillation module E^d.

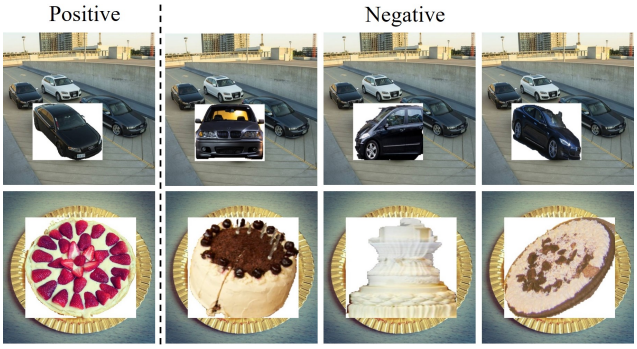

Figure 4. Examples of positive and negative samples used to train our models. Given a background in S-FOSD dataset, we have one positive foreground and one or more negative foregrounds, which are filled with white background pixels.

classified as incompatible, that is, compatibility score is lower than a threshold (0.3 in our experiments). Therefore, given a background image, we have a single positive foreground and one or more negative foregrounds in S-FOSD dataset (see Figure 4). The obtained positive and negative samples are used to train the composite image discriminator in this section and the student network in Section 4.2, 4.3.

4.2. Background and Foreground Encoders

We employ two encoders E^b and E^f to extract feature respectively from background image $\mathbf{I}^b$ and foreground image $\mathbf{I}^f$. Correspondingly, the background encoder E^b outputs background feature map $\mathbf{F}^b$ and the foreground encoder E^f outputs foreground feature map $\mathbf{F}^f$, both of which have the same shape, *i.e.*, $\mathbf{F}^b, \mathbf{F}^f \in \mathcal{R}^{h\times w\times c}$. Here $h \times w$ is the spatial size and c is the number of channels. Then we apply GAP layer to yield background and foreground feature vectors, denoted as $\bar{\mathbf{F}}^b, \bar{\mathbf{F}}^f \in \mathcal{R}^c$. After mapping the foreground and background into the common feature space, we enforce the compatible background and foreground to be close to each other in this space. Thus, the compatibility between foreground and background can be measured by computing the cosine similarity between their features. To this end, following previous works [37, 42, 44, 47], we train the encoders using triplet loss [28], which tends to pull the positive sample (*i.e.*, compatible foreground) to the anchor (*i.e.*, given background) and push the negative sample (*i.e.*, incompatible foreground) away from the anchor in the feature space. During training stage, we tend to minimize the following loss function:

$$\mathcal{L}_{trp} = \max\left(0, m + S(\bar{\mathbf{F}}^b, \bar{\mathbf{F}}^f_n) - S(\bar{\mathbf{F}}^b, \bar{\mathbf{F}}^f_p)\right), \quad (2)$$

where $S(\cdot)$ represents cosine similarity and m is a positive margin in the range of (0, 1). $\bar{\mathbf{F}}^f_p$ (*resp.*, $\bar{\mathbf{F}}^f_n$) is the feature of positive (*resp.*, negative) foreground for background $\bar{\mathbf{F}}^b$. By minimizing $\mathcal{L}_{trp}$, for a given background, the feature similarity with compatible foreground is expected to be greater than that with incompatible foreground by the margin m. Additionally, the above encoders are usually employed in previous methods [42, 44, 47]. Given a query background and a foreground database, these approaches rank the compatibility of different foregrounds by measuring their feature similarity to the given background.

4.3. Knowledge Distillation Module

After evaluating the discriminator in Section 4.1 and the encoders in Section 4.2, we observe that discriminator can achieve much better results than encoders, probably because the forward propagation in the discriminator allows thorough interaction between background and foreground, providing useful contextual cues for estimating background-

foreground compatibility. However, retrieving foreground images using discriminator requires to composite with each foreground image, which brings heavy computational burden. On the contrary, encoders achieve inferior performance, yet have significantly faster speed due to the exemption from background-foreground interaction. This motivates us to distill discriminator knowledge into encoders.

