Where and Who? Automatic Semantic-Aware Person Composition

Fuwen Tan [*1], Crispin Bernier [1], Benjamin Cohen [1], Vicente Ordonez [1], and Connelly Barnes [1,2]

[1]University of Virginia, [2]Adobe Research

Abstract

Image compositing is a method used to generate realistic yet fake imagery by inserting contents from one image to another. Previous work in compositing has focused on improving appearance compatibility of a user selected foreground segment and a background image (i.e. color and illumination consistency). In this work, we instead develop a fully automated compositing model that additionally learns to select and transform compatible foreground segments from a large collection given only an input image background. To simplify the task, we restrict our problem by focusing on human instance composition, because human segments exhibit strong correlations with their background and because of the availability of large annotated data. We develop a novel branching Convolutional Neural Network (CNN) that jointly predicts candidate person locations given a background image. We then use pre-trained deep feature representations to retrieve person instances from a large segment database. Experimental results show that our model can generate composite images that look visually convincing. We also develop a user interface to demonstrate the potential application of our method.

1. Introduction

Image compositing aims to produce images that can trick humans into believing they are real, although they are not. Image composites can also result in fantastic images that are limited only by an artist's imagination. However, the process of creating composite images is challenging, and it is not fully understood how to make realistic composites. A typical compositing task proceeds in four steps: (1) choose a foreground segment that is semantically compatible with a given background scene; (2) place the segment at a proper location with the right size; (3) perform operations such as alpha matting [25] or Poisson blending [21] to adjust the local appearance; (4) apply global refinements such as relighting or harmonization [30]. The first two steps require semantic reasoning while the last two steps deal with appearance compatibility. Whether a human perceives a composite image as real or fake depends on all these factors. However, while existing compositing systems tackle the last two steps automatically, most of them leave the semantic tasks (steps 1 and 2) to the users.

In this work, we explore the semantic relationships between a collection of foreground segments and background scenes using a data-driven method for automatic composition. We restrict the foreground category to "human" because humans play a central role in a large proportion of image composites, and because we can easily collect enough exemplar data for training and testing. For simplicity, we also choose to ignore occlusion by assuming that human segments are fully visible from the camera viewpoint.

This research is motivated by recent breakthroughs in scene recognition [31] and object-level reasoning [22] through deep neural networks, which have brought unprecedented levels of performance for similar semantic tasks. Thus, we apply these techniques to estimate the semantic compatibility between candidate foreground segments and image backgrounds using a large scale visual dataset. Given these observations, our method contains three components: First, a location proposal step where we predict the location and size of each potential person instance using a novel Convolutional Neural Network (CNN) architecture. Second, a retrieval step, where we find a specific segment that semantically matches the local and global context of the scene. Lastly, a final compositing step, where we leverage off-the-shelf alpha matting [6] to adjust the transition between a composited segment and its surroundings so that the segment appears compatible with the background.

To evaluate our method, we conduct quantitative and qualitative experiments including a user study. We demonstrate that our generation pipeline can be useful for interactive layout design or storyboarding tasks which can not be easily fulfilled using other tools.

We summarize here our technical contributions: (1) A model that predicts probable locations for the presence of a person instance for an input background using contex-

*fuwen.tan@virginia.edu

tual cues. (3) A fully automatic person compositing system which generates convincing composite images. To the best of our knowledge, this is the first attempt towards this task; (4) Conducting quantitative and qualitative evaluations, including a user study and a proof-of-concept user interface.

2. Related work

Composite image generation. Early methods for compositing such as alpha matting [25] and gradient-domain compositing [21] can seamlessly stitch a foreground object with a background image by blending a local transition region. To enforce global appearance compatibility, Lalonde and Efros [17] proposed to model the co-occurrence probability of the foreground object and the background image using a color distribution. Similarly, Xue et al. [34] proposed to investigate the key statistical properties that control the realism of an image composite. Recently, Zhu et al. [37] trained a single CNN-based model to distinguish composite images from natural photographs and refine them by optimizing the predicted scores. Furthermore, Tsai et al. [30] developed an end-to-end deep CNN based model for image harmonization. These methods give visually pleasing results, but unlike our work, they all leave the semantic tasks to the users, such as choosing foreground segments and placing them at proper positions with the right size.

The work of Lalonde et al. [18] took a step further by building an interactive system to insert new objects into existing photographs by querying a vast image-based object library. Chen et al. [7] developed a similar interactive system but took user sketches as input. Hays et al. [10] proposed an automatic patch retrieval and blending method for scene completion using millions of photographs. Unlike our work, these methods relied on hand-crafted features and the composite regions were still indicated manually by the users.

