# OCONet: Image Extrapolation by Object Completion

Richard Strong Bowen[1†] Huiwen Chang[2] Charles Herrmann[1†]
Piotr Teterwak[3†] Ce Liu[2] Ramin Zabih[1,2]
[1]Cornell Tech [2]Google Research [3]Boston University
{rsb, cih, rdz}@cs.cornell.edu {huiwenchang,celiu}@google.com piotrt@bu.edu

## Abstract

*Image extrapolation extends an input image beyond the originally-captured field of view. Existing methods struggle to extrapolate images with salient objects in the foreground or are limited to very specific objects such as humans, but tend to work well on indoor/outdoor scenes. We introduce OCONet (*Object COmpletion Networks*) to extrapolate foreground objects, with an object completion network conditioned on its class. OCONet uses an encoder-decoder architecture trained with adversarial loss to predict the object's texture as well as its extent, represented as a predicted signed-distance field. An independent step extends the background, and the object is composited on top using the predicted mask. Both qualitative and quantitative results show that we improve on state-of-the-art image extrapolation results for challenging examples.*

## 1. Introduction

Image extrapolation, which extends pixels beyond image borders, is an important technique for computational photography. It is related to image interpolation techniques such as [4,5,6], which also infer missing pixels, and allow users to change image dimensions/aspect ratios without changing the content of the original images. Extrapolation, however, is a much more challenging problem since there is much less information available; while inpainting methods are given the entire boundary of the missing region, in image extrapolation we only know one border. This less constrained problem means the the method needs to extrapolate both textures and structures in a convincing manner.

Image extrapolation methods include both classical [5, 7, 8, 9, 10] and learning-based approaches [1, 3, 11, 12]. Classical methods often use guide images, for example [13] finds similar images on the Internet and stitches them together to expand the input image. This method makes strong assumptions, and is only applicable for pictures taken at locations such famous landmarks, where a large set of reference images are available.

Learning-based approaches for image extrapolation have only recently emerged, notably including Boundless [1], Wide-Context [3], Panorama Synthesis [11], and Pluralistic Image Completion [12]. The success of generative adversarial networks (GAN's) [14, 15] motivated these methods. Although similar methods have existed for interpolation for several years, the difficulty of extrapolation required more specialized and more powerful generative models.

Despite the recent progress in image extrapolation by methods such as [1,3] on textural images, the problem is still far from being solved for objects. While domain-specific image interpolation exists for a few important classes (e.g. for people [16]), the generic problem for images with salient objects remains unsolved.

The complexity of natural scene composition makes it challenging for a generic encoder-decoder network trained with adversarial losses to uncover the diverse shapes and final details of foreground object shapes given an input. It is easier to model the shape and appearances of each object class independently, e.g. cars, airplanes, people and dogs, as suggested by [15].

In this paper we introduce OCONet (Object COmpletion Networks) to address the image extrapolation problem for a broad set of images with general object classes. Recent advances in high-quality instance segmentation, e.g. ShapeMask [17], allow us to obtain object class and accurate foreground object shape masks even when only a small fraction of the object is visible inside the image boundary. Using this information, we trained a class-conditioned object model to infer both the shape and pixels of foreground objects, as well as a background model to extrapolate the background. The completed object is simply composited on top of the extrapolated background to obtain the final result. As shown in figure 1, we produce significantly better results on the object of interest. Extensive quantitative and qualitative experiments show that our model significantly outperforms the prior state-of-the-art.

To summarize, our contributions are as follows:

- We introduce object completion networks – OCONet–

[†]Work performed while author was at Google Research

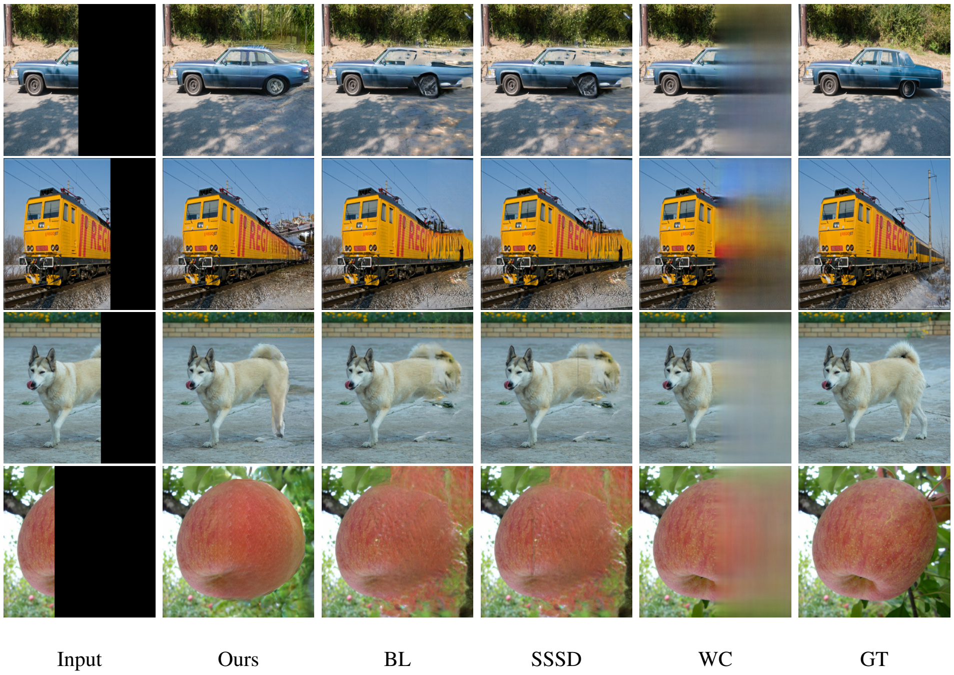



Figure 1: Examples of our method on 4 different object categories: cars, trains, dogs, and apples. Comparisons include BL=Boundless [1], SSSD=Self-Supervised Scene De-Occlusion [2] , WC=Wide Context [3], GT=Ground Truth.

which complete a single object independent from the rest of the extrapolation problem.