To achieve this goal, we design a knowledge distillation module E^d, which interacts foreground feature with background feature and enforces the interaction output to match with the composite image feature $\mathbf{F}^c$ from the discriminator. Recall that the composite image feature is extracted from cropped composite image, with the crop bounding box denoted as B (see Section 4.1). So we apply RoIAlign [10] with the bounding box B to obtain local background feature map $\widetilde{\mathbf{F}}^b$ from global background feature map $\mathbf{F}^b$ produced by background encoder E^b. Then, we resize RoIAlign output to be of the same shape as $\mathbf{F}^b$, *i.e.*, $\widetilde{\mathbf{F}}^b \in \mathcal{R}^{h\times w\times c}$. The local background feature is supposed to encode contextual information surrounding the foreground. Meanwhile, we utilize foreground encoder E^f to extract foreground feature map $\mathbf{F}^f$. Given the foreground feature map $\mathbf{F}^f$ and local background feature map $\widetilde{\mathbf{F}}^b$, we feed their concatenation into E^d to produce a distilled feature map $\mathbf{F}^d \in \mathcal{R}^{h\times w\times c}$. Note the discriminator and two encoders adopt the same backbone network and input image size in our implementation, so their feature maps have the same shape, *i.e.*, $\mathbf{F}^c, \mathbf{F}^d \in \mathcal{R}^{h\times w\times c}$. During training phase, we enforce the distilled feature $\mathbf{F}^d$ to mimic the composite image feature $\mathbf{F}^c$ by L_1 loss:

$$\mathcal{L}_{kd} = \|\mathbf{F}^d - \mathbf{F}^c\|_1. \tag{3}$$

We pool the distilled feature map into a vector and send it to a binary classifier to predict the compatibility. The classifier is also trained using binary cross-entropy loss:

$$\mathcal{L}_{cls} = -\log\left(p(\mathbf{F}^d)_y\right), \tag{4}$$

in which $p(\cdot)_y$ is similarly defined as in Eqn. 1.

Finally, we train the encoders $\{E^f, E^b\}$ and distillation module E^d simultaneously. The overall optimization function can be written as

$$\mathcal{L} = \mathcal{L}_{trp} + \lambda_{kd}\mathcal{L}_{kd} + \lambda_{cls}\mathcal{L}_{cls}, \tag{5}$$

where λ_{kd} and λ_{cls} are trade-off parameters. During inference, our model finds compatible foregrounds for a given background by ranking the predicted compatibility scores.

4.4. Generalization to Real-world Application

To boost the performance of our method on real-world data, we make some modifications to the training procedure of our teacher network and student network, which helps achieve more competitive results on the R-FOSD dataset.

For the teacher network, we utilize the binary classifier in Section 4.1 to extend training samples. Specifically, given a background image, we use the classifier to predict compatibility score for each foreground and treat those with scores larger than 0.8 as positive samples (including the ground-truth foreground), based on which the ratio of positive and negative foregrounds per background is increased from 1:10 to 5:10. Besides, we apply data augmentation to generate additional positive and negative samples from the ground-truth foreground for each background image. Precisely, based on the ground-truth foreground, we produce 2 positive foregrounds through color jitter and Gaussian blur, and 1 negative foreground through affine transformation. After augmentation, the ratio of positive and negative samples changes from 5:10 to 7:11.

For the student network, we select the top-5 foregrounds returned by pretrained teacher network as positive samples (including the ground-truth foreground) and adopt the same negative samples as the teacher network. After applying the same data augmentation scheme as the teacher network, we also have positive and negative samples with the ratio 7:11 to train the student network.

Finally, considering that user-provided bounding boxes may not be very accurate, we also augment bounding boxes when training teacher network and student network. Specifically, we randomly pad the bounding box with the maximum padding space being 30% of the bounding box's width and height.