Context based scene reasoning. Using context for scene reasoning has a long history [8]. Pioneering works include Bar and Ullman [26] and Strat and Fischler [2], which incorporated contextual information for recognition. Context based methods are also popular in object-level classification. Bell et al. [4] proposed a Recurrent Neural Network framework to detect objects in context. These works modeled correlations among contents within the image, while our method predicts contents that are not yet present. Related to our work, Torralba et al. [28] introduced a challenge to test to what extent can object detection succeed by only contextual cues. More recently, Sun et al. [27] proposed a siamese network to detect missing objects in an image. While these methods predicted contents that were not present in the images, they all focused on the binary determination of whether there should be any object at a specific location or not. In contrast, our method attempts to predict both the location and size of a potential foreground segment, and retrieve a segment with proper appearance that is compatible with the surrounding context. Concurrently to our research, Wang et al. [33] proposed to model affordances by predicting the skeletons of persons that were not already present. This method achieves good results, but requires matching of a similar indoor scene, and only predicts a skeleton, not a full color composite. Finally, Kermani et al. [15] synthesized 3D scenes by learning factor graph and arrangement models from an RGB-D dataset.

Context based image editing. By conditioning on the local surroundings, Pathak et al. [20] performed semantic inpainting by using a generative adversarial network. Yang et al. [35] proposed a multi-scale CNN model for high-resolution image inpainting via neural patch synthesis. Iizuka et al. [13] developed a CNN based method for image completion by enforcing global and local consistency. Recently, Chen et al. [5] presented a cascaded refinement network to synthesize images conditioned on semantic layouts. While these methods synthesize novel contents from context at pixel level, the locations or layouts of the synthesized regions were still provided by the users. Our method predicts such regions and retrieves plausible segments.

3. Overview

Figure 1 shows an overview of our system. It has three main components: bounding box prediction, person segment retrieval, and compositing. We now give a brief discussion of each of these.

In Section 4, we introduce our proposed CNN based model to predict a bounding box of the potential segment. We formulate the bounding box prediction as a joint classification problem by discretizing the spatial and size domains of the bounding box space. Specifically, we design a novel two branch network which can be trained end-to-end using supervised learning, and tested in a cascade manner.

In Section 5, we introduce a candidate pool we built for segment retrieval. A context based segment retrieval scheme is devised to find a person segment from the candidate pool that semantically matches both local context and the global scene. The key component for achieving this is a hybrid deep feature representation. Finally, we use an alpha matting technique to composite the retrieved segment with the background at the predicted location and size.

In Section 6, we evaluate our bounding box prediction model quantitatively by measuring the histogram correlation between the ground truth bounding boxes distribution and our prediction. We also evaluate the visual realism of composite images with a human subject evaluation.

4. Bounding box prediction

In this section, we introduce our learning based method to predict the bounding box of a potential person segment given a single background image. Our key insight here is

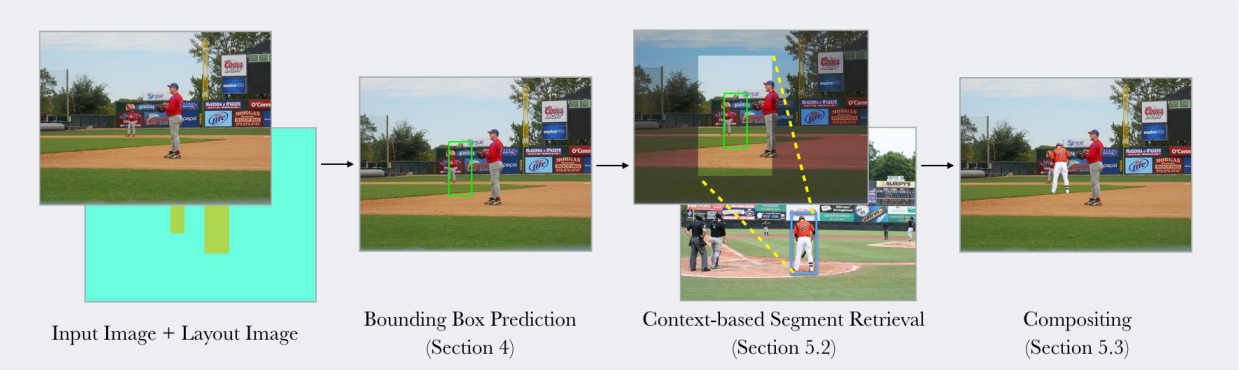

Figure 1: Overview of our pipeline: our system consists of three computational stages, which are depicted above.

that the correlation between the foreground segment and the background scene can be learned directly from human-annotated object layouts of natural images.

We first discuss how we collect and preprocess the data (Section 4.1). Next, we explain the input for the model (Section 4.2), and give the prediction target for the model (Section 4.3). We then give the model itself (Section 4.4).

