Compositing-aware Image Search

Hengshuang Zhao[1⋆], Xiaohui Shen[2], Zhe Lin[3],
Kalyan Sunkavalli[3], Brian Price[3], Jiaya Jia[1,4]

[1]The Chinese University of Hong Kong, [2]ByteDance AI Lab,
[3]Adobe Research, [4] Tencent Youtu Lab
{hszhao,leojia}@cse.cuhk.edu.hk, shenxiaohui@bytedance.com,
{zlin,sunkaval,bprice}@adobe.com

Abstract. We present a new image search technique that, given a background image, returns compatible foreground objects for image compositing tasks. The compatibility of a foreground object and a background scene depends on various aspects such as semantics, surrounding context, geometry, style and color. However, existing image search techniques measure the similarities on only a few aspects, and may return many results that are not suitable for compositing. Moreover, the importance of each factor may vary for different object categories and image content, making it difficult to manually define the matching criteria. In this paper, we propose to learn feature representations for foreground objects and background scenes respectively, where image content and object category information are jointly encoded during training. As a result, the learned features can adaptively encode the most important compatibility factors. We project the features to a common embedding space, so that the compatibility scores can be easily measured using the cosine similarity, enabling very efficient search. We collect an evaluation set consisting of eight object categories commonly used in compositing tasks, on which we demonstrate that our approach significantly outperforms other search techniques.

1 Introduction

Image compositing is a fundamental task in photo editing and graphic design, in which foreground objects and background scenes from different sources are blended together to generate new composites. While previous work has considered the problem of rendering realistic composites [1–5] when the foreground and background images are given, users often find it challenging and time-consuming to find compatible foreground and background images to begin with.

Specifically, a foreground is considered compatible with the background if they *roughly match* in terms of semantics, viewpoint, style, color, etc., so that realistic composites can be generated with a reasonable amount of subsequent

⋆This work was partly done when H. Zhao was an intern at Adobe Research.

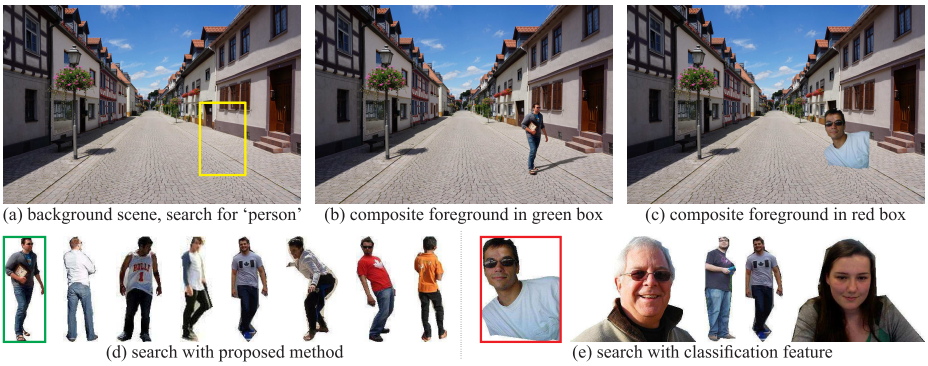


Fig. 1: Compositing-aware image search. Given a background image as a query, the task is to find foreground objects of a certain category that can be composited into the background at a specific location, as indicated by the rectangle.

editing. For example in Fig. 1, a user intends to insert a person standing on the street at the location indicated by the yellow box. With the foreground in the green box, a realistic image can be rendered (Fig. 1(b)) by adjusting the color and adding a shadow. On the other hand, when given an incompatible foreground, it is practically impossible to generate a realistic composite with any editing technique (Fig. 1(c)).

The compatibility of a foreground and background pair can be determined by various aspects, whose importance may vary for different object categories and background scenes. For example, viewpoint is more important when inserting a car on the road, whereas semantic consistency might be more critical when composing a skier with snowy mountains. Existing search techniques usually only focus on one certain aspect, or manually extract features and define the matching criteria [6, 7], which cannot adapt to different object categories and background scenes.