- We show that the sign-distance field (SDF) is effective as an internal representation of the segmentation mask for 2D shape completion (extrapolating the mask).
- We demonstrate substantially improved quantitative and qualitative extrapolation results for a number of important object classes on OpenImages [18].

# 2. Related Work

## 2.1. Inpainting

Prior work on inferring unseen pixels has mostly focused on image inpainting, the task of completing an image with context on all sides. Image inpainting methods can be divided into two categories: non-parametric classical methods and learning-based methods, which are typically neural network-based. Classical methods, such as PatchMatch [5], typically borrow image statistics from the known region to complete the unknown area. This works fairly well for textures, but less well for objects because the methods only enforce local consistency.

Learning-based methods mark a big step forward in enforcing global consistency. They mostly consist of encoder-decoder models, typically trained with an adversarial loss [14]. Notable works include the Context Encoder [6], [19] for adding local and global discriminators, [20] for adding contextual attention which can borrow texture patches, and [21, 22] which solve the issue that convolutions cannot discriminate between valid pixels in the known region and invalid ones in the unknown region. Some more recent techniques have added stochasticity to the completions [12, 23] by using conditional variational autoencoders.

## 2.2. Image Extrapolation

Our work focuses on inferring pixels outside of the input image, a task also known as uncropping, outpainting or image extension. Similar to inpainting, this problem has been studied for a long time and many non-parametric methods have been developed. However, this task is significantly more difficult than inpainting, since it effectively requires extrapolating pixels rather than interpolating them. As [1] demonstrates, successful inpainting methods perform quite poorly on this harder task.

Early techniques often relied on images of the same scene taken from a different camera position or angle; these images would then be combined to produce an extended field-of-view using a technique called image stitching [24, 25, 26, 27], which finds locations to transition between the images and then composites them into the same output space. Photo Uncrop [13], one of the first papers to extrapolate from a single image of a scene, used an image database to find images similar to the input image and then stitched them together to extend the field-of-view. Recently, even non-learning based approaches to image stitching have moved towards techniques which are aware of objects [28] and saliency [29].

More recent learning-based methods for image extrapolation, such as Wide-Context Semantic Image Extrapolation [3] and Boundless [1] only receive a single image as input and use deep learning to fill in plausible extrapolations. These typically use an encoder-decoder structure and adversarial loss as a starting point, and are trained on diverse datasets. [1] uses a Wasserstein GAN [30, 31] framework and discriminator conditioning to stabilize the GAN training; while [3] introduces a "feature expansion" operator to do extrapolation and an implicit diversified MRF loss to improve texture. Both of these techniques perform well on backgrounds but often struggle with objects. Our work directly addresses this weakness. Spiral Generative Networks [32] introduces a spiral curve ordering to generating the unseen pixels. Their published examples do not contain the kind of challenging imagery that is our focus. It would be interesting to test their technique on our dataset, but as of this writing their code is not publicly available.

Domain specific image extrapolation techniques include Deep Portrait Image Completion [16], which is specific to people and uses additional human-related priors such as a pose sub-net; and [11], which is optimized for scenic panoramas with a recurrent outpainting in latent space. Other work focuses on providing more flexibility to the process; for example, [33] generates a diverse set of possible results from a small input such as a foreground object, and [34] uses an editable configuration of bounding boxes to control the appearance of the output image. Self-supervised scene de-occlusion [2] focuses on the problem of scene deocclusion, which allows a user to edit the depth order of objects in a scene. One part of this process included uncropping occluded objects; however, this uncropping task differs substantially from our task in both magnitude and style. The deocclusion network relies on the occlusion masks as inputs; these masks constrain the problem and limit the possible shapes that the uncropped object can take. Additionally, these objects often only require a small amount of uncropping, different than the large, variable scale uncropping addressed in this paper.

## 3. Technical Approach

OCONet is broken down into several stages, shown in figure 2. Here we briefly describe our method in the case of a single object on the border.[1] Our models are implemented in TensorFlow [35], and a more detailed description is provided in the supplemental material.

1. **Input** The input, shown at left in figure 2, is a cropped color image $I$ of size $H \times W \times 3$.

2. **Interest mask generation** A mask of shape $H \times W \times 1$ is provided to the network – as user input or inferred by a separate instance segmentation system. This mask indicates which object should be extrapolated. This is then stacked into an Image, Mask Tensor: $[I; M]$.

3. **Object completion** A class-conditioned network maps $[I; M]$ to a 4-D texture: pixels and estimated mask. The mask is represented as a signed distance field.