4.1. Data preprocessing

The data we use for learning such layout correlation is from MS-COCO [19]. This dataset contains tens of thousands of images with both bounding box and segment annotations for each object instance in 80 different categories.

Because a large proportion of object instances are occluded, we automatically filter out heavily occluded person instances using three passes of filtering: (1) We filter the person instances whose bounding boxes have large overlapping areas with other objects. Specifically, we exclude instances whose Intersection over Union (IoU) with any other instance is larger than 0.3. (2) We also exclude person instances that are close to the edge of the images as they are probably incomplete. In particular, we filter the instance if the distance between its bounding box and the edge of the image is less than 18 pixels. (3) Finally, we remove instances whose areas are less than 2500 square pixels.

After applying the filtering routines, we obtain 36,636 person instances from the training split of MS-COCO, and 16,962 from the validation split.

4.2. Input imagery

For each person instance in the dataset, we attempt to learn the mapping from its background context to the person's bounding box. Learning such a mapping function requires us having an input image in which the person is not already present. However, to do this, we have to "erase" the person instances from the source images. Our solution is to remove the person instances automatically by using the human-annotated segments from MS-COCO. We remove each person via the inpainting method of Barnes et al. [3], implemented as Content Aware Fill from Adobe Photoshop. The resulting inpainted results sometimes exhibit artifacts such as repetitive patches. To prevent the model from over-fitting on these artifacts, the inpainted image is further blurred using a Gaussian with a sigma of 3.2. We denote the blurred image as I_B.

Given the recent breakthroughs in CNN-based object detection systems [22, 4], in addition to using our inpainted (and blurred) images directly as input, we also incorporate the informative output from an object detector. We use the Faster RCNN object detector [22] to obtain object detections in the inpainted images. The bounding boxes of the detected objects in different categories are then rendered using a randomly generated color palette, with each color corresponding to a category. The color values within an overlapping region are set to the mean color value. We find that using different color palettes achieves similar performance. The layout image (indicated as I_L) represents the object layout of the image, as shown in Figure 1.

4.3. Prediction target

The target of our prediction model is the bounding box of a person. We first discuss how we represent the bounding box using normalized coordinates, and then explain how we discretize these coordinates for use in classification.

The bounding box representation from a ground truth annotation in the dataset is a four dimensional vector: $(x_{\min}, y_{\min}, x_{\max}, y_{\max})$, where $(x_{\min}, y_{\min})$ represents the top-left coordinate and $(x_{\max}, y_{\max})$ represents the bottom-right coordinate. For images of different resolutions, a normalized bounding box representation is required for consistent prediction. To do this, our system pads each rectangular image by the minimum amount so a square image is obtained, using a padding color that is the mean color for the ImageNet dataset [23]. The bounding box is first shifted to account for the square padding, then transformed into normalized coordinates $(x_{\text{stand}}, y_{\text{stand}}, w, h) \in [0, 1]$, where $x_{\text{stand}} = \frac{1}{2s}(x_{\min} + x_{\max})$, $y_{\text{stand}} = \frac{1}{s} y_{\max}$, s is the width of the square image, and w, h are the width and height of

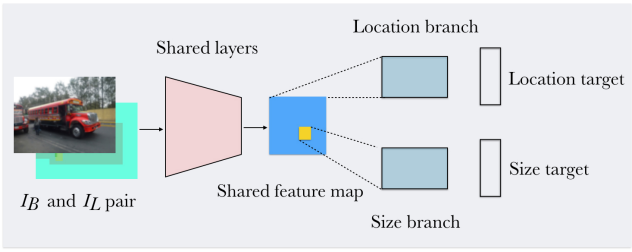

Figure 2: Overview of the prediction model: the proposed network exploits a two branch architecture: the first branch predicts location and the second branch predicts size.

the box relative to the square image. Thus, $(x_{\text{stand}}, y_{\text{stand}})$ is the lowest center (standing) point of the bounding box.

Direct regression in a four dimensional continuous space is challenging. To facilitate the bounding box prediction, our system discretizes the (x, y) location domain into a 15×15 grid board, and then represents $(x_{\text{stand}}, y_{\text{stand}})$ as the index of the grid g_{xy} where it is located. Similarly, the (w, h) size domain is also discretized so that (w, h) is represented as another grid index g_{wh}. By doing this, we formulate the bounding box prediction as two classification problems with 225 (15 x 15) different classes for each.

4.4. Prediction model

Given I_B, I_L as inputs and g_{xy}, g_{wh} as targets, our next challenge is to learn the underlying mapping between them. Our approach is to learn the location (g_{xy}) and size (g_{wh}) simultaneously as they are highly correlated. In particular, we develop a novel CNN-based model which can be trained in an end-to-end manner.