In this paper, we propose a learning based approach for compositing-aware image search. Instead of manually designing the matching criteria or hand engineering features, we learn new feature representations for foreground objects and background images respectively from a large amount of training data, which can adaptively encode the compatibility according to different foreground objects and background scenes. Specifically, we design a two-stream convolutional neural network (CNN) to learn the feature embedding from background images and foreground objects, where the object category information is encoded together with the images through multimodal compact bilinear pooling [8]. Triplets from existing datasets with segmentation mask annotations are constructed to learn a common embedding space, where the compatibility of a foreground and background image can be easily measured using the cosine similarity between their corresponding feature vectors. As a result, efficient search can be performed on huge amounts of foreground assets with existing visual search techniques such as Product Quantization [9]. To make the training more stable from large-scale yet

noisy data, we further develop novel sampling algorithms to expand the triplets by finding additional similar foregrounds.

To evaluate the effectiveness of our proposed algorithm, we collected an evaluation dataset consisting of eight common foreground categories used in image compositing. Our experiments on the evaluation set show that our learned feature representations can adaptively capture the most important factors in terms of compatibility given different background images and foreground categories, and significantly outperforms other search techniques.

2 Related Work

Traditional text-based search paradigms mostly measure the semantic relevance between the text queries and the images, without considering other factors that are important for image compositing, and therefore often return many irrelevant results. Image-based search is often an alternative solution when the search criteria are hard to describe with text. Specific features describing various characteristics such as semantics and appearances [10], styles [11], and spatial layouts [12] are learned to serve different tasks. However, with no suitable foreground images available, it is often ineffective if using the background image as query due to the significant appearance gap between foreground images and background images.

Early efforts on this task such as Photo Clip Art [6] used handcrafted features to find the foreground assets according to several matching criteria such as camera orientation, lighting, resolution and local context. More recently, Tan *et al.* [7] used off-the-shelf deep CNN features to capture local surrounding context particularly for person compositing. However, these approaches lack generality as they only consider limited aspects and cannot adapt to different object categories and background scenes. Moreover, they assume the foreground objects have surrounding background context, and are therefore not feasible on those foreground images with pure background, which is very common in the images in stock sites[1][2] and preferred by users.

Zhu *et al.* [13] trained a discriminative network to estimate the realism of a composite image, which can possibly be used to select compatible foregrounds. However, in compositing tasks, the foregrounds need to be manually adjusted in the end in order to make the final image realistic, as it is very rare, if not impossible, to directly find a foreground perfectly matched with the scene. It is therefore not reliable to determine the realism of a composite image without users in the loop. It is also computationally impractical to try out every foreground candidate from a huge number of assets. Moreover, their trained model mainly considers color compatibility due to their training procedure.

By contrast, benefiting from end-to-end feature learning, our approach is general and adaptive to different object categories and image scenes, and at the same time very efficient on large-scale foreground assets.

[1] https://shutterstock.com
[2] https://stock.adobe.com

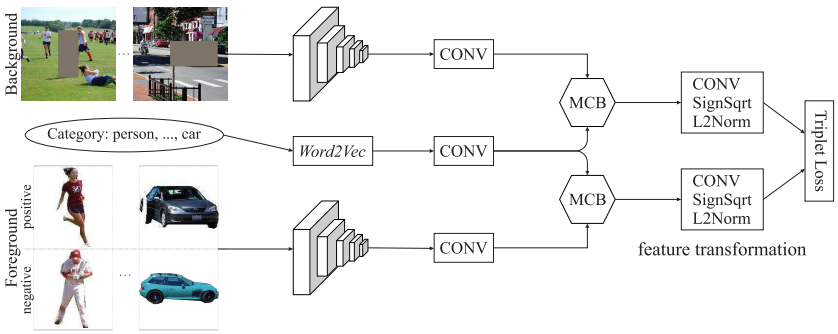


Fig. 2: Overall framework of the proposed *compositing-aware image search* (CAIS) system. A symmetric two-stream feature embedding network is utilized to extract background and foreground image features separately, in which a MCB module is introduced to incorporate category information. A feature transformation module is then performed to generate the final feature representations.

3 Proposed Approach

In this section, we describe the details of our proposed *compositing-aware image search* (CAIS) algorithm. Given a background image, a foreground object category, and the location in the background scene where the foreground will be composed, our task is to return foreground images of that category that are suitable for compositing. As discussed in the introduction, it is difficult to hand-design the matching criteria as the compatibility can be decided by many factors, which may vary in different background scenes and with different object categories. Therefore we aim to learn the feature embedding between the background scenes and foreground assets from a large amount of training data, so that the learned feature representations can encode rich information specifically for image compositing, and can adapt to various image content. Besides, the search algorithm should have the ability to deal with multiple foreground categories in a single framework. In this way, our designed network should be sensitive to category information.