5. Experiments

5.1. Dataset and Evaluation Metrics

Our S-FOSD dataset is employed for both training and testing, while R-FOSD dataset is only for testing. We employ different evaluation metrics for two datasets considering their difference in the acquisition of ground-truth foregrounds. For each metric, we report the mean evaluation results by averaging the results over all categories. Moreover, we leave the implementation details in Supplementary.

S-FOSD Dataset. The training set has 57,219 pairs of foregrounds and backgrounds covering 32 categories, with a maximum of 4800 pairs and a minimum of 300 pairs in each category. The test set provides 20 backgrounds and 200 foregrounds (including 20 foregrounds from the same images as the backgrounds) for each category. The foreground/background images in the training set and test set have no overlap. Following previous works [42, 47], only the foreground object from the same image is viewed as ground-truth for each background and we adopt Recall@k (R@k) as evaluation metric, which represents the percentage of background queries whose ground-truth foreground appears in top k retrievals ($k = 1, 5, 10, 20$ in our experiments).

R-FOSD Dataset. The R-FOSD dataset adopts the same foreground set as the test set of S-FOSD dataset and collects 20 backgrounds for each category. Each pair of background and foreground is shown to three human annotators to label the compatibility. The resulting dataset contains 4∼190 compatible foregrounds per background, and we adopt mean Average Precision (mAP), mAP@20, and Precision@k (P@k) for evaluation, which are widely used in image retrieval and previous works [44, 47]. Precision@k means the percentage of compatible foreground objects in the top k retrievals, k=1, 5, 10, 20 in our experiments. To reduce unplausible results on the R-FOSD dataset, we exclude the objects with aspect ratios deviating more than 1.2 from that of query bounding box for all methods and compute metrics on the remaining objects.

5.2. Comparison with Existing Methods

Baselines. We compare our approach with previous methods [42, 44, 37, 47] on our two datasets. Shape [42] ranks foreground objects by comparing their aspect ratios with that of the query bounding box on background. CFO [42], UFO [44], FFR [37], and GALA [47] adopt the pipeline described in Section 4.2, which use two encoders to extract background and foreground features, and then rank the compatibility of foreground by computing its feature similarity with background. To adapt to our focus on geometric and semantic incompatibility, we replace the style feature of FFR [37] with semantic feature, in which we obtain the semantic feature by using VGG-19 [30] pretrained on ImageNet [8] following [44]. For GALA [47] that considers lighting and geometry compatibility, we discard lighting transformation and only keep geometry transformation [47] to match our scenario. Besides, we do not compare with IFR [19], as it requires pattern labels of the foreground, which are unavailable in our datasets. For fair comparison, we employ the same backbone network and training set for all baseline methods except Shape [42]. We use the same positive and negative samples for different methods when training on S-FOSD dataset.

Quantitative comparison. The results of baselines and our method on the proposed two datasets are summarized in Table 1. Moreover, when evaluating on S-FOSD dataset, the only ground-truth foreground has the same aspect ratio as the query bounding box, thus Shape [42] can find the ground-truth by shortcut, making the evaluation results meaningless. It can be seen that among all baselines, GALA [47] and FFR [37] are two competitive ones, which employ contrastive learning with self-transformation [47] and MarrNet [36] to enhance the perception of geometric information of foreground, respectively. Nevertheless, our method outperforms all previous approaches by an obvious margin on both S-FOSD and R-FOSD datasets.

Qualitative comparison. We show the retrieval results of different baselines on our two datasets in Figure 5. For each query background, we show the returned top-5 foregrounds by different methods, which shows that our method can generally find compatible foregrounds by taking both semantic and geometric factors into account. For example, for the top-left example, given a query bounding box at the bottom-right seat, the baseline methods may return a standing or frontal person, which is unsuitable to be placed in the query bounding box. In contrast, the top returned persons by our method all have sitting posture, which appear more reasonable in terms of semantic compatibility. In the bottom-right example, given the geometry of background scene, the annotated ground-truth positive foregrounds (green box) have consistent orientation with respect to the train track in background. Our method retrieves more positive foregrounds than other approaches, demonstrating the effectiveness of our method on geometry compatibility. More qualitative results are present in Supplementary.