4. **Background extrapolation** We replace the pixels in $I$ at location $M$ with 0 so the object does not affect the background extrapolation. We then use existing techniques [1] to produce a background.

5. **Compositing** The predicted mask (thresholded to transform from predicted-SDF to a 0-1 mask) is used to do a simple compositing.

**Interest Mask** The interest mask indicates which object we should complete. At training time, we use the ground-truth segmentation annotations. At test time, we replace the mask with the results of an off-the-shelf instance segmentation model [17]. We note that any segmentation model can be plugged in to our method, so improvements in instance segmentation will produce improvements in our model; a comparison between inferred and ground truth interest masks is given in the supplemental.

**Object Completion** We infer the additional pixels using an encoder-decoder with skip connections and gated convolutions [21]. This network also predicts a mask, predicted as a signed distance function (full details in section 3.3). The object completion network is trained with 3 loss terms. First is a mask loss: an L1 loss on the mask output. On the pixels, we apply a mask-modulated variant of LPIPS loss [36], using 5 layers of a pretrained VGG16 [37] baseline network, as well as a simple L2 pixel loss (also mask-modulated). Class-conditioning is achieved by learning a single code per object class, which is concatenated between the decoder and encoder. Full details are in the supplemental.

**Background Model and Compositing** The final step is to composite this foreground object onto an extrapolated background. Since we mostly focus on the foreground object in this work, we simply use a Boundless [1] model for the background. We composite foreground objects using the (clipped) predicted mask as an alpha mask. We leave more sophisticated compositing for future work.

For our background prediction, we make some small changes to the Boundless training scheme. These are motivated by the observation that a Boundless model, if run on our cropped images, will also try to extend the foreground

[1] In our dataset, a typical extrapolation problem has a dominant foreground object. The fact that the object completion network can work independently suggests an extension to the rarer multiple-object case, although compositing becomes less straightforward.

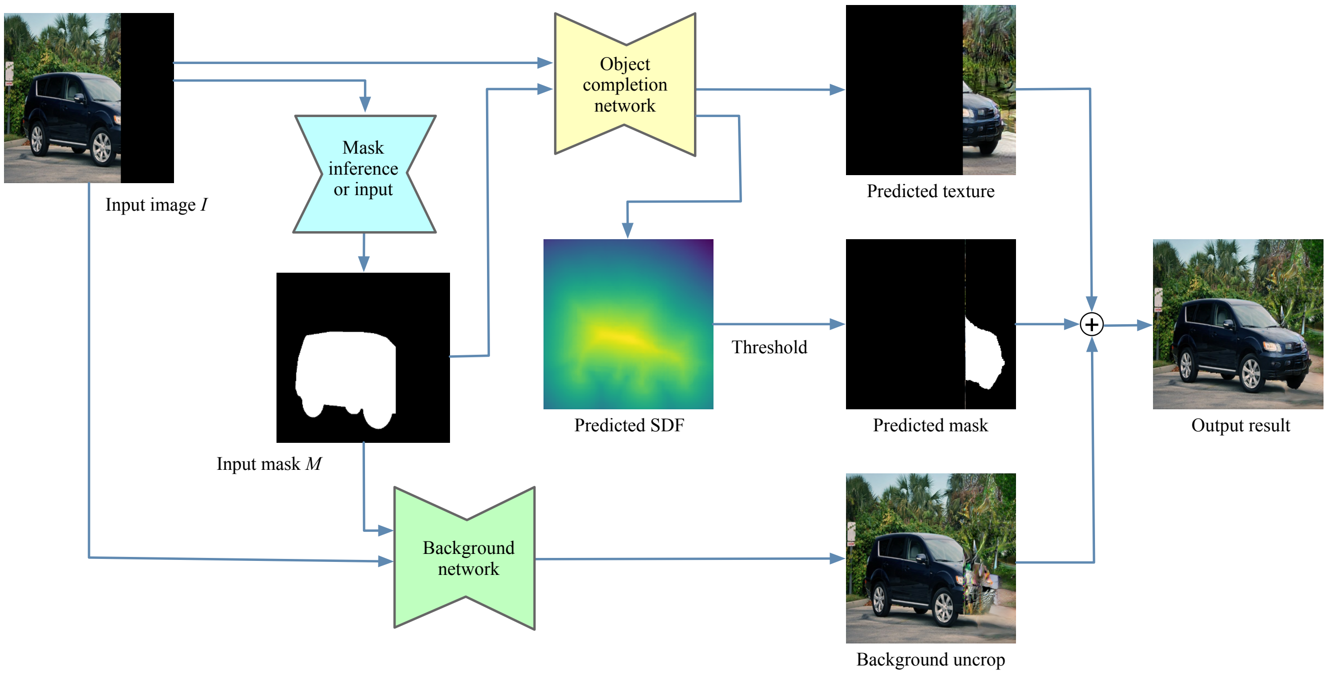



Figure 2: Stages of OCONet. Note that the segmentation mask $M$ can either be obtained automatically through inference from a neural network (like ShapeMask) or given as input by a user. Additionally, $M$ serves both as an input to the object completion network as well as a mask to the background network input, so that the latter only receives background content.

object. This extended foreground may contain blurry edges and potentially extends further than our model's predicted mask, eliminating the sharp object boundary. In order to prevent these potential issues, we zero out the foreground object in the generator's input using the object mask $M$. We then use two discriminators. One duplicates the discriminator in Boundless; this discriminator sees the entire uncropped region (but not the input region). The other discriminator sees the entire image, but in both ground truth and generated images, we zero out the entire object. Pixel losses are modulated by the same mask. These encourage the generator to produce an extension that is both seamless across the uncrop boundary and does not contain copied grayed-out object pixels (these pixels can cause a haloing effect).