In our model, the images (I_B, I_L) are first concatenated along the depth channel, and fed through a shared front-end network, as shown in Figure 2. This network is shared in the sense that the same weights are used before the split into the location and size branches. The shared network contains three residual bottleneck modules with projection shortcuts, similarly as in He et al. [11]. Starting from the output feature map of the shared network, the model is then separated into two smaller branches, with the first branch predicting the location (g_{xy}), and the second branch predicting the size (g_{wh}). These two branches also incorporate dilated convolutional layers introduced in [36] in order to use larger receptive fields without using an additional number of parameters. Table 1 lists the layer-by-layer details of the proposed network architecture.

Note that the size of the predicted box should be consistent with the local context. For instance, person segments should not be larger than instances of larger objects appearing in their surroundings, such as buses or cars. Therefore, in the size prediction branch, after a small 3 x 3 dilated convolution, our system first remaps the normalized coordinates $(x_{\text{stand}}, y_{\text{stand}})$ into the spatial coordinates of the output activations, to obtain grid coordinates $(x_{\text{grid}}, y_{\text{grid}})$. A (512 x 3 x 3) activation slice is then extracted along the depth channel. This is done by extracting activations from a box with 3x3 spatial size, such that the lowest center coordinate of the box is $(x_{\text{grid}}, y_{\text{grid}})$. We call this process Region of Interest (ROI) slicing. This smaller activation map is then fed through the rest of the layers of the size branch. By doing this, the size prediction network attends to a sub-region of the feature map that captures the local context.

One subtle point for this design is that, during training, the normalized coordinates $(x_{\text{stand}}, y_{\text{stand}})$ we use for ROI slicing come from the ground truth bounding box. However, during testing, $(x_{\text{stand}}, y_{\text{stand}})$ are generated from the location we predict. Therefore, during inference our network runs in two stages: the first stage predicts only location; the second stage predicts the size based on the location. Please see our supplemental for additional implementation details.

Layers	Activation Size
Shared layers	
Input	6 x 480 x 480
Conv: 64 x 7 x 7, stride 2	64 x 237 x 237
Max pool 3 x 3, stride 2	64 x 118 x 118
Conv block, (64, 64, 128) filters*	128 x 59 x 59
Conv block, (64, 64, 128) filters*	128 x 30 x 30
Conv block, (128, 128, 512) filters*	512 x 15 x 15
Location prediction branch	
Conv: 64 x 3 x 3, dilation 2	64 x 15 x 15
Conv: 1 x 3 x 3, dilation 2	15 x 15
Size prediction branch	
Conv: 512 x 3 x 3, dilation 2	512 x 15 x 15
ROI slicing	512 x 3 x 3
Global maxpooling	512
Two fully connected layers	225

Table 1: Architecture of our prediction model.
* The last layer has stride 2 and a projection shortcut [11].

5. Person segment retrieval and compositing.

In this section, we introduce a simple context-based person segment retrieval and compositing scheme based on a hybrid deep feature representation. We first discuss how we create the pool of candidate person segments and perform retrieval. Then we describe how we perform compositing.

5.1. Creating the candidate pool of person segments

To build a candidate pool for person segment retrieval, we use the annotated data from the validation split of the MS-COCO dataset. We chose this split because these images are also held out from the training of the bounding box prediction. We apply the same filtering routines as in Section 4.1 to exclude segments that are heavily occluded, small or incomplete. Finally, we manually filter the remaining segments to remove partially occluded instances. In total, our candidate pool contains 4100 person segments.

Figure 3: Person segment retrieval: given the input image (A) and the predicted bounding box (the green box in (A)), the proposed system incorporates features from both the global scene and the local context (covered by the yellow box in (A)) to retrieve a favorable person segment (within the yellow box in (B)) and composite it on the input image.

Although these segments come with ground truth segmentation annotations, most of the annotations are not accurate enough for compositing applications. Therefore, we also perform manual segmentation using the lasso tool from Adobe Photoshop for the segments we present and used in the human subject study (see Section 6).

To demonstrate the generalization of our system, all the background testing images we present in the paper are from the YFCC100M split of the VisualGenome dataset [16] and the SUN dataset [32].

5.2. Context based person segment retrieval

Given a background image and a predicted bounding box, our goal is to retrieve a person segment from the candidate pool that not only matches the global scene semantics but also appears compatible with the local context. Various hand-crafted feature descriptors [29][12] have been proposed to facilitate image retrieval. Recently, using intermediate neural network activations as feature representations has shown to perform competitively for various semantic retrieval tasks even when the underlying network has been pre-trained in an unrelated classification task [1]. However, previous methods mostly aim to retrieve images that "look similar" with respect to a query image, while our goal is to retrieve segments which are not present but "look natural" when composited on a background scene.

