SemIE: Semantically-aware Image Extrapolation

Bholeshwar Khurana[1,2*] Soumya Ranjan Dash[1,2*] Abhishek Bhatia[1,2*]
Aniruddha Mahapatra[2] Hrituraj Singh[2,3] Kuldeep Kulkarni[2]

[1]IIT Kanpur [2]Adobe Research India [3]Triomics

{bkhurana, sodash, abbhatia, anmahapa, kulkulka}@adobe.com, hrituraj@triomics.in

Abstract

*We propose a semantically-aware novel paradigm to perform image extrapolation that enables the addition of new object instances. All previous methods are limited in their capability of extrapolation to merely extending the already existing objects in the image. However, our proposed approach focuses not only on **(i)** extending the already present objects but also on **(ii)** adding new objects in the extended region based on the context. To this end, for a given image, we first obtain an object segmentation map using a state-of-the-art semantic segmentation method. The, thus, obtained segmentation map is fed into a network to compute the extrapolated semantic segmentation and the corresponding panoptic segmentation maps. The input image and the obtained segmentation maps are further utilized to generate the final extrapolated image. We conduct experiments on Cityscapes and ADE20K-bedroom datasets and show that our method outperforms all baselines in terms of FID, and similarity in object co-occurrence statistics. Project url:* https://semie-iccv.github.io/

Figure 1. Illustration of our results. Dotted white rectangle refers to the input image. Our method not only extrapolates the objects present in the input but also generates new objects (blue bounding boxes) while maintaining the texture consistency.

1. Introduction

Image extrapolation or out-painting refers to the problem of extending an input image beyond its boundaries. While the problem has applications in virtual reality, sharing photos on social media like Instagram, and even generating scenes during game development especially if the scenes are repetitive, it is relatively under-explored compared to the image inpainting counterpart, which has been extensively researched. Image inpainting solutions based on deep networks and generative adversarial networks (GANs), when applied to the out-painting problem, have been shown to yield poor results [36]. This has led to researchers exploring and proposing new solutions to the out-painting problem [47, 41, 38]. However, the solutions have been mainly restricted to images that involve outdoor domains like natural scenes where the problem is limited to just extending the existing textures for 'stuff' classes like mountains, water, trees [11, 36] or single-object images of classes like faces, flowers, and cars. These methods are not suitable to other domains like traffic scenes and indoor scenes where a desirable image extrapolation necessitates 1) extending not only the 'stuff' classes but also the 'things' classes like cars, persons, beds, tables that have very definite structure as well as 2) adding new objects based on the context that were not present in the original image. So, why cannot we use the existing techniques [47, 41, 38, 36] for such domains? The answer is they fail spectacularly by filling the extrapolated region with artifacts (see figures 5 and 6). They attempt to extrapolate the image by capturing the low-level statistics like textures and colors from the input image while ignor-

*BK, SRD, AB contributed to the work during their Adobe internship.

ing the high-level information like object semantics and object co-occurrence relationships. In short, they are limited in their ability to perform satisfactory image extrapolation that demands the creation of new object instances and the extension of multiple objects from diverse classes.

We address the shortcomings of the previous works by extrapolating the image in the semantic label map space, which enables us to generate new objects in the extrapolated region. Additionally, semantic label maps belong to a lower dimensional manifold than images, making it easier to extrapolate them. However, just having a semantic label map does not allow us to have control over every instance in the extrapolated image. We propose to generate an estimate of the panoptic label directly from the extrapolated semantic label map, different from [20, 4]. Instance boundary maps obtained from panoptic labels also help in creating crisper boundaries between objects belonging to the same semantic category. Unlike semantic label map to image generation [31, 15, 37], we have to maintain texture consistency between the input region and the extrapolated output. To account for this, we propose Instance-aware context normalization (IaCN), which leverages the estimated panoptic label maps to transfer average color information as a feature map for texture consistency in the extrapolated object instances. In addition, we propose the use of patch co-occurrence discriminator [32] to maintain global texture similarity in input and extrapolated region.

Our contributions can be summarized below:

- We propose a novel paradigm for image out-painting by extrapolating the image in the semantic label space to generate novel objects in the extrapolated region.
- We propose the generation of panoptic label maps from the extrapolated semantic label maps to facilitate the generation of high quality object boundaries in the extrapolated image.
- We propose Instance-aware Context Normalization (IaCN) and the use of patch co-occurrence discriminator to maintain texture consistency of extrapolated instances.

Through extensive experiments on Cityscapes and ADE20K datasets, we show that our method outperforms all previous state-of-the-art methods in terms of FID and similarity in object co-occurrence metrics.