In particular, to deal with this multiclass fine-grained ranking problem, we design a symmetric two-stream network, with each stream taking the background image or foreground object as input respectively, and generating a corresponding feature vector as output. Also, to incorporate the category information, a light weighted word feature extraction branch is added. The image and word features are then fed into a multimodal compact bilinear pooling (MCB) module [8]. MCB has proven to be an effective technique in the context of visual question answering (VQA) in fusing information from multiple modalities with negligible amount of additional parameters. Here we use it to jointly encode the category information and the image content.

During training, we encourage the feature vectors from compatible foreground and background images to be more similar than those from incompatible

pairs. During testing, the learned features can be directly used to calculate the similarity in terms of compatibility for image compositing, which enables efficient large-scale image search. In the following sections, we first introduce our detailed network architecture, and then present our training and sampling strategies that are proved to be of great effectiveness.

3.1 Network Architecture

The architecture of our two-stream feature embedding network is illustrated in Fig. 2. The top stream takes the background scene as input. We use image mean values to fill the rectangle that indicates the location where the object to be inserted, so that the information regarding desired object location, size and aspect ratio can be provided to the network. Meanwhile, the bottom stream takes the foreground image with pure background (*e.g.*, white background) as input. We focus on those pure foreground assets in our work, as they are abundant in those stock image sites and preferred by the users, and at the same time difficult to be retrieved by traditional search techniques.

The search algorithm should have the ability to deal with multiple foreground categories in a single framework. The importance of different factors in determining the compatibility may vary cross different categories. One straightforward solution is training a category-specific network for each category, or in a practically more reasonable design, learning a shared feature encoder and then branching out for each category to learn category-specific features. Nevertheless, neither solution can scale up to many categories, as the number of parameters would linearly increase with the number of class labels. To have a single compact model that can handle multiple categories at the same time, we propose to encode the category information into the foreground and background streams through multimodal compact bilinear pooling (MCB) [8]. During testing, by changing the class label we intend to search, the learned features can adapt to the most important compatibility factors with respect to the object category.

Specifically, to learn the features, we adopt the popular ResNet50 [14] (up to the 'pool5' layer) as our initial weights, after which global average pooling is performed to obtain a feature map of size $1 \times 1 \times 2048$. While the background stream and foreground stream are initialized with the same weights from ResNet50, we expect after learning they can encode different information, with the top stream focuses more on scene context, and the bottom stream learns object-oriented features. To learn the category-specific feature mapping, we use the word2vec [15] model to extract a 300 dimension vector as the input of the word encoding branch. After several convolutional layers, it is then fused with the background and foreground features in each separate MCB modules. A light weighted feature transformation module, including one convolution layer, an element-wise signed square root layer ($y = sign(x)\sqrt{|x|}$) and an instance-wise ℓ_2 normalization operation, is further appended to the network, resulting in a unit feature vector for background and foreground respectively, which encodes both the category information and image content.

3.2 Objective Function

To train the network, we construct triplets consisting of a background image as an anchor, a compatible foreground as the positive sample, and a incompatible foreground as the negative sample. We then adopt triplet loss [16] to train the proposed network and enforce the feature similarity between background anchor and positive foreground to be closer to the one between anchor and negative sample. Since the feature vectors have unit length after ℓ_2 normalization, we can easily calculate their similarity using squared ℓ_2 distance[3]. To encourage the distinguishing ability between positive and negative sample pairs, a positive margin α_i is introduced for class i. For convenience, we group feature extraction, multimodal compact bilinear pooling and ℓ_2 normalization into operation representation $\mathcal{F}$. Thus we want:

$$\|\mathcal{F}_i^b(B_i) - \mathcal{F}_i^f(F_i^p)\|_2^2 + \alpha_i < \|\mathcal{F}_i^b(B_i) - \mathcal{F}_i^f(F_i^n)\|_2^2 \tag{1}$$

where $\mathcal{F}_i^b$ and $\mathcal{F}_i^f$ are operations of category i in background and foreground streams separately. B_i and F_i^p, F_i^n stands for background image and its related positive (*i.e.*, compatible) and negative foreground objects. In the training, we are going to minimize the following loss function $\mathcal{L}$:

$$\begin{aligned}\mathcal{L}(B_i, F_i^p, F_i^n) = max(0, \|\mathcal{F}_i^b(B_i) - \mathcal{F}_i^f(F_i^p)\|_2^2 + \alpha_i \\ -\|\mathcal{F}_i^b(B_i) - \mathcal{F}_i^f(F_i^n)\|_2^2)\end{aligned} \tag{2}$$

