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 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--
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 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 de-occlusion, 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 groundtruth 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

---

1In 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.

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

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

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:

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

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

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

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

\[
f(x) = \begin{cases} 
\min_{s \in S} d(x, s) & x \not\in S \\
-\min_{s \in S} d(x, s) & x \in S 
\end{cases}
\]

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.
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.
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.

### 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