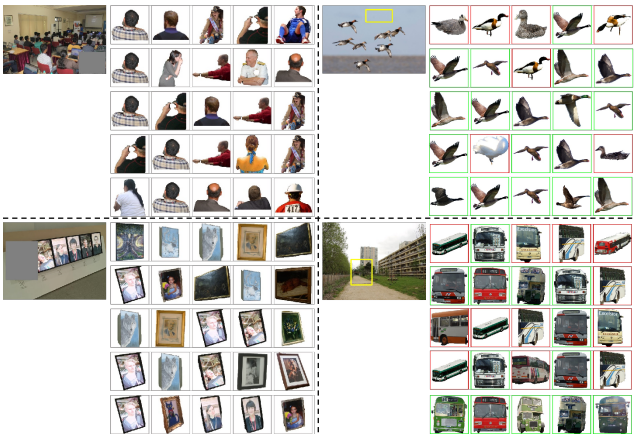

Figure 5. Qualitative comparison on our S-FOSD (left) and R-FOSD (right) datasets. Each of the four examples contains a background image and several rows of the retrieval results, from top to bottom: CFO [42], UFO [44], GALA [47], FFR [37], and ours. Additionally, in the right part, green (*resp.*, red) box represents the foreground with compatible (*resp.*, incompatible) label.

5.3. Ablation Study

In this section, we start from the general pipeline with two encoders of previous methods [42, 44, 47, 19, 37] and evaluate the effectiveness of each component in our method. Different models are evaluated on our S-FOSD dataset and the results are summarized in Table 2. In row 1, we apply two encoders $\{E^b, E^f\}$ to extract background feature $\bar{\mathbf{F}}^b$ and foreground feature $\bar{\mathbf{F}}^f$, which are used to predict the background-foreground compatibility by measuring their feature similarity. It is worth mentioning that all models in Table 2 rank different foregrounds by predicted compatibility scores except the row 1 and row 2, which measure compatibility by calculating feature similarity.

Method	S-FOSD Dataset				R-FOSD Dataset					
	R@1↑	R@5↑	R@10↑	R@20↑	mAP↑	mAP@20↑	P@1↑	P@5↑	P@10↑	P@20↑
Shape [42]	-	-	-	-	50.49	56.22	47.66	49.72	50.03	49.92
CFO [42]	56.09	83.59	91.25	96.88	52.06	62.17	58.33	56.29	55.90	52.85
UFO [44]	54.69	81.72	90.94	95.31	52.73	63.63	64.12	59.11	56.30	53.82
FFR [37]	57.03	86.25	93.28	97.97	53.10	64.12	61.92	59.64	57.56	55.06
GALA [47]	57.50	85.17	93.00	97.33	52.02	62.83	62.56	57.33	55.76	52.93
DiscoFOS	**79.06**	**94.84**	**97.34**	**99.38**	**56.70**	**68.56**	**67.75**	**64.64**	**62.25**	**58.90**

Table 1. Comparison with existing methods on our S-FOSD dataset and R-FOSD dataset. Best results are denoted in boldface.