3.3 Computational Efficiency

We found that our design is much more effective than sharing all the features across multiple categories, which cannot encode sufficient category-specific information, as demonstrated in Sec. 5. Our solution is also much more computationally efficient than learning separate feature representations dedicated for each category independently, and demonstrate to have very competitive results compared with those individual models. As for running time during testing, it includes feature extraction on input image (14.04ms), MCB module encoding (0.62ms), feature transformation (3.15ms) and similarity calculation (4.32ms with 100 foreground images). Moreover, Product Quantization [9] can be easily used to support real-time retrieval with millions of foreground assets.

4 Training Data Acquisition

To learn a new feature representation for image compositing, it is crucial to have a large amount of training data. However, unfortunately there is no available training set specifically for the compositing-aware image search task. Collecting such a training set also seems impractical, as it is not only very time-consuming to manually label many pairs of background and foreground images, but also

[3] It is equivalent to their cosine similarity as $\|\mathbf{x} - \mathbf{y}\|^2 = 2 - 2\cos(\mathbf{x}, \mathbf{y})$.

(a) image (b) bg (c) fg (d) image (e) bg (f) fg

Fig. 3: Data preparation and augmentation. 'bg' and 'fg' denote background and foreground images respectively.

requires expertise in image compositing and photo editing to decide if the two are compatible. On the other hand, there are several publicly available datasets that contain object instance segmentation masks such as MS-COCO [17], PASCAL VOC 2012 [18] and ADE20K [19]. Utilizing those mask annotations, we can decompose the image into background scenes and foreground objects. Since they are generated from the exact same image, we know for sure that they are compatible, and usually more suitable than any other possible candidate pairs. Therefore, to form a triplet, we can treat the background scene of the image as the anchor, and the foreground from the same image as the positive sample, and then randomly sample a foreground from any other image as the negative sample. In this way, we can generate plenty of triplets for our feature learning.

Specifically, based on these three datasets, we select eight categories that frequently appear and are widely used in image compositing for our task: 'person', 'car', 'boat', 'dog', 'plant', 'bottle', 'chair' and 'painting'. The statistics regarding the training data are listed in the supplementary materials.

Triplet Preparation Given an image with object masks, the process of generating background and foreground samples is illustrated in Fig. 3. During testing, the background scene image does not have the foreground in it. To mimic this situation in training, we obtain the rectangle bounding the foreground based on the mask, and fill in the rectangle with image mean values. It essentially removes the foreground object from the scene. When a user draw a bounding box to indicate the location of object insertion during testing, we can apply the same filling operation to make the training and testing input consistent. To make background images more consistent so that the training is more stable, we crop a square image from the original background, which contains as much context as possible, and place the filled rectangle as close to the image center as possible, as shown in Fig. 3 (b) and (e). As for the foreground sample, we paste the foreground in a square image with pure white background at the center location, as shown in Fig. 3 (c) and (f).

By including the filled rectangle in the background image, the learned background features can respond to the location, size and aspect ratio of the object to be inserted when measuring compatibility. For example, when inserting a person on the lawn, a tall rectangle implies the user may want a standing person, while a wide rectangle may indicate a sitting person. At the same time, such constraint should not be very strict, as the rectangle drawn by the user may

(a) semantic context (b) shape

Fig. 4: Triplet extension. The blue ones are the original foregrounds, while the others are retrieved using (a) semantic context information and (b) shape information, respectively.

(a) background image (b) positive foreground candidates

Fig. 5: An example background image with its labeled positive foreground candidates.

not be very accurate. Motivated by this, we introduce the data augmentation process to relax the size and scale constraints between paired foreground and background images to a limited extent. For background augmentation, we add random padding of the bounding box with maximum possible padding space being half of the bounding box's width and height. The new padded region is filled with mean value as well. Similarly for foreground augmentation, we add random padding and fill in the padded region with white color. For the negative foreground in the triplet, it is randomly chosen from another image with similar augmentation procedure. It would inevitably choose some foreground objects that are actually compatible with the background. However, we argue that the foreground from the same image is still more compatible, and accordingly Eqn. 1 should still suffice. Moreover, as will be presented in the next section, we propose a triplet extension approach to include those foreground images as positive samples, which significantly improves the feature learning performance.

