SeGAN: Segmenting and Generating the Invisible

Kiana Ehsani[1], Roozbeh Mottaghi[2], Ali Farhadi[1,2]
[1] University of Washington, [2] Allen Institute for AI (AI2)

Abstract

Objects often occlude each other in scenes; Inferring their appearance beyond their visible parts plays an important role in scene understanding, depth estimation, object interaction and manipulation. In this paper, we study the challenging problem of completing the appearance of occluded objects. Doing so requires knowing which pixels to paint (segmenting the invisible parts of objects) and what color to paint them (generating the invisible parts). Our proposed novel solution, SeGAN, jointly optimizes for both segmentation and generation of the invisible parts of objects. Our experimental results show that: (a) SeGAN can learn to generate the appearance of the occluded parts of objects; (b) SeGAN outperforms state-of-the-art segmentation baselines for the invisible parts of objects; (c) trained on synthetic photo realistic images, SeGAN can reliably segment natural images; (d) by reasoning about occluder-occludee relations, our method can infer depth layering.

1. Introduction

Humans have strong ability to make inferences about the appearance of the invisible and occluded parts of scenes [1, 34]. For example, when we look at the scene depicted in Figure 1 we can make predictions about what is behind the coffee table, and can even complete the sofa based on the visible parts of the sofa, the coffee table, and what we know in general about sofas and coffee tables and how they occlude each other. Devising algorithms to infer the appearance of what is behind an object exhibits several challenges. Predicting the appearance of the occluded regions of objects requires reasoning over multiple intertwined cues. Recognizing if an object is occluded or not is the first challenge to begin with. Second, knowing what pixels to color requires extending the boundaries of objects from their visible regions to invisible parts which requires some form of knowledge about the shapes of objects. The complex relations between the appearance of objects and the change in viewpoint and occlusion patterns form the third challenge. Deformable objects can even make the problem ill-defined. Fourth, it is challenging to provide large-scale, accurate, and reliable training data to train models for occlusion reasoning.

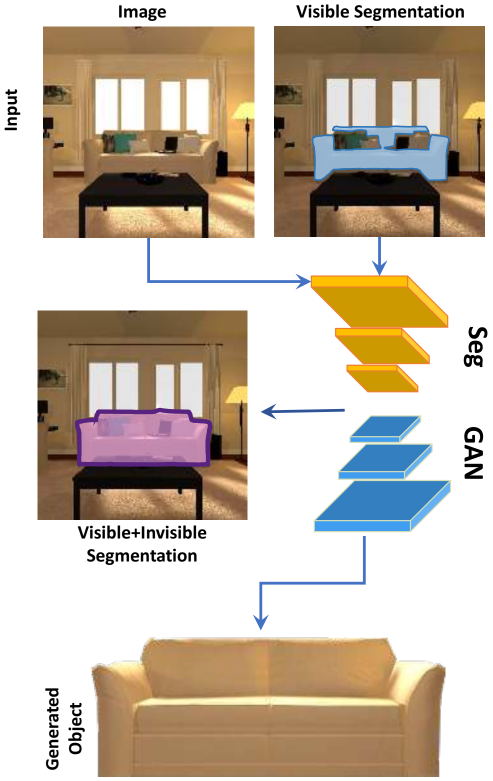

Figure 1. Our goal is to jointly segment and paint the invisible regions of objects. For instance, we predict how the sofa looks like when the occluders (cushions, laptop and coffee table) are removed. The input to our model is an image and a mask for the visible regions of objects (shown in blue).

In this paper, we study the problem of generating the invisible parts of objects. Doing so requires segmenting the invisible part of the object and then generating the appearance of (painting) it. Inspired by the principles of learning-the-easy-thing-first, we propose SeGAN, a novel model that combines segmentation and generation and jointly optimizes for both of them. More specifically, given an input image and a segmentation for the visible regions of an object, our proposed GAN-based model learns to predict a segmentation for the occluded regions and generate the appearance by painting the invisible parts. Using segmentation masks of the invisible part as our intermediate step enables our network to learn about what pixels to paint before painting them. The generator network then paints the selected pixels. By jointly learning segmentation and generation networks SeGAN learns about the interdependencies between objects, their occlusion patterns, the shape and appearance of object segments. This allows us to address the first three challenges.

The key remaining challenge is training data; where can we find large-scale and accurate training data for what is behind the visible part of images? We argue that the proposed solution for Amodal segmentation in [55] is not suitable for our approach. Human judgements for predictions about the invisible parts of objects is subjective. Also, superimposing segments of images over other images [27] would result in unnatural occlusion boundaries. In this paper, we propose to use photo-realistic synthetic data to learn how to extend segmentation masks from the visible parts of objects to the invisible regions and how to generate the appearance of the invisible part. Doing so allows us to obtain large-scale and accurate training data for the invisible regions of objects in images.

Our experiments show that SeGAN can, in fact, segment and generate the invisible regions of objects. Our results also show that our proposed segmentation network can learn to segment the occluded regions of objects and outperforms various state of the art segmentation baselines. We also show that our segmentation network can reliably segment the invisible parts of objects in natural images, when trained on our photo-realistic training set. By reasoning about occlusion patterns, our model can also make predictions about occluder-occludee relationships resulting in depth ordering inferences. Note that SeGAN is category-agnostic and does not require semantic category information.

2. Related Works

There is a large body of work on object detection [14, 13, 18, 41, 42, 45], semantic segmentation [31, 2, 3, 28, 36, 53, 7, 35, 30, 24] and instance segmentation [39, 26, 8, 6, 40, 51, 52] using deep learning. These methods are designed for the visible regions of objects and they are not able to capture occlusions or provide a depth ordering for objects in an image. In contrast, our goal is to reconstruct occluded regions.