### 3.1. Adversarial Loss

We apply an adversarial loss as a fine-tuning step to improve image quality. We use a PatchGAN [38] with spectral normalization [39]. The discriminator sees both the generator's output pixel and its ground truth loss. We use hinged Wasserstein loss, i.e., the discriminator loss function for a real or generated example is

$$\mathcal{L}_{\text{disc}} = \begin{cases} \frac{1}{N_{\text{pix}}} \sum_{(x,y)} \max(1 - D(x,y), 0) & \text{(real)} \\ \frac{1}{N_{\text{pix}}} \sum_{(x,y)} \max(1 + D(x,y), 0) & \text{(generated)} \end{cases}$$

where $N_{\text{pix}}$ is the number of pixels at the last layer of the discriminator and $D$ is the discriminator output. The generator loss is composed of GAN loss, feature matching [40] loss and reconstruction loss:

$$\mathcal{L}_{\text{gen}} = \frac{1}{N_{\text{pix}}} \left( \sum_{(x,y)} -D(x,y) \right) + \lambda_{\text{adv}} \mathcal{L}_{\text{object}} + \lambda_{\text{fm}} \mathcal{L}_{\text{fm}}$$

where $\mathcal{L}_{\text{object}}$ is the reconstruction loss described above (and in more detail in the supplemental) and $\mathcal{L}_{\text{fm}}$ is a feature-matching loss:

$$\mathcal{L}_{\text{fm}} = \sum_i \frac{1}{N_i} \sum_{x,y} \left( \hat{\phi}_i(I_{\text{real}}) - \hat{\phi}_i(I_{\text{gen}}) \right)^2$$

with $\hat{\phi}_i$ being features in the $i$th layer of the discriminator, normalized along the channel dimension. In our experiments $\lambda_{\text{adv}} = \lambda_{\text{fm}} = 1$.

### 3.2. Dataset

We construct a dataset for training the model from a subset of Open Images [18, 41]. We consider all images in the dataset for which we have a per-pixel segmentation mask. We further filter down to objects whose mask is at least 1024 pixels total and not within ten pixels of the boundary. A final filtration step tries to avoid badly occluded objects by requiring that the second largest connected component of the instance (if there is one) is no more than 5% the size of the largest connected component. Our dataset then consists of image-object pairs; for example, an image with two large objects with segmentations will constitute two image-objects pairs (the image with object 1 and the image with object 2). Dataset statistics are given in the supplemental.

At train time, given an object of interest, we call the minimum of its bounding box's width and height its representative size $R$. We then choose a random square crop of the original image $I$ so that 1) the entire object is in the crop, and 2) the square crop's side length is not more than $4R$. That is, we choose a bounding box with a minimum side length of $\max(w_o, h_o)$ and a maximum side length of $\min(w_I, h_I, 4R)$ where $w_o$ and $h_o$ are the object bounding box dimensions and $w_I$ and $h_I$ are the original image dimensions. This provides augmentation in scaling and positioning.

This crop is resized to $256 \times 256$, and augmented with a random horizontal flip. We randomly choose the crop between 25% and 75% of the way from left to right of the instance bounding box. At validation time, we use a deterministic variant of the above: we use a fixed ratio instead of random, the side lengths are deterministically halfway between the minimum and maximum, and the object is centered. The splits are identical to the original Open Images data. To evaluate the system's behavior in an end-to-end automatic way, we replace the mask on the cropped image with one detected by an off-the-shelf instance segmentation algorithm trained on COCO [42]. All example images shown in this paper and FID scores are computed this way, fully automatically; additional details are in the supplemental.

### 3.3. Use of signed-distance fields

The key challenge in predicting an uncropped mask is the inevitable uncertainty, since multiple shapes could plausibly complete the cropped mask. We investigated several natural approaches that predict a 0-1 per-pixel value. However, we obtained significantly better performance by predicting a signed-distance field [43] instead. Given a set of pixels $S$, its signed-distance field is defined as

$$f(x) = \begin{cases} \min_{s \in S} d(x, s) & x \notin S \\ -\min_{s \notin S} d(x, s) & x \in S \end{cases}$$

For a training example, the ground truth SDF can be easily computed using the Euclidean transform. Note that while the indicator for $S$ is discontinuous, $f$ is smooth. SDFs are common in 3d shape representation [44].

For the mask prediction task, the most direct technique would be to predict a value between 0-1 per-pixel using either an L1 or cross-entropy loss. Cross-entropy loss encourages the model, at each pixel, to output its estimate of the probability that that pixel is part of the mask. This has the effect of producing a blurry mask (large regions of intermediate values) when the model is uncertain, as shown in figure 3. In contrast, L1 encourages the model to output 1 or 0, which naturally leads to sharp edges. As such, the model will at each pixel produce a 1 if the probability is greater than a half and a 0 otherwise, similar to the median prediction. This tends to fail on thin, ambiguous structures like the horse's legs, as shown in figure 3.