Triplet Extension Paired foregrounds and backgrounds from the same images are easy to harvest, but they are much less than that of negative pairs (e.g, m *vs.* $m(m-1)$ if there are m images in a certain class). The severe imbalance in the number of training samples, coupled with the noise in negative pair sampling where some compatible foregrounds are mistreated as negative samples, makes our feature learning rather difficult. To overcome these limitations, we propose a triplet extension strategy by augmenting with more positive foreground samples.

Given a foreground, we aim to find similar foregrounds using two matching criteria: semantic context and shape information.

For semantic context information, since those foreground images are generated from the ones with background scenes, we can fill in the background of those foreground images with their original background, and then extract semantic features using ResNet50 trained on image classification. Similar foreground are then retrieved by comparing the ℓ_2 distances of the extracted features. We found that such design yields much more consistent results than extracting features on the foreground images with pure background. Some sample retrieval results using the semantic context information on the 'person' category are shown in Fig. 4 (a). For the shape information, we simply calculate the intersection over union (IoU) score of two foreground masks after aligning them around the mask center. Foregrounds with higher IoU scores over masks are considered more similar. Sample retrieval results using this criteria on the 'car' category are shown in Fig. 4 (b).

In practice, we observed that when the objects have more rigid shapes that are more sensitive to viewpoints, shape information is more effective in finding similar foregrounds; while when the objects have more diverse appearance that may vary according to different scenes, using semantic context information produces more consistent results. Based on this observation, we choose to use shape information to augment positive foregrounds for 'bottle', 'car', 'chair' and 'painting', and adopt semantic context information to retrieve similar foregrounds for 'boat', 'dog', 'person' and 'plant'. Given a foreground and its corresponding background from the same image, we retrieve top N similar foreground images, and treat them as compatible foregrounds for the background as well. We found that such triplet extension strategy can largely increase the number of positive training pairs, and meanwhile reduces the noise in negative pair sampling. As a result, it significantly improves the feature learning, as shown in Sec. 5.

5 Experiments

Before presenting experimental results, we describe the implementation details in the following. We carry our experiments on the public platform Caffe [20]. We fix the learning rate as 0.001 for training until model achieves convergence. Momentum and weight decay are set to 0.9 and 0.0001 respectively. Batch size is set to 12 and margin in the triplet loss is set to 0.1. In triplet extension, we use top 10 retrieved foreground images as additional positive foreground samples. For model input, square background and foreground images are resized to 256×256 before being fed into their related feature extraction streams. To ease the training process, we performed two-stage feature learning: first learn the features without the MCB module, thus harvesting the common properties that can be shared across different categories like viewpoint, style and color. Once the model converges, we use the learned network as initialization, and jointly train the model with the MCB and feature transformation module, thus capturing category specific attributes for certain classes like semantics and shape.

Table 1: Ablation study on triplet extension criteria. 'Basic' denotes training without triplet extension.'Semantics' and 'Shape' denote using semantic context and shape information. 'Combine' stands for our combined criteria.

Meth.	boat	bottle	car	chair	dog	paint.	person	plant	mean
Basic	60.66	40.84	28.72	14.18	57.74	27.44	31.69	44.79	38.26
Shape	48.80	44.96	36.37	20.73	42.62	32.48	18.65	41.89	35.81
Semantics	66.16	43.97	29.69	18.36	62.48	28.28	51.25	53.23	44.18
Combine	71.58	42.33	36.71	19.74	62.32	30.95	50.84	51.16	45.70

Table 2: Ablation study on output dimension of the MCB module.

Dim.	No MCB	2048	8192	10240	20480	40960
mean mAP(%)	46.02	46.17	46.46	47.18	48.42	47.91

5.1 Evaluation Set and Metric

While the image compositing task as a whole requires a lot of components including various editing and blending operations, in this paper we mainly address the first step in the task, i.e., finding compatible foreground assets given a background image. In order to make the evaluation focus on this step, we created an evaluation set composed of background images and compatible/incompatible foreground objects. Specifically, given a background image and a location where the object is going to be inserted, we insert every possible foreground candidate at that location to generate the composite, and label the foreground compatible only when it is possible to make the composite realistic with some basic image editing operations. Some labeled compatible foreground images for the background in Fig. 5 (a) are shown in Fig. 5 (b).