Occlusion reasoning has been studied in the literature extensively. [47] propose a CRF for segmenting partially occluded objects. [44] infer occlusion edges of polygons that represent objects. [11] make DPM more robust to occlusion by inferring whether a cell inside the object bounding box belongs to the object or not. [16] use scene priors to infer the label for the occluded background regions. [49] propose a layered object detection and segmentation method, where the goal is to infer depth ordering for the detected objects. [20] propose an occlusion model for object instance detection based on 3D interaction of objects. [15, 12] propose methods for detection and pose estimation of occluded people. [38] learn occluder-occludee patterns to improve object detectors. [21] synthesize scenes by retrieving segments from training images, which requires reasoning about depth layers in the scene. [46] provide a semantic label for each pixel in an image along with the occlusion ordering for objects. [5] use top-down information to tackle occlusions in multi-instance segmentation. We differ from all of these methods in that we complete the segmentation mask for the occluded objects and generate the appearance for the occluded regions of each object instance. Also, we show transfer from synthetic to natural images.

The problem of bounding box completion has been tackled by [23], where the goal is to find the full extent of the object bounding box. Amodal segmentation methods have been proposed by [55, 27], where they aim to provide a complete mask for occluded objects. The annotations that [55] provide is mainly based on the subjective judgment of the annotators (since the occluded parts of objects are not visible). In contrast, we modify our scenes by removing occluders and obtain an accurate groundtruth mask and texture for the occluded objects. The groundtruth annotation of [27] is obtained by pasting an object over an arbitrary image. Our argument is that occlusion relationships are not arbitrary and follow certain characteristics, and the way that we collect our occlusion data enables us to better model the occlusion relationships. Also, in contrast to these methods, we generate the appearance for the occluded regions.

Conditional Generative Adversarial Networks (cGANs) [33] have been used for different applications such as prediction of future frames [32], style transfer [25], colorizing and synthesizing images from edge maps [22], etc. Image inpainting using cGANs and DCGANs has been explored by [37] and [50]. In this paper, we combine cGANs with a convolutional network to segment and paint the occluded regions of objects simultaneously. Our problem is different from inpainting since our goal is to paint regions outside the input mask.

Recently, [4] proposed a regression-based approach to synthesize images from a given semantic segmentation layout. Our method differ from [4] since their goal is not to reconstruct occluded regions. Also, our method performs both segmentation and painting and it is category-agnostic.

3. Model

Our goal is to segment and paint the occluded regions of objects. The inputs to our model are a segmentation mask for the visible (non-occluded) regions of an object and an RGB image. The output is an RGB image where the oc-

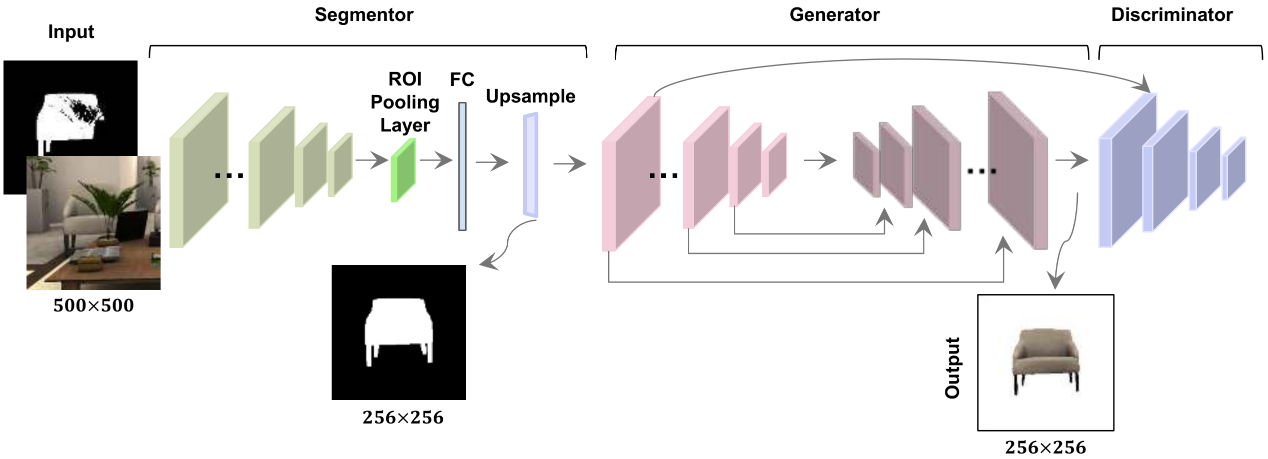

Figure 2. **Model architecture.** Our network has three parts: *segmentor*, *generator*, and *discriminator*. The input to our model is an RGB image and a mask for the visible region of an object which is obtained automatically by [51]. The output is an RGB image that shows the appearance and segmentation for the full object (visible and reconstructed invisible regions). The segmentor part outputs an intermediate mask (the mask shown in the middle) that represents the full object, which is passed to the generator part of the network.

cluded regions of that object have been reconstructed. The segmentation masks for visible regions can be obtained automatically from any instance segmentation method (e.g., [51]).

We introduce SeGAN that infers the mask for the occluded regions and paints those regions in a joint fashion. Our model has two main parts: (1) segmentation and (2) painting. The *segmentation* part provides a mask for the occluded and non-occluded regions of objects, which is fed into the painting part of the model. The *painting* part generates the appearance for the occluded region of the object. These two parts of the network are trained jointly. The architecture of the model is shown in Figure 2.