	E^b	E^d	R@1↑	R@5↑	R@10↑	R@20↑
1	$\bar{\mathbf{F}}^b$		54.83	81.17	90.67	95.00
2	$\hat{\mathbf{F}}^b$		52.50	79.69	88.75	93.00
3	$\bar{\mathbf{F}}^b$	$p(\cdot)_y$	56.72	87.19	94.69	97.97
4	$\bar{\mathbf{F}}^b$	$[\bar{\mathbf{F}}^b, \bar{\mathbf{F}}^f]$	60.83	87.33	95.00	98.17
5	$\bar{\mathbf{F}}^b$	$[\hat{\mathbf{F}}^b, \bar{\mathbf{F}}^f]$	65.16	90.31	95.31	98.59
6	$\bar{\mathbf{F}}^b$	$[\mathbf{F}^b, \mathbf{F}^f]$	68.28	90.94	95.63	98.75
7	$\bar{\mathbf{F}}^b$	$\widetilde{\mathbf{F}}^b \oplus \mathbf{F}^f$	64.53	87.66	92.97	96.72
8	$\bar{\mathbf{F}}^b$	$[\widetilde{\mathbf{F}}^b, \mathbf{F}^f]$	**79.06**	**94.84**	**97.34**	**99.38**
9	$D(\mathbf{I}^c)$		49.22	76.72	84.53	91.88
10	$D(\widetilde{\mathbf{I}}^c)$		**84.38**	**97.03**	**98.91**	**99.84**

Table 2. The ablation studies of our method on S-FOSD dataset. $\bar{\mathbf{F}}^b, \hat{\mathbf{F}}^b$: global/local background feature vector. $\mathbf{F}^b, \widetilde{\mathbf{F}}^b$: global/local background feature map. $\mathbf{F}^f, \bar{\mathbf{F}}^f$: foreground feature map/vector. $[\cdot,\cdot]$: feature concatenation. $\oplus$: feature map composition. The detailed explanations can be found in Section 5.3.

Background Encoder E^b. Based on row 1 of Table 2, we replace the global background feature vector $\bar{\mathbf{F}}^b$ with local background feature vector $\hat{\mathbf{F}}^b$, which is obtained by applying RoIAlign [10] with the crop bounding box B to background feature map $\mathbf{F}^b$, resulting in worse results in row 2. This is probably because that the global background feature contains more useful information than the local background feature. So we adopt the global background feature for background encoder in other experiments.

Composite Image Discriminator D. In our network, the discriminator plays the role of teacher. In row 9 and 10 of Table 2, we feed the discriminator with the whole composite image $\mathbf{I}^c$ and the cropped composite image $\widetilde{\mathbf{I}}^c$, respectively. We see that row 10 achieves significantly better performance than row 9, which can be attributed to the aligned foreground and useful surrounding foreground context in the cropped composite. Then, the discriminator with the whole composite image is inferior to the encoders of row 1, which is roughly consistent with previous findings [44, 42]. Additionally, the discriminator with cropped composite image outperforms the encoders (row 1) with a remarkable margin, which motivates us to build our method.

Knowledge Distillation Module E^d. Based on row 1 of Table 2, we introduce the distillation module without using feature distillation, which is essentially a compatibility classifier $p(\cdot)_y$, in row 3. The classifier takes the concatenation of foreground feature vector $\bar{\mathbf{F}}^f$ and background feature vector $\bar{\mathbf{F}}^b$ as input, and outputs the compatibility score. The comparison between row 1 and 3 verifies that adding compatibility classification is helpful for our task.

Then we add feature distillation and explore the impact of using different forms of background-foreground interactions. In row 4 and 5, we perform interaction based on the pooled feature vectors of two encoders. Specifically, we concatenate the foreground feature vector $\bar{\mathbf{F}}^f$ with global background feature vector $\bar{\mathbf{F}}^b$ (*resp.*, local background feature vector $\hat{\mathbf{F}}^b$) to feed into distillation module in row 4 (*resp.*, row 5). Both results are better than row 3, proving the utility of distilling composite image feature to predict compatibility.