Our key insight here is that, by incorporating the contextual information of both the query background image and the candidate person segments, we could adapt and extend feature-based methods to retrieve segments from images which share similar global scene semantics and local context with the background image. Specifically, for each input image, our system first extracts deep features which describe global scene semantics of the background image. We adopt the activation map from the mean pooling layer of ResNet50 [11]. Similarly, for each candidate person segment, we extract the same feature descriptor for its background image. Measuring the distance between the input image and the candidate images in feature space can help retrieve segments appearing in similar scenes. However, the retrieved segment does not necessarily look natural in the local context if only global compatibility is considered.

To further enforce the local compatibility, given the predicted bounding box, our system crops a local image patch which shares the same center with the bounding box but is twice as large in both width and height, as shown in Figure 3. The same feature descriptor (activations of the mean pooling layer of ResNet50) of this local patch is then extracted. For each candidate person segment, our system extracts similar local feature descriptors. Measuring the distance between these local features can help retrieve segments appearing in similar local contexts as in the target location.

In our implementation, the segment retrieval proceeds in two steps: (1) Our system first filters the segments whose bounding box sizes are quite different from the query box size. To do this, our system aligns the centers of the query and target bounding boxes and computes their Intersection over Union (IoU). Segments with IoUs smaller than 0.4 are excluded. (2) From the remaining candidate segments in our collection, the system retrieves the top one segment that is "closest" to the query input in feature space. Specifically, we use cosine distance between the query and the target segment, each represented by a concatenation of the global and local feature descriptors. To accelerate the retrieval process, we also build a kd-tree structure of the candidate segments.

5.2.1 Selection of features

For the retrieval task, we experimented with a few different feature descriptors (e.g. GIST feature [29], unsupervised learned feature from the Context Encoder work [20], deep activation maps from VGG16 [24] and ResNet50). We adopted the mean pooling feature from ResNet50 based on several observations: (1) Compared with GIST and context encoder features [20], the superiority of deep features was demonstrated by a pilot user study we conducted on Amazon Mechanical Turk. The setup of the pilot study was similar with the one we are going to introduce in Section 6.2. (2) Different from VGG16, ResNet50 incorporated Batch Normalization layers [14], which produce activation maps with similar magnitude scales for different dimensions. This is important when measuring the distance between features. Because the feature maps from VGG16 exhibited various magnitude scales for different dimensions, our experiments showed that they usually resulted in poor retrieved segments. (3) Compared with other layers in ResNet50, the activation map from the mean pooling layer encodes much semantic information (it is one layer before the final classifier) in smaller dimensions (2048), which makes it both effective and efficient for our retrieval task.

5.3. Compositing

With the retrieved segment in hand, our system scales and composites it onto the background such that the seg-

Figure 4: Composites automatically generated from our system. The first row shows the input images, the second row shows the composite results. We include additional results in the supplemental.

ment has the same center and height as the predicted bounding box. Although the segment already has a clean binary mask produced from the Photoshop magnetic lasso tool we discussed in Section 5.1, we apply an off-the-shelf alpha matting method [6] to obtain smooth natural transitions over the composite region. Figure 4 shows example composites produced by our method covering various scenes. As our current pipeline has not considered relighting explicitly, the composites may suffer from lighting inconsistency problems. We leave relighting to future work.

6. Evaluation

6.1. Quantitative evaluation of box prediction

During training of the bounding box prediction model, we use the ground truth bounding boxes as the target for supervised learning. At first glance, it may seem reasonable to use evaluation metrics from object detection systems, such as average precision or precision-recall (PR) curve. However, for each specific background image, there may be multiple locations suitable for composing person instances with various sizes. The goal of the prediction model is to learn the distribution of feasible object layouts instead of overfitting toward the exact ground truth boxes in the dataset. In fact, we try to avoid this situation by blurring the input image so that the system can not overfit to inpainting artifacts.

Therefore, to evaluate the performance of the bounding box prediction model, we measure the correlation between the distributions of the predicted boxes and the ground truth boxes. In particular, we represent the distribution of bounding boxes as two 2D histograms: A position histogram for the $(x_{\text{stand}}, y_{\text{stand}})$ coordinates, and a size histogram for the (w, h) sizes. The bin sizes we use for histograms are 15×15, the same as for the prediction model.