The segmentation part of the network is a CNN that takes a four-channel tensor as input, where three channels correspond to the RGB image, and there is a single channel for the segmentation mask of the visible region of an object. The mask for the visible region is obtained automatically (refer to Section 5 for details). The idea is to use the information from visible regions to segment and paint the invisible regions. We modify ResNet-18 [19] to generate a mask image as output (the output of the last convolutional layer). Then, the mask output is fed into an ROI pooling layer. The ROI pooling layer is followed by a fully connected layer with the output size of 3364 (58×58), and refer to its output by o. An upsampling layer converts o to 256×256. We denote the output of the upsampling layer by O. Our final result is more accurate when we use upsampling.

The painting part of the network generates the invisible or occluded regions of the object. This part is a conditional generative adversarial network (cGAN) [33], which consists of a generator and a discriminator.

The input to the generator, M, is computed as follows:

$$M(I,O,V) = I \odot V + R \odot (O - V) + B \odot (J - O - V), \quad (1)$$

where $\odot$ is element-wise multiplication, I and V are the input RGB image and input binary visible mask ($\texttt{SV}$), J is an all-one matrix of size 256×256, and R and B are 256×256 images, where their first and third channels are 1s, respectively, and the rest of their channels are 0s. All of the binary masks in the above equation are repeated three times to form a 3 channel image. Basically, in the generator's input, the mask for the invisible region (which is provided by the segmentation part of the network) is red, and the region outside the mask is blue.

We adopt Unet [43] for the generator network, which is an encoder-decoder with skip connections from encoders to the corresponding layers in the decoder. The discriminator network includes four convolutional layers, followed by one sigmoid layer. The architecture for this part is similar to that of the Pix2Pix network [22].

The loss function for our model is a combination of the losses for segmentation and painting. For segmentation, we define a customized loss function using binary cross entropy loss that is computed on the prediction of the network and the groundtruth for the full object binary mask (referred to as $\texttt{SF}$). In Section 4, we explain how we obtain accurate groundtruth for the occluded regions. Ideally, the segmentation part should learn 1) not to change the mask for the pixels in $\texttt{SV}$ (mask for visible regions) and 2) to predict the mask for the pixels in $\texttt{SI}$ (mask for invisible regions) correctly. The binary cross entropy loss is defined as:

$$L_{ent}^{S}(g,o) = -\frac{1}{n}\sum_{ij\in S}(g_{ij}log(o_{ij})+(1-g_{ij})log(1-o_{ij})), \tag{2}$$

where S is a subset of pixels (e.g., pixels of the visible region), g_{ij} and o_{ij} are pixels at location (i,j) of the groundtruth $\mathtt{SF}$ and predicted mask, respectively, and $n = |S|$.

Our loss function for segmentation is defined as:

$$L_{segm}(g,o) = \lambda_{bg}L_{ent}^{\overline{SF}}(g,o)+\lambda_{SV}L_{ent}^{SV}(g,o)+\lambda_{SI}L_{ent}^{SI}(g,o), \tag{3}$$

where $\overline{SF}$ is the set of pixels in the image patch not in SF, or in other words the pixels that do not belong to either visible or invisible parts of the object. A sigmoid function is applied to the predicted output so we obtain a real number between 0 and 1. The intuition for defining this objective is to differentiate among making mistakes in segmenting the visible region, invisible region and the background.

The loss function for painting is defined as follows:

$$\begin{aligned} L_{cGAN}(G,D) &= E_{x\sim p_{data}(x),z\sim p_z(z)}[\log D(G(x,z))] \\ &+ E_{x\sim p_{data}(x),z\sim p_z(z)}[\log(1-D(x,G(x,z))], \end{aligned} \tag{4}$$

where G and D are the generator and the discriminator networks, respectively, x is the input and z is a random Gaussian noise vector, which is mainly used for regularizing the generator network.

Previous approaches found L1 and L2 distance losses to be helpful for GANs [37, 22], thus the final loss function for the adversarial part is defined as:

$$L^{*}(G) = L_{cGAN}(G,D) + \lambda L_{L1}(G) \tag{5}$$

The loss function for our SeGAN end-to-end model, L_{full}, is defined as:

$$L_{full} = \lambda_{L^{*}}L^{*}(G) + L_{segm}(g,o) \tag{6}$$

4. Dataset

In this paper, we introduce DYCE, a dataset of synthetic occluded objects. This is a synthetic dataset with photo-realistic images and natural configuration of objects in scenes. All of the images of this dataset are taken in indoor scenes. The annotations for each image contain the segmentation mask for the visible and invisible regions of objects. The images are obtained by taking snapshots from our 3D synthetic scenes. A few examples of images and their annotations are shown in Figure 3.

There are two advantages of a synthetic 3D dataset. First, we can obtain a 2D dataset of the desired size, and there is no restriction over the number of training samples we can generate. Second, we can move the camera to any location to capture interesting patterns of occlusion. We use the scenes of [54] to generate our dataset.

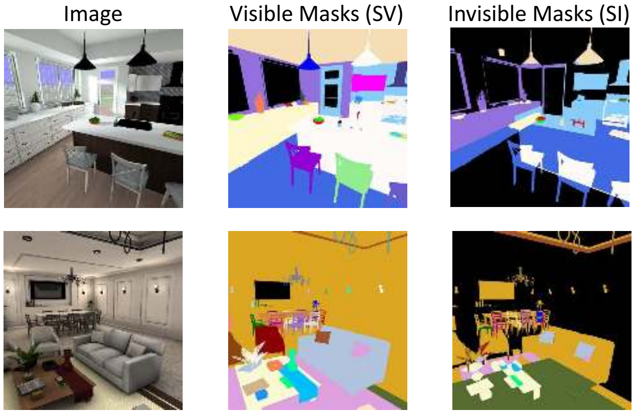

Figure 3. **Example images of the dataset.** The first column shows the images captured from 3D synthetic scenes. The second column shows the segmentation mask for the visible regions. Each instance is encoded by a different color. The third column shows the invisible regions. For example, in the second row, the cushions occlude the sofa. Therefore, the regions behind the cushions have grey color in the third column, which means that those pixels belong to the grey sofa in the second column.