Next, we perform interaction based on the last feature maps of two encoders. To this end, we explore several different ways to fuse background and foreground features. First, we directly concatenate background feature map $\mathbf{F}^b$ and foreground feature map $\mathbf{F}^f$ in row 6. By comparing row 6 and 4, it verifies the advantage of more sufficient interaction. In row 7, we extend to a more intuitive way by applying composition operation to local background feature maps $\widetilde{\mathbf{F}}^b$ and foreground feature map $\mathbf{F}^f$. Specifically, we resize the foreground feature and place it in the query bounding box on the local background feature. Unexpectedly, this leads to slight performance drop compared with row 5, which may be caused by downsampling the foreground feature. Finally, we concatenate the local background feature $\widetilde{\mathbf{F}}^b$ and foreground feature $\mathbf{F}^f$, corresponding to our complete method. As shown in row 8, our full method produces remarkably better results than the above-mentioned row 6, indicating that the representation ability of the distilled feature can benefit from adequate interaction between two encoders.

5.4. Inference Efficiency Analyses

To study the efficiency of our method, we compare our method ("Ours") with the composite image discriminator ("Discriminator"), and background and foreground

Model	Time(200)↓	Time(2000)↓	#Params↓
Discriminator	700.0ms	7000.1ms	20.03M
Encoders	4.0ms	4.3ms	40.05M
Ours	5.2ms	6.0ms	47.13M

Table 3. Comparing inference efficiency of different models, including the discriminator D, encoders $\{E^b, E^f\}$, and our method in the Figure 3. "Time(N)" represents the average time of retrieving compatible foregrounds from N candidates per background.

encoders ("Encoders") on retrieval time and number of parameters, the results of which are reported in the Table 3. We test different models on an NVIDIA RTX 3090 GPU and measure the time of retrieving compatible foreground from N candidates as retrieval time (Time(N)). We repeat the retrieval 50 times and calculate the average time as final result. To mimic the practical application scenarios of foreground object search, we assume that the features of all foregrounds have been extracted and saved before retrieval. In particular, for "Encoders" and "Ours", we save the feature vector and the last feature map of the foreground encoder, respectively. In this way, the retrieval time of "Encoders" and "Ours" only considers the background feature extraction and the matching between background and foreground, in which background feature extraction takes 3.1ms per image.

We can see that the discriminator D is the slowest method due to compositing with each foreground and sending each composite through forward pass. "Ours" runs slower than "Encoders" due to introducing additional interaction between background and foreground, yet only takes 6ms to retrieve from 2000 foregrounds, which enables our model for real-time applications. Regarding the number of parameters, the comparison between two encoders and ours indicates that the additional parameters (7.08M) introduced by distillation module are affordable.

5.5. Additional Experiments in Supplementary

Due to space limitation, we present some experiments in Supplementary, including quantitative comparison on different categories, generalization to new categories, the results of our method using different hyper-parameters in Eqn. (5) and Eqn. (2), and different ratios of positive and negative training samples, the discussion on the limitation of our method, and the results of significance test.

6. Conclusion

In this work, we have contributed two public datasets for Foreground Object Search (FOS), S-FOSD with Synthetic composite images and R-FOSD with Real composite images. We have also proposed a novel FOS method, which improves the general pipeline of previous methods by feature distillation. Extensive experiments on our dataset have demonstrated the utility of our proposed method on foreground object search.

Acknowledgement

The work was supported by the National Natural Science Foundation of China (Grant No. 62076162), the Shanghai Municipal Science and Technology Major/Key Project, China (Grant No. 2021SHZDZX0102, Grant No. 20511100300).

References

[1] Jimmy Ba and Rich Caruana. Do deep nets really need to be deep? In *NIPS*, 2014. 3

[2] Guobin Chen, Wongun Choi, Xiang Yu, Tony X. Han, and Manmohan Chandraker. Learning efficient object detection models with knowledge distillation. In *NIPS*, 2017. 3

[3] Tao Chen, Ming-Ming Cheng, Ping Tan, Ariel Shamir, and Shimin Hu. Sketch2Photo: internet image montage. *ACM SIGGRAPH Asia*, 2009. 2

[4] Wenyan Cong, Li Niu, Jianfu Zhang, Jing Liang, and Liqing Zhang. BargainNet: Background-guided domain translation for image harmonization. In *ICME*, 2021. 1