For this experiment, the ground truth bounding boxes we use are from the validation split of the MS-COCO dataset, which are held out from the training stage. The generated boxes are predicted from the same set of images but with the person segments erased and inpainted. To measure the histogram correlation, we use the metric:

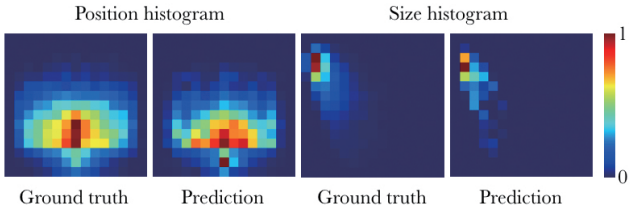

Figure 5: Ground truth bounding box statistics and statistics measured from our prediction model. For the position distribution, the correlation between ground truth and prediction is 0.9458. For the size distribution, the correlation between ground truth and prediction is 0.9378.

$$d(A,B) = \frac{\sum_i (A_i - \overline{A})(B_i - \overline{B})}{\sqrt{\sum_i (A_i - \overline{A})^2 \sum_i (B_i - \overline{B})^2}} \quad (1)$$

where A and B represent the histograms of the ground truth and the predictions respectively, $\overline{A}$ and $\overline{B}$ are means of A and B, and N is the bin count, which is 225 in our case.

Under this proposed metric, the correlation between the ground truth and the prediction is 0.9458 for the position histograms, and 0.9378 for the size histograms. As judged by these high correlation scores, our prediction model can mimic real person layouts in natural images. Figure 5 shows the 2D histograms we use for this evaluation.

In Figure 6, we also visualize example heatmaps of the predicted locations (the softmax layer of the location prediction network). We can see that although our network is trained to predict a fixed unique location, it can approximate the location distribution reasonably well. We include additional heatmaps of predicted locations in the supplemental.

6.2. Qualitative evaluation via user study

To evaluate the visual realism of the composite images, we conduct a human subject study using Amazon Mechanical Turk (AMT). For purposes of comparison, three strong baseline methods are also evaluated, as described below.

- **Baseline 1:** the bounding box is sampled from the ground truth distribution, and the segment is retrieved using our segment retrieval method. This allows us to

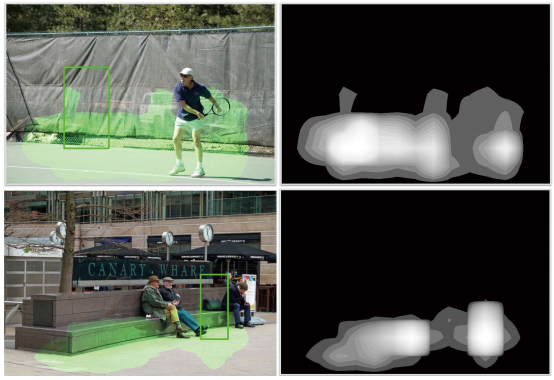

Figure 6: Example heatmaps of predicted locations. The green boxes show top 1 bounding boxes from our system.

Model	Percent real for textured	Percent real for silhouette
Baseline 1	0.176 ± 0.054	0.402 ± 0.100
Baseline 2	0.200 ± 0.059	0.442 ± 0.117
Baseline 3	0.276 ± 0.064	0.505 ± 0.109
Top 1	0.440 ± 0.078	0.567 ± 0.110
Best of top 8	0.517 ± 0.075	0.742 ± 0.107
Real	0.898 ± 0.041	0.864 ± 0.075

Table 2: Results from the user study. Shown are mean and standard deviations for the percent of images marked real.

determine the impact of our bounding box predictions;

- **Baseline 2:** the bounding box is predicted by our system, then a segment is randomly sampled from the candidate pool. The purpose of this baseline is to evaluate the impact of our segment retrieval method;
- **Baseline 3:** the bounding box is predicted by our system but the segment is retrieved using a global GIST feature [29] under Euclidean distance, resembling the work of [10]. This allows us to evaluate the effect of deep feature representations on this problem.

For our method, we evaluate the top 1 composite from our system. We also include a manually chosen "best" images of the top 8 outputs based on the following criteria: combinations of the top 2 location predictions, top 2 size predictions and top 2 retrieved segments. For each background image we evaluate five composite images. Additionally, we believe that future work with larger retrieval datasets or better relighting algorithms could potentially improve results. Thus, to assess the effects of texture and lighting, we also constructed "silhouette" images where the person's matte is simply filled with a uniform white color.

6.2.1 User study setup

During the study, the participants were presented with a sequence of images and told to press R if an image appears real or F if it appears fake. For the silhouettes we calibrated users by showing them what "real" silhouettes look like. For each image, the user had to respond in 10 seconds, otherwise the data was ignored. To avoid interference effects, we showed each participant examples from only one method. For each model, we evaluate 80 composite images. Composite images for different models share the same set of backgrounds. For quality control, we also included equal numbers of real images and obviously fake composites. We discarded responses from users who obtained less than 80% accuracy on the quality control. For textured images, we collected 25 opinions per image, whereas for the silhouette images, we collected 11 opinions.