4.1. Generating 2D Images from 3D Scenes

For generating the images, we change the location and the viewpoint of the camera in order to get a variety of images. For each scene, we generate 500 images from different viewpoints of the camera. We restrict the areas that the camera can be located. We move the camera in locations that the head of a person can be located in order to obtain common patterns of occlusion that people observe. We also restrict the orientation of the camera such that the camera points to objects in the scenes. Otherwise, the dataset will contain many images with no objects (for example, images depicting a portion of a wall).

The procedure for generating the segmentation mask for the visible and invisible regions of objects is as follows. For each object, we generate an image with all other objects removed. Then, we compare this image with the original image, where no object is removed from the scene. The pixels that are the same in both images are the visible pixels of this particular object. To obtain the mask for the invisible region, we subtract the mask of the visible region from the mask of the full object.

4.2. Statistics

The number of the synthetic scenes that we use is 11, where we use 7 scenes for training and validation, and 4 scenes for testing. Overall there are 5 living rooms and 6 kitchens, where 2 living rooms and 2 kitchen are used for testing. On average, each scene contains 60 objects and the number of visible objects per image is 17.5 (by visible we mean having at least 10 visible pixels). There is no common object instance in train and test scenes.

5. Experiments

Our model performs segmentation and painting jointly. Hence, in this section, we evaluate our model from these two perspectives. In addition, we show results of generalization to natural images. Finally, we present our evaluation for the depth layering task. Our training and test sets include 41924 and 27617 objects depicted in 3500 and 2000 images, respectively.

5.1. Implementation details

The segmentation part is initialized by the weights of ResNet-18 [19] that are pre-trained on ImageNet [9]. We use random initialization for the painting part of the network.

All input images and their masks are resized to 500×500. We used bilinear interpolation for resizing. Thus, the segmentation mask might contain values in the interval $(0, 1)$. To obtain bounding boxes for the ROI pooling layer we expand the box around the input SV masks by a random ratio between 10-30% from each side. Note that we ignore the portions that lie outside the image. We compute the segmentation loss on groundtruth segmentation masks of size 58×58 (for each object, we crop the image using the expanded bounding box and scale the cropped image to 58×58). Then, we upsample the predicted mask to 256×256 using a bilinear upsampling layer and use the 256×256 mask as the input to the painter network. We do not train the upsampling layer. The generator outputs a three channel 256×256 image, which includes the RGB values for the full object (invisible and visible regions).

We use the following coefficients in the loss function: $\lambda_{bg} = 1, \lambda_{SV} = 5, \lambda_{SI} = 3$, $\lambda_{L1} = 100$, and $\lambda_{L^*} = 0.1$. These values are obtained using a validation set. Also, to help the network to converge, we first train the segmentation network and the generator network jointly and then train the whole network end to end.

5.2. Evaluation

Segmetation & Painting. We evaluate our model, SeGAN, in two settings. First, we use the output of the Multipath network [51], which is a state-of-the-art model for generating the segmentation mask for objects as our input mask for the visible regions (SV masks). Secondly, to factor out the effects of SV segmentation approach from our results, we also show the results using the groundtruth mask as the input for the visible region of the object.

After obtaining segmentation masks from Multipath, we find the segmentation mask that corresponds to the visible region of the groundtruth training object. The segmentation mask that has the largest intersection over union with the visible region of the groundtruth mask is selected as the input mask during training. For evaluation, we consider all masks generated by Multipath.

We evaluate our model using three metrics for segmentation and two for painting. For segmentation, we evaluate how well we predict (1) the mask for the occluded regions (SI), (2) the mask for non-occluded regions (SV), and (3) the mask for the full object (SF=SV $\cup$ SI). The intuition for evaluating the mask for visible regions is to check whether our approach distorts the input mask when the object is not occluded. For all of these settings, we compute intersection over union between the predicted mask of the model and the groundtruth mask. For evaluating painting, we use L1 and L2 distance of the predicted output and the ground truth image.

Table 1 summarizes our results for segmentation. First, our method (referred to as 'SeGAN w/ predicted SV') significantly outperforms Multipath for the task of predicting masks for the full object (SF) and invisible regions (SI). It is interesting to see that our method improves the segmentation of the visible regions (SV) as well. We have two variations of the Multipath network as our baselines: one trained only on natural images data (trained on MS COCO dataset [29]) and one trained on the combination of natural images data and our synthetic images (trained to predict occluded and visible regions). Another baseline is IBBE [27] (an amodal segmentation method), which is fine-tuned on our synthetic data. We also compare with Pix2Pix [22], where it receives the same inputs as our model and generates the appearance for the full object. Using groundtruth masks (SeGAN w/ GT SV) shows that our method will perform even better if it receives more accurate masks for the visible regions as the input. Figure 4 shows qualitative segmentation results of our method.

We now evaluate our network on appearance generation (painting). Our first baseline is a nearest neighbor method. We feed the image into ResNet-18 pretrained on ImageNet and obtain features from the layer before the classification layer. Similarly, we feed the mask image for the visible region into ResNet and obtain features for the mask as well. We concatenate these features. For each test example, we find the training image and mask that has the smallest distance (L2 distance on the concatenated features).