The evaluation set contains the eight object categories we selected for this task as mentioned in Section 4. Each category has 10 background images with various scenes. We draw a bounding box on each of the background image in appropriate position that is suitable for object insertion. For candidate foreground images, we utilize object instance masks from validation sets of MS-COCO, VOC 2012 and ADE20K. Each category has 100∼400 candidate foreground objects, with 223 candidates on average. For ground truth, background images in each category has 16∼140 compatible foreground candidates.

Intuitively, given a background image, a good search algorithm should rank all the compatible foregrounds higher than others. It naturally leads to adopting *Mean Average Precision* (MAP) as our evaluation metric, which is commonly used in image retrieval. We average the MAPs of all the 10 testing samples for each category to obtain category-wise MAP, and also report the mean evaluation results by averaging the results over all the categories. The MAP scores shown in the tables are all in percentage.

Table 3: Ablation study on network structures.

Method	boat	bottle	car	chair	dog	paint.	person	plant	mean
Separate modules	69.65	49.71	42.93	22.57	62.00	34.72	54.75	53.17	48.69
Ours	71.04	55.00	39.84	18.97	65.45	34.09	51.14	51.83	48.42

Table 4: Comparison with other search methods. 'RealismCNN.' stands for the method in [13]. 'Shape' and 'Classification' denote searching using shape features and classification features.

Method	boat	bottle	car	chair	dog	paint.	person	plant	mean
RealismCNN	46.81	49.05	15.56	08.60	50.12	27.37	21.48	37.48	32.06
Shape	46.12	39.08	34.77	11.54	44.77	26.43	15.25	43.09	32.63
Classification	63.30	**55.51**	14.93	11.03	45.90	23.96	33.48	46.10	36.78
Ours	**71.04**	55.00	**39.84**	**18.97**	**65.45**	**34.09**	**51.14**	**51.83**	**48.42**

5.2 Ablation Study

Triplet Extension We first perform ablation study on different triplet extension criteria. To focus more on the effects caused by triplet sampling, the study is conducted in the first-stage feature learning, i.e., when learning the shared features without MCB. Results are listed in Table 1. We can see using shape information alone in triplet extension in fact made the results worse, possibly because many irrelevant foreground images are returned for categories such as 'person' and 'dog', making the training data even noisier. With semantic context, the results are significantly improved, demonstrating the importance of triplet extension. Finally, our combined strategy yields the best results, outperforming the 'Basic' method by 7.44% in absolute difference and 19.45% in relative improvement.

Network Structure We also did the ablation study on the output dimension of the MCB module, as shown in Table 2. "No MCB" means the network without the MCB module. Therefore the network is shared across different categories, with no category information is encoded. The ones with the MCB module obtain better performance, which demonstrates the effectiveness of encoding category information and learning category-adaptive features. The performance improves when the dimension increase, and is saturated after the dimension reaches 20480. Therefore we set the dimension to be 20480 in the subsequent experiments. Also note that training in one stage is less stable and converges poorer than the two stage solution (mean MAP 44.65% *vs.* 48.42% in two stage training).

We further investigate different network designs on feature learning in dealing with multiple object categories. As mentioned in Section 3, one straightforward solution to handle multiple categories is learning a shared feature encoder and

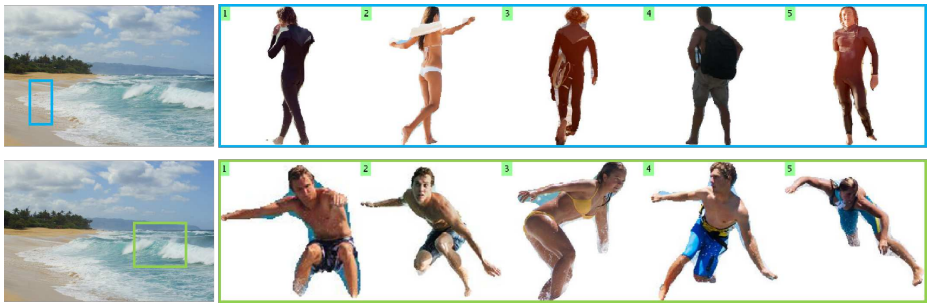


Fig. 6: Our search results are tuned to location and aspect ratio of the bounding box.