6.2.2 Quantitative results

Table 2 shows the mean "realism" scores of each image. Standard errors for the scores were computed by applying bootstrapping to the means. For the textured images, we notice that both the top 1 and "best" of top 8 composites outperform all baseline methods. The "best" of top 8 composites performs slightly better than top 1. However, there is still a performance gap between the "best" of top 8 composites and real images. One explanation is that our current system has not considered shadows and lighting explicitly. For the silhouette images, the performances are in the same order but with higher scores of realism. In particular, the score of the "best of top 8" composites is much closer to that of the real images. This indicates that texture and lighting cues are more frequently responsible for "giving away" that a composite is not real, as opposed to location, size, and silhouette cues, which give results similar to real images.

For the mean scores, we also tested for significance using a two-sided Student's t-test. The Holm-Bonferroni method was adopted to control the familywise error rate at the significance level of 0.05. For the textured images, our method is significantly better than the baselines ($p < 0.0002$). However, top 1 and "best" of top 8 are not significantly different ($p = 0.0972$). For the silhouette images, the best of top 8 method is significantly better than the baseline methods and top 1 ($p < 0.00005$). Top 1 is significantly better than baselines 1 and 2 ($p < 0.0076$), but not significantly different than baseline 3 ($p = 0.2$). For textured and silhouette, real is significantly better than all other models ($p < 0.00002$).

7. Prototype user interface

We have also developed a proof-of-concept user interface for composite image generation and interactive layout refinement. Existing compositing tools typically require intensive user interactions, such as finding compatible foreground and background pairs, and finding suitable locations and sizes for composition. We allow users to create and refine composite images with automatic guidance and less manual searching for segments, positions, and sizes.

In our current interface, given an input image, the user is asked how many people should be composited on the back-

Figure 7: Examples of the comparison for different methods: (A) input images; (B) baseline 1; (C) baseline 2; (D) baseline 3; (E) top 1; (F) "best" of top 8. Note that for the last row, the top 1 composite is also the "best" of the top 8 composites.

Figure 8: In our proof-of-concept user interface, given the input image in column (A), users are asked how many people to add to the scene. Column (B) shows results of our automatic compositing. Column (C) shows the refinement results of (B) via user interactions.

ground. The top 1 automatic composite is then returned. For each predicted bounding box, 9 candidate segments are also displayed. The user can then refine the composite by replacing, translating or scaling each person segment. Figure 8 shows example results of automatic compositing and user refinement. Please refer to the supplementary video for more such examples.

8. Limitations and conclusion

Our current compositing system, however effective, still has a few limitations. (1) Although there is no underlying assumption, the outputs from our system tends to bias towards similar positions (e.g. the central region of the image) with similar focal lengths. As all our training and testing images are from standard datasets, we conjecture that the similarity is from the datasets, as people tend to appear in the central regions in natural images. (2) Our bounding box prediction model depends on the performance of an object detector. There are situations where the results of the detection may hinder the predictions of our model. (3) While combining the global and local contexts helps retrieving segments that are compatible with the background, the retrieved segments still may not "interact" correctly with each object in the scene. Figure 9 shows two examples when such interactions are important. (4) Our system may potentially benefit from recent success of Generative Adversarial Nets [9] by building an end-to-end matting system with adversarial loss. (5) Our system has not explicitly integrated lighting and shadow consistency with the background. Global relighting methods such as [30] may further improve photo-realism. (6) Our system has not considered categories other than person and is not trained end-to-end.

Figure 9: Limitations of our system. In the first example (top row), though our system retrieves a "sitting" person segment, it does not align well with the background chair. In the second example (bottom row), the system correctly retrieves a tennis player, but the action is wrong.

In conclusion, we propose a fully automatic system for semantic-aware person composition. The system accomplishes compositing by first predicting the bounding box of the potential instance and then retrieving a segment that appears compatible with the local context and global scene appearance. Quantitative and qualitative evaluations show that our system predicts person layouts for a given background scene and outperforms robust baselines.

Acknowledgement

We thank our anonymous reviewers for helpful feedback. Thanks to the photographers for licensing photos under Creative Commons in the COCO, YFCC100M, VisualGenome and SUN datasets. This project was partially funded by the

NSF grants HCC 1011444. Vicente Ordonez is partially funded by an IBM Faculty Award and an NVIDIA GPU Grant.

References

[1] A. Babenko, A. Slesarev, A. Chigorin, and V. Lempitsky. Neural codes for image retrieval. *European Conference on Computer Vision (ECCV)*, 2014.