As another baseline, we use the Context-Encoder network [37]. The main task of their network is to complete a cropped patch of an image using the context around it, so we crop a box around the object but leave the visible pixels unchanged. In other words, we remove all the pixels on the image patch that can potentially belong to the invisible

Model	Training Data	Input Mask	Loss Mask	Visible ∪ Invisible	Visible	Invisible
IBBE [27]	natural and synthetic	SV	SF	31.0	25.2	5.1
Multipath [51]	natural and synthetic	SV	SF	36.0	34.8	12.3
Multipath [51]	natural	SV	SV	50.3	49.3	9.4
Pix2Pix [22]	natural and synthetic	SV	SF	52.3	49.6	11.9
SeGAN (ours) w/ predicted SV	natural and synthetic	SV	SF	**66.4**	**60.1**	**19.1**
SeGAN (ours) w/ GT SV	synthetic	SV	SF	76.4	63.9	27.6

Table 1. **Segmentation evaluation.** We compare our method with [27], [51], and [22] on the synthetic test data. SV and SF refer to the mask for visible regions of objects and the full object, respectively. We evaluate how well we predict 'Visible' regions, 'Invisible' regions and their combination. The bottom row is not comparable with other rows since it uses groundtruth information.

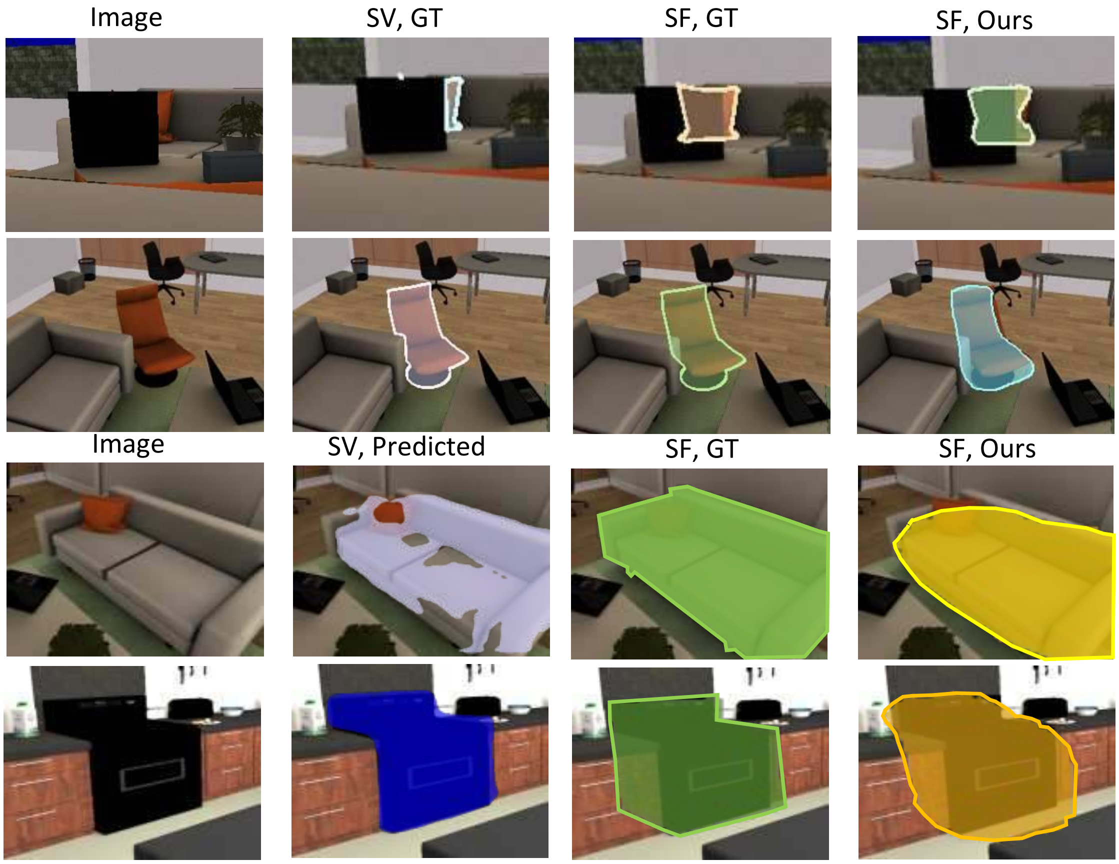

Figure 4. **Qualitative results of segmentation.** We show the results using groundtruth for the visible region (SV, GT) in the first two rows, and using the predicted mask in the last two rows. The groundtrouth for the full object (SF, GT), and our predicted mask for the full object (SF, Ours) are also shown.

regions and calculate the loss on the full object.

As our last baseline, we train the Pix2Pix network [22] on our dataset by feeding just the pixels for the visible part of the image as an input and calculating the loss on the full object (visible and invisible regions). The network is supposed to learn to generate the appearance of the full object.

Table 2 shows the results for painting. Our model outperforms all of the baselines methods. Again, for factoring out the performance of the SV prediction methods we repeat all of the experiments with groundtruth SV masks. The qualitative results for this experiment can be seen in Figure 5.

Model	Input	L1	L2
Nearest Neighbor (NN)	Image + SV_p	0.20	0.12
Context-Encoder [37]	Image	0.18	0.09
Pix2Pix [22]	Image + SV_p	0.15	0.09
SeGAN (ours)	Image + SV_p	**0.11**	**0.06**
SeGAN (ours)	Image + SV_{gt}	0.05	0.02

Table 2. **Painting evaluation.** We use L1 and L2 distances as the evaluation metric so lower numbers are better. p and gt subscripts refer to predicted masks (by [51]) and groundtruth, respectively.