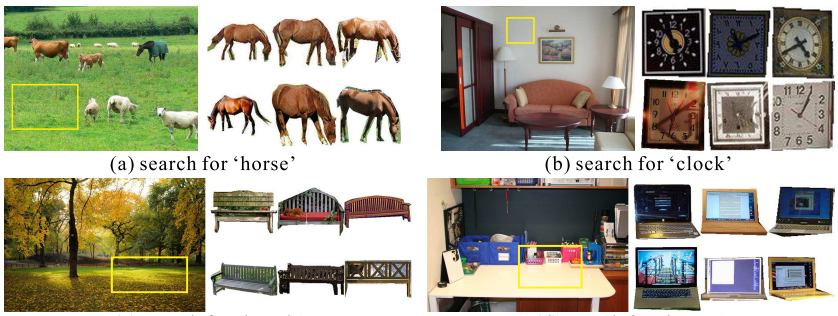
then learning category-specific feature mapping for each category separately. In our implementation, we keep the shared ResNet50 backbone model, remove the MCB module, and learn an individual feature transformation module for each of the eight categories. The results are reported in Table 3. While it obtains good performance, it comes with much more parameters, and is not feasible with larger number of categories, as we need to train each separate branch for every class. Our adopted solution shown in the second row has very similar performance while being much more compact.

5.3 Comparison with Other Search Methods

We compare our proposed CAIS approach with three baseline methods: RealismCNN [13], Shape feature and Classification feature. The rectangle drawn in the background image indicate desired size and aspect ratio by the user. Therefore we can match the drawn rectangle with the rectangle bounding the foreground object by calculating the IoU score of the two rectangles after aligning them around center position. We denote this baseline method by search with 'Shape feature'. In addition, we can also use semantic features learned through image classification, which are commonly used in image-based visual search, to retrieve foreground. For RealismCNN, we generate composite images by fitting the foreground candidates into the drawn rectangle in the background image together with Poisson blending [1], and use the realism score predicted by the RealismCNN to rank all the candidates. The results of these three baseline search methods as well as ours are shown in Table 4. Our approach significantly outperforms all the other methods. It is 11.64% higher than the second best one in term of absolute difference and 31.65% better in terms of relative improvement.

The visual search results are shown in Fig. 9, from which we can see our method accounts for different factors and returns more compatible foreground objects. Moreover, our learned features can consider the location and aspect ratio of user-drawn rectangles, and return suitable foregrounds accordingly, as shown in Fig. 6. More examples are in our supplementary materials.

Fig. 7: Sample results of Poisson blending that are adopted in user study.



(a) search for 'horse' (b) search for 'clock'

(c) search for 'bench' (d) search for 'laptop'

Fig. 8: Generalization to new categories.

5.4 User Study

To further evaluation the search results in terms of the compositing quality, we performed a user study to compare the composites generated by our retrieved foreground objects and the ones generated by the foregrounds that are retrieved using the classification feature, which performs best among the three baseline methods in quantitative evaluation. Poisson blending [1] is used to blend the images and reduce boundary artifacts. Some sample results are shown in Fig. 7.

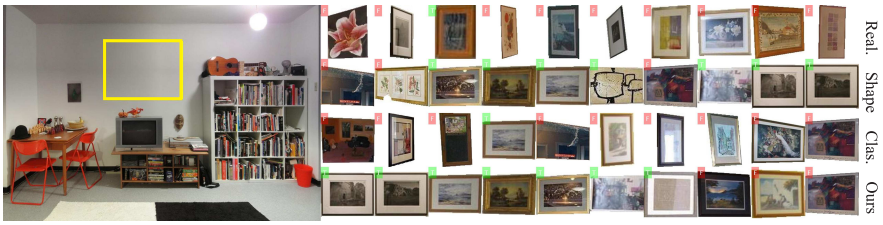
We randomly selected 20 background images from our evaluation set, and use the top retrieved foreground by each method to generate the composites. In the study, the participants are asked to choose the results they think are more realistic. Overall we have 30 subjects participate the study. On average, 70.38% composites with foregrounds retrieved by the proposed method were rated more realistic than those searched by classification feature.