2. Related Work

Image Extrapolation: Prior works in image synthesis have had great breakthroughs in image inpainting [25, 42, 44] , conditional image synthesis [15, 26, 31, 37, 39], and unconditional image synthesis [1, 28, 7]. On the contrary, image extrapolation models have been relatively less successful. The works on image extrapolation can be broadly classified on whether they use non-parametric methods or parametric methods. Several non-parametric methods [8, 9] have been able to perform only a limited peripheral texture extension. Furthermore, their heuristics do not capture the variation in color, texture and the information of shape and structure of an object. These methods [9] limit themselves to simple pattern extrapolation and are very brittle to increasing extrapolation. Other classical approaches [3] leverage patch matching to extrapolate the image. However, these methods are limited in their ability to generate new objects or hallucinate new textures. With the advent of GAN [10] based approaches, significant progress has been made in image extrapolation. [47, 41, 36] use a single-stage method to extrapolate the input image. Most of these works deal with scene completion using object completion or merely extending the significant texture near the image boundary. [38] proposed a method of feature expansion from input region to predict context for the extrapolated image, while [12] generates the extrapolated image by incrementally extending image on each side using a generated reference image. Consequently, as we go further from the input boundary, the relative volume information from input reduces resulting in the generation of substandard extrapolation. Moreover, all of these approaches currently lack the semantic understanding of the scene and semantic structure of objects in the scene.

Semantic Editing for Image Manipulation: Recently, there have been a few works that manipulate images by editing in the semantic label space. [2] concentrates on synthesizing images using semantic label space but in an unconditional setting, where the semantic label space is generated from scratch from a random seed. [14, 22] proposed methods to insert an object in an image by editing the semantic label of the input image, given the class information and the bounding box of the object. Such methods are unsuitable to be adapted for image out-painting as we do not have the class information and the bounding boxes for the new objects to be inserted in the extrapolated region. Another drawback of such methods is that they require as many forward passes as the number of the new objects and hence are not scalable in terms of time complexity. [30] proposes to edit an image by allowing the user to provide a semantic guideline of the regions to be manipulated. Different from the above works, our image extrapolation method automatically extrapolates the semantic label map of the image without any user input and estimates the corresponding panoptic label map for the extrapolated image to be generated. The closest work to ours is an in-painting model, SPGNet [34] where the hole in an image is firstly filled in the semantic label space before the final image is generated. However, unlike SPGNet, we generate an estimate of panoptic label map, along with the semantic extrapolation, and further leverage it to ensure texture transfer for the extrapolated instances and sharp instance boundaries in the final image. We show through experiments that our method is significantly better than SPGNet.

3. Our Method

Our goal is to extrapolate a given image $\mathbf{X} \in \mathbb{R}^{h \times w \times c}$ on its periphery using a sequence of deep neural networks. $\mathbf{Y} \in \mathbb{R}^{h_1 \times w_1 \times c}$ is the extrapolated image where $h_1 \geq h$ and $w_1 \geq w$. Here, c represents the number of channels corresponding to the image, which is 3 for an RGB image. The pipeline shown in figure 2 involves four major stages:

- Image segmentation: Generation of semantic label map from the input image.
- Semantic label map extrapolation: Extend periphery in the semantic label space.
- Panoptic label generation: The semantic label map is processed to obtain an apriori estimate of corresponding panoptic label map.
- Instance-aware image synthesis: Generation of image from the semantic label map and panoptic label map by leveraging the proposed IaCN module and patch co-occurrence discriminator.

3.1. Image Segmentation

Given an image $\mathbf{X} \in \mathbb{R}^{h \times w \times c}$, corresponding one-hot vector for semantic label map $\mathbf{L_1} \in \{0,1\}^{h \times w \times c_1}$ can be obtained using state-of-the-art segmentation techniques [48, 35, 4, 46, 43]. For our method, we use PSPNet [48].

3.2. Semantic Label Extrapolation

We train a network, dubbed 'Peripheral Object Generation (POGNet)', $G_\mathbf{S}$ to semantically extrapolate $\mathbf{L_1}$ and obtain an estimate of the semantic label map, $\mathbf{L_2}$ of the final extrapolated image to be generated. In addition to generating $\mathbf{L_2}$, we also output the corresponding instance boundary channel. Although [34] uses input image with semantic label map to generate extrapolated semantic label map, having explicit supervision with ground truth instance boundary map acts as a better regularizer during training for obtaining more precise object shapes. POGNet is trained using a multi-scale discriminator as proposed in [37], enabling $G_\mathbf{S}$ to capture the object co-occurrence information at various scales.

Adversarial Loss: Instead of regular GAN loss [10], we use LS-GAN loss [27] ($\mathcal{L}_{GAN}$).