We also performed a human study using Mechanical

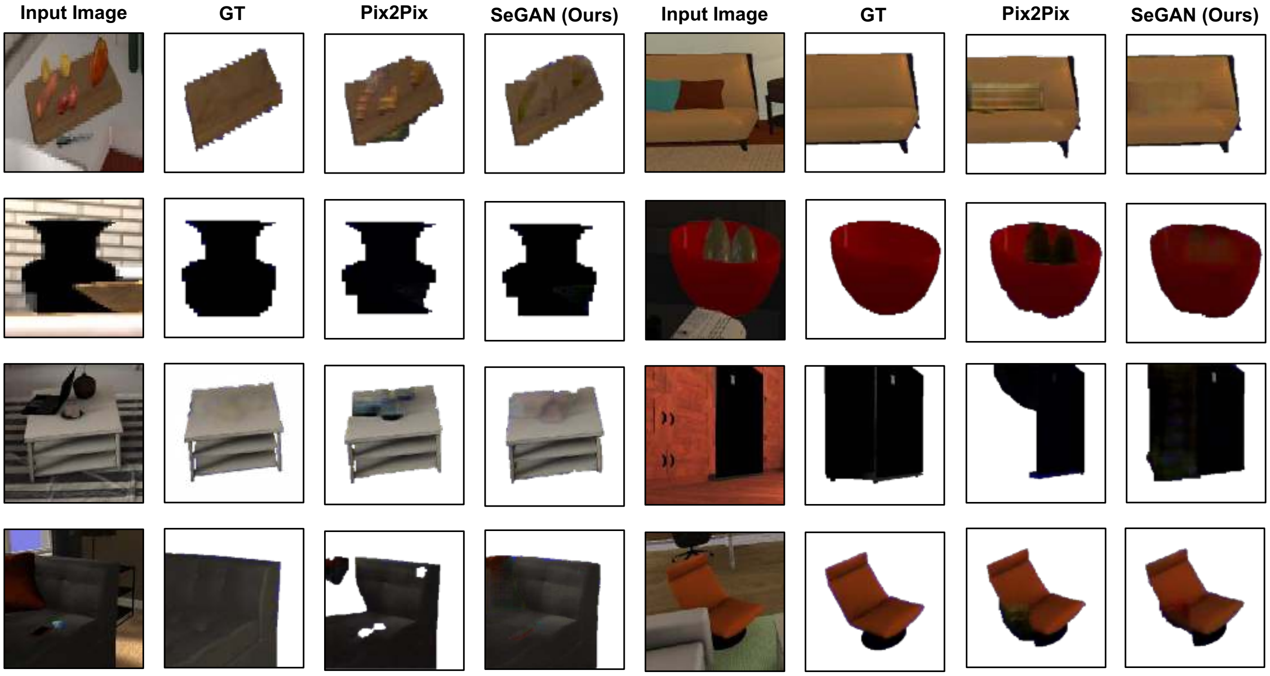

Figure 5. **Qualitative results of painting.** We show the input image, the groundtruth object (without any occlusion) and the result of Pix2Pix [22] network.

Turk. Table 3 shows how often the result of a particular method is chosen as the best generated image.

Model	NN	Pix2Pix [22]	SeGAN (ours)
Chosen as best	0.46%	29.74%	**69.78%**

Table 3. **Human study results.** We show how often the subjects select the generated images of our method versus two baselines: nearest neighbor (NN) and Pix2Pix [22].

Generalization to natural images. We now evaluate whether our model generalizes to natural images when it is trained on our synthetic data. A major issue is that there is no dataset of natural images that provides large-scale and accurate mask annotations and texturing for both occluded and non-occluded regions of objects. [55, 27] construct datasets for object occlusion, but occlusion patterns of [27] are unnatural, and the dataset of [55] does not include the appearance for the occluded regions. Therefore, we report the results only for the segmentation task.

To evaluate the performance of our model on natural images, we construct a dataset using PASCAL 3D dataset [48]. PASCAL 3D associates a 3D CAD model to each object instance in the dataset and it also provides annotations in terms of azimuth, elevation and distance of the camera with respect to the objects. Therefore, we can project the 3D CAD model onto the image and obtain the segmentation mask for full objects. Note that the projection does not have any occlusion information so it generates the mask for the full object (SF) as if it is not occluded by any object. On the other hand, datasets such as [17] provide the segmentation mask for the visible region of the objects. Hence, we can obtain the mask for the occluded regions by subtracting these masks from the mask we obtain by projecting the CAD models. We use five indoor categories of PASCAL 3D dataset (i.e. bottle, chair, diningtable, sofa, and tvmonitor) for our experiments. As before, we run Multipath on these natural images to obtain the segmentation masks for

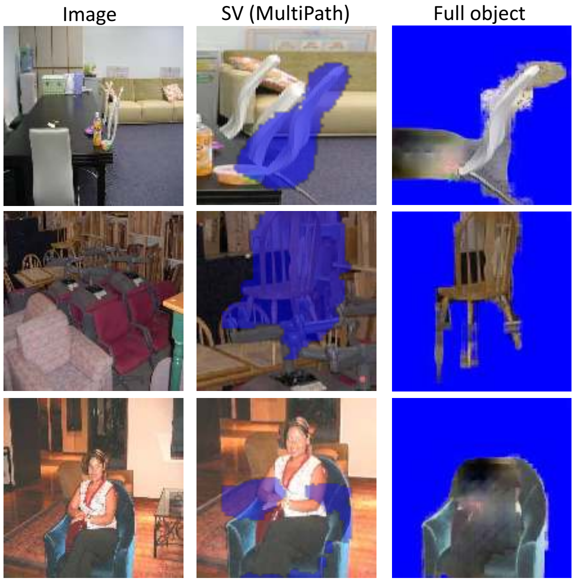

Figure 6. **Qualitative results of generalization to natural images.** We show the prediction for the visible region using Multipath (SV) and segmentation and painting results of our method.