Instead, we predict the sign-distance field, a similar representation as Hu *et al*. [45]. This gives us the best of both worlds: the model output is smooth (because the SDF is smooth), but our final mask can be sharp since we can select only the pixels with positive estimated SDF values (choosing positive SDF values is the same as thresholding the predicted SDF at zero; any threshold produces a mask, but because the SDF is predicted everywhere, including the given region, a choice other than 0 would produce a discontinuity at the extrapolation boundary). Our intuition for using SDF's follows that of Hu *et al*. [45]; SDF's implicitly consider the shape and size of the object being modelled. Additionally, the SDF is relatively stable between the small variations in plausible completions of objects; the uncertain regions near the boundary will always have an SDF value near 0. On the other hand, the predicted mask in uncertain regions near the boundary will have sharp 0-1 discontinuities. Qualitatively, the SDF representation outperforms both binary cross-entropy and L1 losses by a significant margin. As shown in figure 3d, the raw SDF successfully captures the distinct legs of the horse.

In each case there is some kind of averaging over possible completions:

- L1 loss drives the network to choose at each pixel the *median* mask value, leading to sharp but clipped masks.
- Cross-entropy loss drives the network to choose at each pixel the *mean* mask value, leading to blurry masks.
- Our SDF setup drives the network to choose the median SDF value, which is smoother, but achieves sharpness by thresholding.

In addition, SDF seems to benefit much more from an adversarial loss than 0-1 per-pixel masks. Adversarial loss for 0-1 per-pixel masks lead to little or no improvements; this can be explained by the significant difference in appearance between a predicted 0-1 per-pixel mask (which will have intermediate values) and real masks (which will be binary). For SDF masks, both the ground truth and predicted fields are smooth functions. Note that the predicted SDF and ground truth SDF can have the same values; specifically, they do not have the same distributional issue as 0-1 per-pixel masks. Adversarial loss leads to improved performance on thin structures, as shown in figure 4 which shows the predicted mask for the bird before and after applying adversarial loss; after adversarial loss, the edges have sharpened and the tail of the bird is present.

(a) Input (b) Cross entropy loss (c) L1 loss (d) SDF (e) Mask implicit in SDF

Figure 3: Example of mask prediction techniques. L1 loss tends to produce masks that clip out uncertain parts; cross-entropy gives blurry masks. SDF ameliorates these difficulties, at the cost of some blobiness in the predicted mask

(a) Input (b) No adversarial loss (c) Adversarial loss (d) No adversarial loss (e) Adversarial loss

Figure 4: Predicted masks (b) before and (c) after fine-tuning with an adversarial loss; the thin structure of the bird's tail has been improved, shown in the corresponding mask and texture (d,e).

## 4. Experiments

We evaluate the proposed object-focused extrapolation method on the Open Images dataset [18], as described in section 3.2. We train our model using ground-truth masks, but to evaluate the method in the presence of possible errors in the mask, we run an off-the-shelf instance segmentation method [17] on the cropped image and replace the ground-truth mask with a detected one. Because our off-the-shelf detector is trained on COCO, we choose some classes that exist in both and run only on filters with this class. The list of classes can be seen in table 1.

We compare our method with state-of-the-art image extrapolation methods: Boundless [1], Wide-Context Image Extrapolation [3], and Self-Supervised Scene De-occlusion(SSSD) [2]. For Boundless, we train on our training dataset with the hyperparameters from [1]. For Wide-Context, we obtain their code from the web. Wide-Context requires a fixed uncrop ratio, so we train and test their networks to extrapolate the right half of images from our dataset, as opposed to using the per-instance uncrop ratio described in section 3.2. We verify that we can train their network by obtaining similar quality to the published results on Celeb-A-HQ [46], but we found that the network did not converge when trained on our dataset with the same settings; this is in agreement with a note on their Github page suggesting that training on large-scale datasets is unstable. A few comparisons with our best effort at training their model is shown in figure 1; the model seems to extrapolate the images similar to a diffusion model, with no edges in the uncropped region.

We also compare with SSSD by framing uncropping as a de-occlusion problem: we treat the uncropping region as a single, rectangular occluding object and use their object-completion model to complete the shape and texture of the query occluded object. We use the pre-trained model of SSSD trained on Coco-A datasets [47]. Since SSSD is not trained on artificial objects of this kind, unsurprisingly it does not perform well at extrapolating the background; for a fairer comparison, we use a boundless model on the original input image, and matte their extrapolated objects onto it using their masks. We find that often, SSSD does not extend the masks very far; therefore, the results often look similar to the Boundless result.

### 4.1. Quantitative Evaluation

We compare with Boundless [1] and SSSD [2] and show FID score [48] and L1. We find substantial improvement in FID, as is reflected in the qualitative results. We find a rough tie in L1, but point out that pixelwise metrics are incorrect for evaluating generative models due to the large number of plausible completions [20] and the fact that they assume pixelwise independence [49]. Deep perceptual metrics better capture human judgements [36]; however, we include the pixelwise score for completeness.

### 4.2. Qualitative Evaluation

Figure 5 shows comparisons between our method and previous state-of-the-art methods [1, 2]. Comparisons with Wide-Context [3] are not shown here as discussed above.