5.5 Generalization to New Categories

To further exhibit the representation ability of our learned shared feature across multiple classes, we test our method on new categories that have not been trained. The search results are illustrated in Fig. 8. Even without training on the new classes, the algorithm still works reasonably well. Interestingly, the retrieved clocks are all in rectangular shape, mostly because of the bias induced

(a) Compositing-aware image search for 'person'

(b) Compositing-aware image search for 'car'



(c) Compositing-aware image search for 'painting'

Fig. 9: Visual search results. In each example, the yellow box indicates the position of foreground object to be inserted. The 1st to the 4th rows show the retrieved results using RealismCNN, shape information, classification features and our approach, respectively. The text boxes with 'green' and 'red' color in the top left corner of the foregrounds represent 'positive' and 'negative' foregrounds respectively. Our returned results contain more compatible foregrounds for image compositing.

from the 'painting' category during training. Our method can easily scale up to much more categories if new training data are available, as the category information can be incorporated through the word feature branch, while the network architecture would still remain the same.

6 Concluding Remarks

In this paper, we present a general compositing-aware image search algorithm that aims on large-scale foreground assets for image compositing. Our proposed novel training and sampling strategies facilitate the feature embedding between background scenes and foreground objects, and thus enable efficient and accurate search with light online computation. We further show the learned feature representations can generalize to new categories and used for other search scenarios.

References

1. Pérez, P., Gangnet, M., Blake, A.: Poisson image editing. In: ACM Transactions on graphics (TOG). (2003)
2. Sun, J., Jia, J., Tang, C.K., Shum, H.Y.: Poisson matting. In: ACM Transactions on Graphics (TOG). (2004)
3. Sunkavalli, K., Johnson, M.K., Matusik, W., Pfister, H.: Multi-scale image harmonization. In: ACM Transactions on Graphics (TOG). (2010)
4. Xue, S., Agarwala, A., Dorsey, J., Rushmeier, H.: Understanding and improving the realism of image composites. In: ACM Transactions on Graphics (TOG). (2012)
5. Tsai, Y.H., Shen, X., Lin, Z., Sunkavalli, K., Lu, X., Yang, M.H.: Deep image harmonization. In: CVPR. (2017)
6. Lalonde, J.F., Hoiem, D., Efros, A.A., Rother, C., Winn, J., Criminisi, A.: Photo clip art. In: ACM transactions on graphics (TOG). (2007)
7. Tan, F., Bernier, C., Cohen, B., Ordonez, V., Barnes, C.: Where and who? automatic semantic-aware person composition. In: WACV. (2018)
8. Fukui, A., Park, D.H., Yang, D., Rohrbach, A., Darrell, T., Rohrbach, M.: Multimodal compact bilinear pooling for visual question answering and visual grounding. In: EMNLP. (2016)
9. Jegou, H., Douze, M., Schmid, C.: Product quantization for nearest neighbor search. TPAMI (2011)
10. Gordo, A., Almazan, J., Revaud, J., Larlus, D.: Deep image retrieval: Learning global representations for image search. In: ECCV. (2016)
11. Collomosse, J., Bui, T., Wilber, M., Fang, C., Jin, H.: Sketching with style: Visual search with sketches and aesthetic context. In: ICCV. (2017)
12. Mai, L., Jin, H., Lin, Z., Fang, C., Brandt, J., Liu, F.: Spatial-semantic image search by visual feature synthesis. In: CVPR. (2017)
13. Zhu, J.Y., Krähenbühl, P., Shechtman, E., Efros, A.A.: Learning a discriminative model for the perception of realism in composite images. In: ICCV. (2015)
14. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. (2016)
15. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv:1301.3781 (2013)
16. Schroff, F., Kalenichenko, D., Philbin, J.: Facenet: A unified embedding for face recognition and clustering. In: CVPR. (2015)
17. Lin, T., Maire, M., Belongie, S.J., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: ECCV. (2014)
18. Everingham, M., Gool, L.J.V., Williams, C.K.I., Winn, J.M., Zisserman, A.: The pascal visual object classes VOC challenge. IJCV (2010)
19. Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., Torralba, A.: Scene parsing through ade20k dataset. In: CVPR. (2017)
20. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R.B., Guadarrama, S., Darrell, T.: Caffe: Convolutional architecture for fast feature embedding. In: ACM MM. (2014)