Model	Vis. ∪ Invis.	Visible	Invisible
Multipath [51] (natural)	46.8	**58.9**	13.9
Multipath [51] (natural+syn.)	46.3	54.0	8.4
Ours w/ predicted SV	**47.3**	58.1	**18.7**
Ours w/ GT SV	50.7	58.9	23.1

Table 4. **Evaluation of generalization to natural images.** The input masks of our network are SV masks, and the groundtruth mask for the loss function is SF.

the visible regions.

Table 4 shows the results of generalization to natural images. Although, our method only uses synthetic occlusion information for training, it is still able to provide accurate results on natural images. More importantly, our method outperforms Multipath on segmentation of the invisible regions (SI), while there is only a slight degradation in the performance for the visible regions. Combining natural and synthetic data makes the performance worse. It probably makes the training more difficult. Example predictions of our method on natural images are shown in Figure 6.

Depth layering. The final experiment for evaluating our network is depth layering i.e. it infers the depth layers for a pair of objects with respect to the camera. Depth layering can be inferred from occlusion relationships since typically the occluder is in front of the occludee and is closer to the camera. The groundtruth data for this problem can be easily obtained from our dataset since we have access to the occlusion relationships in the data.

For this task, we first predict the segmentation mask for the full object (SF) using our network. Then amongst the rest of the objects in this scene we find the ones, whose segmentation mask for visible region (SV) intersects with the predicted mask for the invisible region (SI). An object q occludes object p if intersection over union of the segmentation mask for the visible region (SV) of object q with the segmentation mask of the invisible region (SI) of object p is above a threshold. The threshold that we use is 5%.

As the baseline for this task, we use the method of [10], which infers depth map from a single image. To obtain the depth layering result for this method, for each image, we project the groundtruth segmentation mask for the visible region (SV) onto the output of the depth estimation and compute the average depth value for all the pixels that are inside the mask. This will be the estimated depth for this object. Then, for each pair of occluding objects we compare their depth, and the one that has a lower depth will be closer to the camera and, therefore, we consider it as the occluder.

The evaluation metric is defined as the average ratio of the number of correct predictions over the number of occlusion pairs for each image.

We report the average over all images in the test dataset. The number of occlusion pairs in our test set is 4M pairs.

Single Image Depth [10]	Ours
60.18	**84.04**

Table 5. **Depth layering results.** We compare our depth layering results with the results of [10], which estimates depth from single RGB images. We modify [10] to use masks.

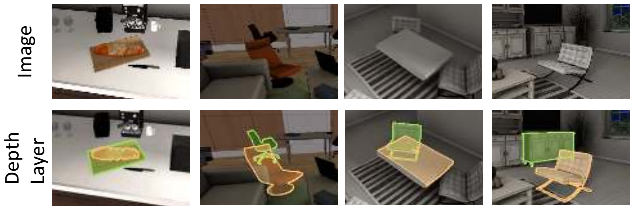

Figure 7. **Qualitative results of depth layering.** We predict that the object shown with the orange mask is closer to the camera than the object shown with the green mask.

The result for this task is shown in Table 5. This task has been evaluated on the synthetic data since we have accurate depth and occlusion information. Figure 7 shows the qualitative results of depth layering.

There are several reasons why our model is more accurate. First, in many cases the occluder and occludee do not differ enormously in depth, which makes it difficult for the baseline to infer the occlusion relationship. Second, the predicted depth map is computed over the entire image. Hence, many low-level details have been removed while our predicted mask is for a specific object and does not lose much information around the occlusion boundaries. Third, our network predicts the invisible regions of the object, but the depth estimator does not have this capability.

6. Conclusion

In this paper, we address the problem of segmentation and appearance generation for the invisible regions of objects. We introduced SeGAN, which is a Generative Adversarial Network that generates the appearance and segmentation mask for invisible and visible regions of objects. Getting large-scale and accurate training data for this task is challenging. Our solution is to use photo-realistic synthetic data where we can obtain the exact boundaries of the invisible regions. Our experimental evaluations show that our model outperforms segmentation baselines, while it generates the appearance (as opposed to a binary mask). We also showed that our method outperforms GAN-based baselines for appearance generation and painting. We show generalization to natural images when the method is trained on synthetic scenes. Moreover, we evaluate our model for the task of depth layering and show improvements over a single image depth estimation baseline.

Acknowledgements: This work is in part supported by ONR N00014-13-1-0720, NSF IIS-1338054, NSF-1652052, NRI-1637479, Allen Distinguished Investigator Award, and the Allen Institute for Artificial Intelligence.

References

[1] A. Aguiar and R. Baillargeon. Developments in young infants' reasoning about occluded objects. *Cognitive psychology*, 2002. 1

[2] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv*, 2016. 2

[3] L.-C. Chen, Y. Yang, J. Wang, W. Xu, and A. L. Yuille. Attention to scale: Scale-aware semantic image segmentation. In *CVPR*, 2016. 2

[4] Q. Chen and V. Koltun. Photographic image synthesis with cascaded refinement networks. In *ICCV*, 2017. 2

[5] Y.-T. Chen, X. Liu, and M.-H. Yang. Multi-instance object segmentation with occlusion handling. In *CVPR*, 2015. 2

[6] J. Dai, K. He, Y. Li, S. Ren, and J. Sun. Instance-sensitive fully convolutional networks. In *ECCV*, 2016. 2

[7] J. Dai, K. He, and J. Sun. Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In *ICCV*, 2015. 2

[8] J. Dai, K. He, and J. Sun. Instance-aware semantic segmentation via multi-task network cascades. In *CVPR*, 2016. 2

[9] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR*, 2009. 5

[10] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *ICCV*, 2015. 8

[11] T. Gao, B. Packer, and D. Koller. A segmentation-aware object detection model with occlusion handling. In *CVPR*, 2011. 2

[12] G. Ghiasi, Y. Yang, D. Ramanan, and C. Fowlkes. Parsing occluded people. In *CVPR*, 2014. 2