For image extrapolation, we find that making the network aware of object boundaries leads to dramatic improvements. Since our uncropping network produces a sharp mask around the object, the composition step does not have to do any addi-

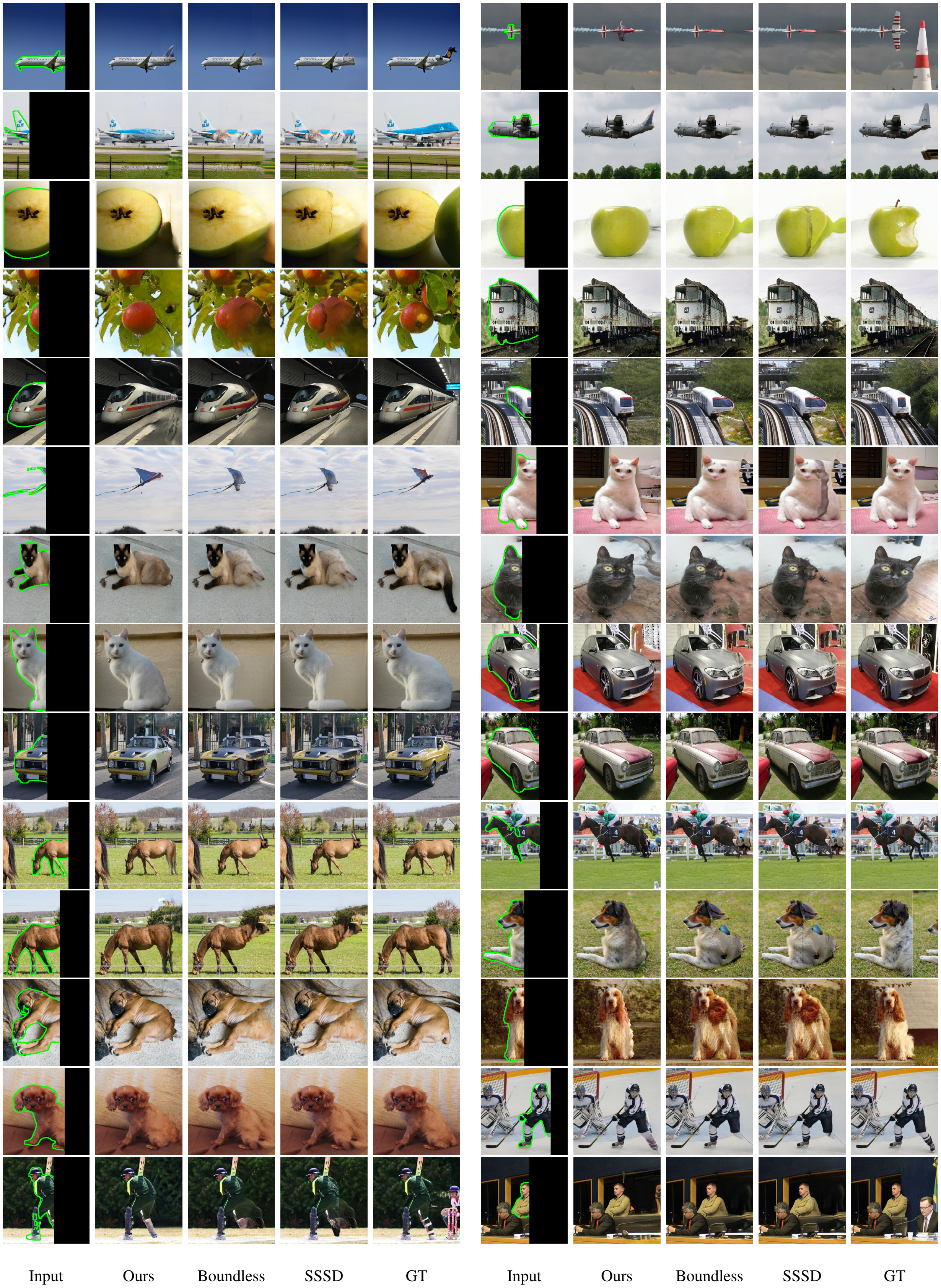
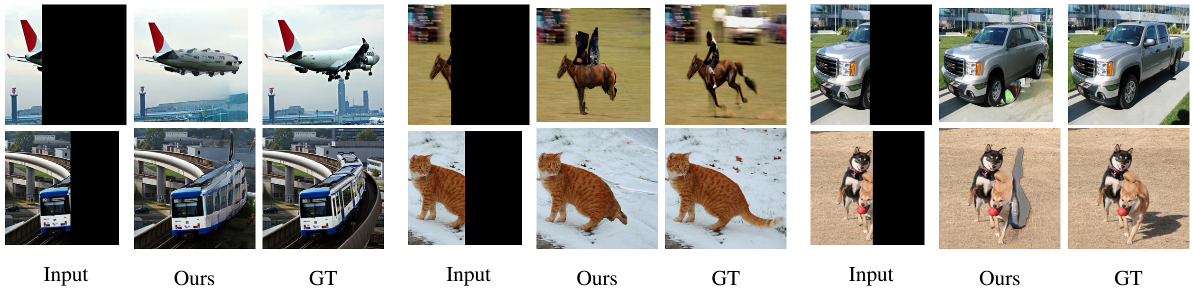



Figure 5: Qualitative comparison between the state-of-the-art methods on a variety of classes.