[5] Wenyan Cong, Xinhao Tao, Li Niu, Jing Liang, Xuesong Gao, Qihao Sun, and Liqing Zhang. High-resolution image harmonization via collaborative dual transformations. In *CVPR*, 2022. 3

[6] Wenyan Cong, Jianfu Zhang, Li Niu, Liu Liu, Zhixin Ling, Weiyuan Li, and Liqing Zhang. DoveNet: Deep image harmonization via domain verification. In *CVPR*, 2020. 1, 3

[7] Xiaodong Cun and Chi-Man Pun. Improving the harmony of the composite image by spatial-separated attention module. *IEEE Transactions on Image Processing*, 29:4759–4771, 2020. 3

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009. 4, 7

[9] Marc-André Gardner, Yannick Hold-Geoffroy, Kalyan Sunkavalli, Christian Gagné, and Jean-François Lalonde. Deep parametric indoor lighting estimation. In *ICCV*, 2019. 3

[10] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42:386–397, 2020. 6, 8

[11] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *ArXiv*, abs/1503.02531, 2015. 3

[12] Yan Hong, Li Niu, Jianfu Zhang, and Liqing Zhang. Shadow generation for composite image in real-world scenes. *AAAI*, 2022. 3

[13] Y. Huang, Jiaxiang Wu, Xingkun Xu, and Shouhong Ding. Evaluation-oriented knowledge distillation for deep face recognition. In *CVPR*, 2022. 3

[14] Eric Kee, James F O'brien, and Hany Farid. Exposing photo manipulation from shading and shadows. *ACM Transactions on Graphics*, 33(5):1–21, 2014. 3

[15] Alina Kuznetsova, Hassan Rom, Neil Gordon Alldrin, Jasper R. R. Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, Tom Duerig, and Vittorio Ferrari. The open images dataset v4. *International Journal of Computer Vision*, 128:1956–1981, 2020. 2, 3

[16] Jean-François Lalonde, Derek Hoiem, Alexei A. Efros, Carsten Rother, John M. Winn, and Antonio Criminisi. Photo clip art. *ACM SIGGRAPH*, 2007. 2

[17] Donghoon Lee, Sifei Liu, Jinwei Gu, Ming-Yu Liu, Ming-Hsuan Yang, and Jan Kautz. Context-aware synthesis and placement of object instances. In *NIPS*, 2018. 3

[18] Anat Levin, Dani Lischinski, and Yair Weiss. A closed form solution to natural image matting. In *CVPR*, 2006. 3

[19] Boren Li, Po-Yu Zhuang, Jian Gu, Mingyang Li, and Ping Tan. Interpretable foreground object search as knowledge distillation. In *ECCV*, 2020. 1, 2, 3, 7

[20] Quanquan Li, Shengying Jin, and Junjie Yan. Mimicking very efficient network for object detection. In *CVPR*, 2017. 3

[21] Jun Ling, Han Xue, Li Song, Rong Xie, and Xiao Gu. Region-aware adaptive instance normalization for image harmonization. In *CVPR*, 2021. 1

[22] Daquan Liu, Chengjiang Long, Hongpan Zhang, Hanning Yu, Xinzhi Dong, and Chunxia Xiao. ARShadowGAN: Shadow generative adversarial network for augmented reality in single light scenes. In *CVPR*, 2020. 3

[23] Liu Liu, Bo Zhang, Jiangtong Li, Li Niu, Qingyang Liu, and Liqing Zhang. OPA: Object placement assessment dataset. *ArXiv*, abs/2107.01889, 2021. 3

[24] Li Niu, Wenyan Cong, Liu Liu, Yan Hong, Bo Zhang, Jing Liang, and Liqing Zhang. Making images real again: A comprehensive survey on deep image composition. *ArXiv*, abs/2106.14490, 2021. 1, 3

[25] Li Niu, Qingyang Liu, Zhenchen Liu, and Jiangtong Li. Fast object placement assessment. *ArXiv*, abs/2205.14280, 2022. 3