[13] R. Girshick. Fast r-cnn. In *ICCV*, 2015. 2

[14] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014. 2

[15] R. B. Girshick, P. F. Felzenszwalb, and D. A. McAllester. Object detection with grammar models. In *NIPS*. 2011. 2

[16] R. Guo and D. Hoiem. Beyond the line of sight: labeling the underlying surfaces. In *ECCV*, 2012. 2

[17] B. Hariharan, P. A. Arbeláez, L. D. Bourdev, S. Maji, and J. Malik. Semantic contours from inverse detectors. In *ICCV*, 2011. 7

[18] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *ECCV*, 2014. 2

[19] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 3, 5

[20] E. Hsiao and M. Hebert. Occlusion reasoning for object detection under arbitrary viewpoint. In *CVPR*, 2012. 2

[21] P. Isola and C. Liu. Scene collaging: Analysis and synthesis of natural images with semantic layers. In *ICCV*, 2013. 2

[22] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017. 2, 3, 4, 5, 6, 7

[23] A. Kar, S. Tulsiani, J. Carreira, and J. Malik. Amodal completion and size constancy in natural scenes. In *ICCV*, 2015. 2

[24] P. Krähenbühl and V. Koltun. Learning to propose objects. In *CVPR*, 2015. 2

[25] C. Li and M. Wand. Precomputed real-time texture synthesis with markovian generative adversarial networks. In *ECCV*, 2016. 2

[26] K. Li, B. Hariharan, and J. Malik. Iterative instance segmentation. In *CVPR*, 2016. 2

[27] K. Li and J. Malik. Amodal instance segmentation. In *ECCV*, 2016. 2, 5, 6, 7

[28] G. Lin, C. Shen, , and I. van den Hengel, Anton Reid. Efficient piecewise training of deep structured models for semantic segmentation. In *CVPR*, 2016. 2

[29] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollr, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 5

[30] Z. Liu, X. Li, P. Luo, C.-C. Loy, and X. Tang. Semantic image segmentation via deep parsing network. In *ICCV*, 2015. 2

[31] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 2

[32] M. Mathieu, C. Couprie, and Y. LeCun. Deep multi-scale video prediction beyond mean square error. In *ICLR*, 2016. 2

[33] M. Mirza and S. Osindero. Conditional generative adversarial nets. *ArXiv*, 2014. 2, 3

[34] N. Newcombe, J. Huttenlocher, and A. Learmonth. Infants coding of location in continuous space. *Infant Behavior and Development*, 1999. 1

[35] H. Noh, S. Hong, and B. Han. Learning deconvolution network for semantic segmentation. In *ICCV*, 2015. 2

[36] G. Papandreou, L.-C. Chen, K. Murphy, and A. L. Yuille. Weakly-and semi-supervised learning of a dcnn for semantic image segmentation. In *ICCV*, 2015. 2

[37] D. Pathak, P. Krähenbühl, J. Donahue, T. Darrell, and A. Efros. Context encoders: Feature learning by inpainting. In *CVPR*, 2016. 2, 4, 5, 6

[38] B. Pepik, M. Stark, P. Gehler, and B. Schiele. Occlusion patterns for object class detection. In *CVPR*, 2013. 2

[39] P. O. Pinheiro, R. Collobert, and P. Dollar. Learning to segment object candidates. In *NIPS*, 2015. 2

[40] P. O. Pinheiro, T.-Y. Lin, R. Collobert, and P. Dollár. Learning to refine object segments. In *ECCV*, 2016. 2

[41] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. *CVPR*, 2016. 2

[42] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. 2

[43] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241. Springer, 2015. 3

[44] B. C. Russell and A. Torralba. Labelme3d: Building a database of 3d scenes from user annotations. In *CVPR*, 2009. 2

[45] C. Szegedy, S. Reed, D. Erhan, and D. Anguelov. Scalable, high-quality object detection. *arXiv*, 2014. 2

[46] J. Tighe, M. Niethammer, and S. Lazebnik. Scene parsing with object instances and occlusion ordering. In *CVPR*, 2014. 2

[47] J. Winn and J. Shotton. The layout consistent random field for recognizing and segmenting partially occluded objects. In *CVPR*, 2006. 2

[48] Y. Xiang, R. Mottaghi, and S. Savarese. Beyond pascal: A benchmark for 3d object detection in the wild. In *WACV*, 2014. 7

[49] Y. Yang, S. Hallman, D. Ramanan, and C. Fowlkes. Layered object models for image segmentation. In *PAMI*, 2012. 2

[50] R. Yeh, C. Chen, T. Y. Lim, M. Hasegawa-Johnson, and M. N. Do. Semantic image inpainting with perceptual and contextual losses. *arXiv*, 2016. 2

[51] S. Zagoruyko, A. Lerer, T.-Y. Lin, P. O. Pinheiro, S. Gross, S. Chintala, and P. Dollár. A multipath network for object detection. In *BMVC*, 2016. 2, 3, 5, 6, 8

[52] Z. Zhang, S. Fidler, and R. Urtasun. Instance-level segmentation for autonomous driving with deep densely connected mrfs. In *CVPR*, 2016. 2

[53] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. Torr. Conditional random fields as recurrent neural networks. In *ICCV*, 2015. 2

[54] Y. Zhu, R. Mottaghi, E. Kolve, J. J. Lim, A. Gupta, L. Fei-Fei, and A. Farhadi. Target-driven visual navigation in indoor scenes using deep reinforcement learning. In *ICRA*, 2017. 4

[55] Y. Zhu, Y. Tian, D. Mexatas, and P. Dollár. Semantic amodal segmentation. In *CVPR*, 2017. 2, 7