(a) HF texture (b) Thin structures (c) Background model failures

Figure 6: Selected failure modes

| Category | $n$ | FID (lower is better) Ours | Boundless | SSSD | L1 (lower is better) Ours | Boundless | SSSD |
|---|---|---|---|---|---|---|---|
| Airplane | 1256 | **34.15** | 57.33 | 56.61 | **11.2** | 11.4 | 18.7 |
| Apple | 337 | **83.21** | 107.31 | 122.54 | **18.5** | **18.5** | 18.8 |
| Car | 1396 | **44.00** | 63.48 | 68.41 | **21.3** | 22.2 | 22.8 |
| Cat | 613 | **94.04** | 126.68 | 131.85 | **18.0** | 18.4 | 18.9 |
| Dog | 840 | **74.93** | 89.15 | 92.11 | **17.9** | 18.6 | 19.0 |
| Horse | 909 | **63.21** | 90.58 | 90.31 | **20.3** | 21.0 | 21.2 |
| Kite | 104 | **136.60** | 148.99 | 141.61 | 6.46 | **6.44** | 6.58 |
| Person | 802 | **107.36** | 112.08 | 112.46 | **19.8** | 20.2 | 20.4 |
| Train | 261 | **65.08** | 114.44 | 111.36 | **20.6** | 21.3 | 21.8 |
| All | 6518 | **20.82** | 30.67 | 32.02 | **17.9** | 18.4 | 18.8 |

Table 1: FID and L1 score comparison between Boundless [1], SSSD [2], and our approach, in the end-to-end automatic setting. We substantially improve on FID over previous work. $n$ is the number of each type of object present in the dataset.

tional work to achieve sharp object boundaries. Additionally, we find that the network has memorized facts about object-level features, generating cars' wheels, airplanes' wings, and horses' heads. These attributes, along with the overall design, allow the network to produce a more coherent, complete object with sharp boundaries compared to those produced by Wide-Context and Boundless. For object completion, as SSSD isn't designed to complete objects with a large missing regions, it struggles to extrapolate complex object shapes and textures like dogs' legs and cars' back doors.

### 4.3. Analysis of Interest Mask Quality

In order to understand how the quality of the segmentation mask affects the algorithm, we experiment with uncropping using using the dataset's ground truth masks. For the majority of examples, we observe a very small improvement in overall uncrop quality. Our experiments suggest that, on average, off-the-shelf segmentation masks are close enough to ground truth masks for our technique to perform well on either input. ShapeMask fails to produce a segmentation for about a quarter of the inputs; however, in the remaining cases, the segmentation it provides are generally high quality. Occasional failures can cause problems for our method; examples are included in the supplemental.

### 4.4. Failure Modes

Some selected failure examples are shown in figure 6. The failures we have observed fall into three classes. (1) Thin Structures: We find that the SDF representation for mask prediction helps but in very ambiguous cases we may fail to generate thin structures far from the cropping boundary. (2) High-frequency texture: The model sometimes produces high-frequency textures; we believe this is also in the case of uncertainty, with the model being confused about where to place, for example, the far edge of an object (3) Background artifacts: The background model sometimes produces artifacts which will affect our final composite.

## 5. Conclusion

Our work addresses the challenge of image extrapolation for semantic objects. We show that explicitly factoring out object generation produces much stronger extrapolation results both qualitatively and quantitatively. One challenging aspect is how to represent the mask in the way most conducive to learning; we find that using the SDF representation results in a substantial improvement to extrapolation quality.

## References


[1] Piotr Teterwak, Aaron Sarna, Dilip Krishnan, Aaron Maschinot, David Belanger, Ce Liu, and William T. Freeman. Boundless: Generative Adversarial Networks for Image Extension. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 10521–10530, 2019.

[2] Xiaohang Zhan, Xingang Pan, Bo Dai, Ziwei Liu, Dahua Lin, and Chen Change Loy. Self-supervised scene de-occlusion. *arXiv preprint arXiv:2004.02788*, 2020.

[3] Yi Wang, Xin Tao, Xiaoyong Shen, and Jiaya Jia. Wide-context semantic image extrapolation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1399–1408, 2019.

[4] Shai Avidan and Ariel Shamir. Seam carving for content-aware image resizing. In *ACM SIGGRAPH 2007 Papers*, SIGGRAPH '07, San Diego, California, July 2007. Association for Computing Machinery.

[5] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. PatchMatch: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics*, 28(3):24:1–24:11, July 2009.

[6] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2536–2544, 2016.

[7] Miao Wang, Yukun Lai, Yuan Liang, Ralph Robert Martin, and Shi-Min Hu. Biggerpicture: data-driven image extrapolation using graph matching. *ACM Transactions on Graphics*, 33(6), 2014.

[8] Yinda Zhang, Jianxiong Xiao, James Hays, and Ping Tan. Framebreak: Dramatic image extrapolation by guided shift-maps. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1171–1178, 2013.

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

[10] Shai Avidan and Ariel Shamir. Seam carving for content-aware image resizing. In *ACM Transactions on graphics (TOG)*, volume 26, page 10. ACM, 2007.

[11] Zongxin Yang, Jian Dong, Ping Liu, Yi Yang, and Shuicheng Yan. Very Long Natural Scenery Image Prediction by Outpainting. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 10561–10570, 2019.