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

[3] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman. PatchMatch: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics (SIGGRAPH)*, 28(3), Aug. 2009.

[4] S. Bell, C. L. Zitnick, K. Bala, and R. Girshick. Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[5] Q. Chen and V. Koltun. Photographic image synthesis with cascaded refinement networks. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[6] Q. Chen, D. Li, and C.-K. Tang. Knn matting. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 35(9):2175–2188, Sept 2013.

[7] T. Chen, M.-M. Cheng, P. Tan, A. Shamir, and S.-M. Hu. Sketch2photo: Internet image montage. *ACM Transactions on Graphics (SIGGRAPH)*, 28(5):124:1–124:10, Dec. 2009.

[8] S. K. Divvala, D. Hoiem, J. H. Hays, A. A. Efros, and M. Hebert. An empirical study of context in object detection. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[9] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems 27*, pages 2672–2680. 2014.

[10] J. Hays and A. A. Efros. Scene completion using millions of photographs. *ACM Transactions on Graphics (SIGGRAPH)*, 26(3), 2007.

[11] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[12] R. Hu and J. Collomosse. A performance evaluation of gradient field hog descriptor for sketch based image retrieval. *Comput. Vis. Image Underst.*, 117(7):790–806, July 2013.

[13] S. Iizuka, E. Simo-Serra, and H. Ishikawa. Globally and Locally Consistent Image Completion. *ACM Transactions on Graphics (SIGGRAPH)*, 36(4):107:1–107:14, 2017.

[14] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on International Conference on Machine Learning (ICML)*, pages 448–456, 2015.

[15] Z. S. Kermani, Z. Liao, P. Tan, and H. Zhang. Learning 3d scene synthesis from annotated rgb-d images. In *Computer Graphics Forum*, volume 35, pages 197–206. Wiley Online Library, 2016.

[16] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, M. S. Bernstein, and L. Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123(1):32–73, May 2017.

[17] J.-F. Lalonde and A. A. Efros. Using color compatibility for assessing image realism. *IEEE International Conference on Computer Vision (ICCV)*, 2007.

[18] J.-F. Lalonde, D. Hoiem, A. A. Efros, C. Rother, J. Winn, and A. Criminisi. Photo clip art. *ACM Transactions on Graphics (SIGGRAPH)*, 26(3):3, August 2007.

[19] T. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: Common objects in context. *European Conference on Computer Vision (ECCV)*, 2014.

[20] D. Pathak, P. Krähenbühl, J. Donahue, T. Darrell, and A. Efros. Context encoders: Feature learning by inpainting. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[21] P. Pérez, M. Gangnet, and A. Blake. Poisson image editing. *ACM Transactions on Graphics (SIGGRAPH)*, pages 313–318, 2003.

[22] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems (NIPS)*, pages 91–99, 2015.

[23] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[24] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[25] A. R. Smith and J. F. Blinn. Blue screen matting. *ACM Transactions on Graphics (SIGGRAPH)*, pages 259–268, 1996.

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

[27] J. Sun and D. Jacobs. Seeing what is not there: Learning context to determine where objects are missing. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[28] A. Torralba. Contextual priming for object detection. *International Journal of Computer Vision (IJCV)*, 53(2):169–191, 2003.

[29] A. Torralba, K. P. Murphy, W. T. Freeman, and M. A. Rubin. Context-based vision system for place and object recognition. *IEEE International Conference on Computer Vision (ICCV)*, 2003.

[30] Y.-H. Tsai, X. Shen, Z. Lin, K. Sunkavalli, X. Lu, and M.-H. Yang. Deep image harmonization. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[31] J. Xiao, K. A. Ehinger, J. Hays, A. Torralba, and A. Oliva. Sun database: Exploring a large collection of scene categories. *International Journal of Computer Vision (IJCV)*, 119(1):3–22, 2016.

[32] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3485–3492, June 2010.

[33] R. G. Xiaolong Wang and A. Gupta. Binge watching: Scaling affordance learning from sitcoms. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[34] S. Xue, A. Agarwala, J. Dorsey, and H. Rushmeier. Understanding and improving the realism of image composites. *ACM Transactions on Graphics (SIGGRAPH)*, 31(4):84:1–84:10, 2012.

[35] C. Yang, X. Lu, Z. Lin, E. Shechtman, O. Wang, and H. Li. High-resolution image inpainting using multi-scale neural patch synthesis. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[36] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. *International Conference on Learning Representations (ICLR)*, 2016.

[37] J.-Y. Zhu, P. Krähenbühl, E. Shechtman, and A. A. Efros. Learning a discriminative model for the perception of realism in composite images. *IEEE International Conference on Computer Vision (ICCV)*, 2015.