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

[27] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. FitNets: Hints for thin deep nets. *CoRR*, abs/1412.6550, 2015. 3

[28] Florian Schroff, Dmitry Kalenichenko, and James Philbin. FaceNet: A unified embedding for face recognition and clustering. In *CVPR*, 2015. 5

[29] Changyong Shu, Yifan Liu, Jianfei Gao, Zheng Yan, and Chunhua Shen. Channel-wise knowledge distillation for dense prediction. In *ICCV*, 2021. 3

[30] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 4, 7

[31] Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor S. Lempitsky. Resolution-robust large mask inpainting with fourier convolutions. In *WACV*, 2022. 3

[32] Fuwen Tan, Crispin Bernier, Benjamin Cohen, Vicente Ordonez, and Connelly Barnes. Where and who? automatic semantic-aware person composition. In *WACV*, 2018. 2

[33] Shashank Tripathi, Siddhartha Chandra, Amit Agrawal, Ambrish Tyagi, James M. Rehg, and Visesh Chari. Learning to generate synthetic data via compositing. In *CVPR*, 2019. 3

[34] Yi-Hsuan Tsai, Xiaohui Shen, Zhe Lin, Kalyan Sunkavalli, Xin Lu, and Ming-Hsuan Yang. Deep image harmonization. In *CVPR*, 2017. 3

[35] Huikai Wu, Shuai Zheng, Junge Zhang, and Kaiqi Huang. GP-GAN: Towards realistic high-resolution image blending. In *ACM-Multimedia*, 2019. 3

[36] Jiajun Wu, Yifan Wang, Tianfan Xue, Xingyuan Sun, Bill Freeman, and Joshua B. Tenenbaum. MarrNet: 3d shape reconstruction via 2.5d sketches. In *NIPS*, 2017. 7

[37] Zongze Wu, Dani Lischinski, and Eli Shechtman. Fine-grained foreground retrieval via teacher-student learning. In *WACV*, 2021. 1, 2, 3, 5, 7, 8

[38] Jiahui Yu, Zhe L. Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S. Huang. Generative image inpainting with contextual attention. In *CVPR*, 2018. 3

[39] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *ICLR*, 2017. 3

[40] Jinsong Zhang, Kalyan Sunkavalli, Yannick Hold-Geoffroy, Sunil Hadap, Jonathan Eisenmann, and Jean-François Lalonde. All-Weather deep outdoor lighting estimation. In *CVPR*, 2019. 3

[41] Zheng Zhang, Chunluan Zhou, and Zhigang Tu. Distilling inter-class distance for semantic segmentation. In *IJCAI*, 2022. 3

[42] Hengshuang Zhao, Xiaohui Shen, Zhe L. Lin, Kalyan Sunkavalli, Brian L. Price, and Jiaya Jia. Compositing-aware image search. In *ECCV*, 2018. 1, 2, 3, 5, 6, 7, 8

[43] Shengyu Zhao, Jonathan Cui, Yilun Sheng, Yue Dong, Xiao Liang, Eric I Chang, and Yan Xu. Large scale image completion via co-modulated generative adversarial networks. In *ICLR*, 2021. 3

[44] Yinan Zhao, Brian L. Price, Scott D. Cohen, and Danna Gurari. Unconstrained foreground object search. In *ICCV*, 2019. 1, 2, 3, 4, 5, 7, 8

[45] Siyuan Zhou, Liu Liu, Li Niu, and Liqing Zhang. Learning object placement via dual-path graph completion. In *ECCV*, 2022. 3

[46] Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A. Efros. Learning a discriminative model for the perception of realism in composite images. In *ICCV*, 2015. 2

[47] Sijie Zhu, Zhe Lin, Scott D. Cohen, Jason Kuen, Zhifei Zhang, and Chen Chen. GALA: Toward geometry-and-lighting-aware object search for compositing. In *ECCV*, 2022. 1, 2, 3, 5, 6, 7, 8