[12] Chuanxia Zheng, Tat-Jen Cham, and Jianfei Cai. Pluralistic Image Completion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1438–1447, 2019.

[13] Qi Shan, Brian Curless, Yasutaka Furukawa, Carlos Hernandez, and Steven M Seitz. Photo uncrop. In *European Conference on Computer Vision*, pages 16–31. Springer, 2014.

[14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014.

[15] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2019.

[16] Xian Wu, Rui-Long Li, Fang-Lue Zhang, Jian-Cheng Liu, Jue Wang, Ariel Shamir, and Shi-Min Hu. Deep Portrait Image Completion and Extrapolation. *IEEE Transactions on Image Processing*, 29:2344–2355, 2020.

[17] Weicheng Kuo, Anelia Angelova, Jitendra Malik, and Tsung-Yi Lin. ShapeMask: Learning to Segment Novel Objects by Refining Shape Priors. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9207–9216, 2019.

[18] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, Tom Duerig, and Vittorio Ferrari. The Open Images Dataset V4: Unified image classification, object detection, and visual relationship detection at scale. *International Journal of Computer Vision*, March 2020.

[19] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Globally and locally consistent image completion. *ACM Transactions on Graphics (ToG)*, 36(4):1–14, 2017.

[20] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. *arXiv preprint arXiv:1801.07892*, 2018.

[21] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S. Huang. Free-Form Image Inpainting With Gated Convolution. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4471–4480, 2019.

[22] Guilin Liu, Fitsum A. Reda, Kevin J. Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. Image inpainting for irregular holes using partial convolutions. In *The European Conference on Computer Vision (ECCV)*, 2018.

[23] Oleg Ivanov, Michael Figurnov, and Dmitry Vetrov. Variational autoencoder with arbitrary conditioning. *arXiv preprint arXiv:1806.02382*, 2018.

[24] A. Agarwala et al. Interactive digital photomontage. *SIGGRAPH*, 23(3):292–300, 2004.

[25] Matthew Brown and David G Lowe. Automatic panoramic image stitching using invariant features. *International journal of computer vision*, 74(1):59–73, 2007.

[26] Richard Szeliski. Image alignment and stitching: A tutorial. *Foundations and Trends® in Computer Graphics and Vision*, 2(1):1–104, 2006.

[27] Charles Herrmann, Chen Wang, Richard Strong Bowen, Emil Keyder, Michael Krainin, Ce Liu, and Ramin Zabih. Robust image stitching with multiple registrations. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 53–67, 2018.

[28] Charles Herrmann, Chen Wang, Richard Strong Bowen, Emil Keyder, and Ramin Zabih. Object-Centered Image Stitching. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision – ECCV 2018*, Lecture Notes in Computer Science, pages 846–861, Cham, 2018. Springer International Publishing.

[29] Nan Li, Tianli Liao, and Chao Wang. Perception-based seam cutting for image stitching. *Signal, Image and Video Processing*, 12(5):967–974, 2018.

[30] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein Generative Adversarial Networks. In *International Conference on Machine Learning*, pages 214–223, July 2017.

[31] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved Training of Wasserstein GANs. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5767–5777. Curran Associates, Inc., 2017.

[32] Dongsheng Guo, Hongzhi Liu, Haoru Zhao, Yunhao Cheng, Qingwei Song, Zhaorui Gu, Haiyong Zheng, and Bing Zheng ;. Spiral generative network for image extrapolation. In *European Conference on Computer Vision (ECCV)*, August 2020.

[33] Lingzhi Zhang, Jiancong Wang, and Jianbo Shi. Multimodal Image Outpainting With Regularized Normalized Diversification. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 3433–3442, 2020.

[34] Yijun Li, Lu Jiang, and Ming-Hsuan Yang. Controllable and progressive image extrapolation. *arXiv preprint arXiv:1912.11711*, 2019.

[35] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

[36] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018.

[37] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

[38] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-To-Image Translation With Conditional Adversarial Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1125–1134, 2017.

[39] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*, 2018.

[40] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8798–8807, 2018.

[41] Rodrigo Benenson, Stefan Popov, and Vittorio Ferrari. Large-scale interactive object segmentation with human annotators. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[42] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common Objects in Context. In David Fleet, Tomas Pajdla, Bernt

Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, Lecture Notes in Computer Science, pages 740–755, Cham, 2014. Springer International Publishing.

[43] R. Malladi, J. A. Sethian, and B. C. Vemuri. Shape modeling with front propagation: a level set approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(2):158–175, 1995.

[44] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[45] Ping Hu, Bing Shuai, Jun Liu, and Gang Wang. Deep level sets for salient object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

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

[47] Matteo Ruggero Ronchi and Pietro Perona. Describing common human visual actions in images. In Mark W. Jones Xianghua Xie and Gary K. L. Tam, editors, *British Machine Vision Conference (BMVC)*, pages 52.1–52.12. BMVA Press, September 2015.

[48] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 6626–6637. Curran Associates, Inc., 2017.

[49] Kamyar Nazeri, Eric Ng, Tony Joseph, Faisal Z. Qureshi, and Mehran Ebrahimi. EdgeConnect: Generative Image Inpainting with Adversarial Edge Learning. *arXiv:1901.00212 [cs]*, January 2019.