Focal Loss: We use focal loss to compute the discrepancy between the ground truth semantic label map and the output of the POGNet. By giving higher weight to hard-to-generate object classes, focal loss allows us to generate some of the rare classes. The focal loss between the ground-truth and the output at any location is given as:

$$l(z, y) = -y \times log(z)$$

$$\mathcal{L}_{CE}(z, y) = \Sigma_{h,w,c} l(z, y)$$

$$\mathcal{L}_{FL}(z, y) = \Sigma_{h,w,c} l(z, y) \times (1-z)^\gamma$$

The final focal loss, $\mathcal{L}_{FL}^{all}$ is given by the sum of focal losses across all locations in the semantic label map. We use the following training objective for semantic label map extrapolation (we show only the generator losses here):

$$\mathcal{L}_{gen} = \mathcal{L}_{GAN} + \mathcal{L}_{FM} + \lambda_{FL}\mathcal{L}_{FL}^{all} + \lambda_{CE}\mathcal{L}_{CE}, \quad (1)$$

where $\mathcal{L}_{CE}$ is the cross-entropy loss between the ground-truth instance boundary and the corresponding output channel in POGNet and $\mathcal{L}_{FM}$ is the discriminator feature matching loss. More details can be found in the supplementary.

3.3. Panoptic Label Map Generation

As mentioned earlier, we wish to estimate the panoptic label maps (for the to-be-generated extrapolated image) that can be leveraged for IaCN module (discussed in 3.4) as well as obtain crisp and precise boundaries between the object instances. Traditionally, the panoptic label maps are generated from the images. But how do we estimate panoptic label maps, apriori, without knowing the image itself? We adapt the method elucidated in Panoptic-DeepLab [4] by predicting the class-agnostic pixel-wise instance center maps and off-set maps from the instance centers for objects belonging to 'things' classes, directly from the semantically extrapolated map, i.e the output of POGNet. Specifically, we train a generator-only network that takes in the extrapolated segmentation map and produces heat maps for instance centers and the pixel-wise offsets from the nearest instance center. The center heat-maps and the offset outputs are further processed along with the segmentation map to obtain the instance maps. The ground-truth center maps are represented by Gaussian blobs of standard deviation of 8 pixels, centered at the instance centers. We use L_2 loss to compute the instance center loss and L_1 loss to compute the offset losses. The final loss for stage-3 is the weighted sum of the center loss and the offset losses. During the test time, we adapt the procedure mentioned by [4] to group the pixels based on the predicted centers and off-sets to form instance masks. The instance masks and the semantic label map (the input to stage-3) are combined by majority voting to obtain the panoptic label map. An expanded version of the details of training of the network and post-processing are provided in the supplementary material.

3.4. Instance-aware Image Synthesis

This is the final stage (stage-4) which converts the extrapolated semantic label map back into a colored image. This stage takes in input $\mathbf{X}'(\in \mathbb{R}^{h_1 \times w_1 \times c'})$ (Figure 2), which is concatenation of the extrapolated semantic label map obtained from the second stage, the cropped (input) image, the boundary map obtained using the panoptic label map obtained from the previous stage and the feature map

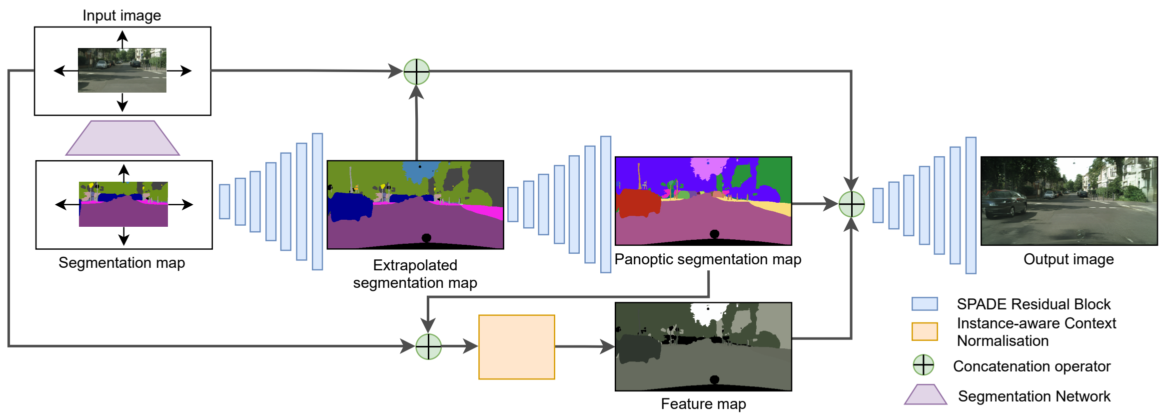


Figure 2. **Overview of the pipeline:** Stage1: The input image is fed into a pre-trained segmentation network to obtain its label map. Stage2: The stage 1 output fed into a network to obtain the extrapolated label map. Stage3: The extrapolated label map is fed into another network to obtain the panoptic label map. Stage4: The input image, extrapolated label map and the panoptic label map are used in conjunction with Instance-aware context normalization module to obtain the final extrapolated image.

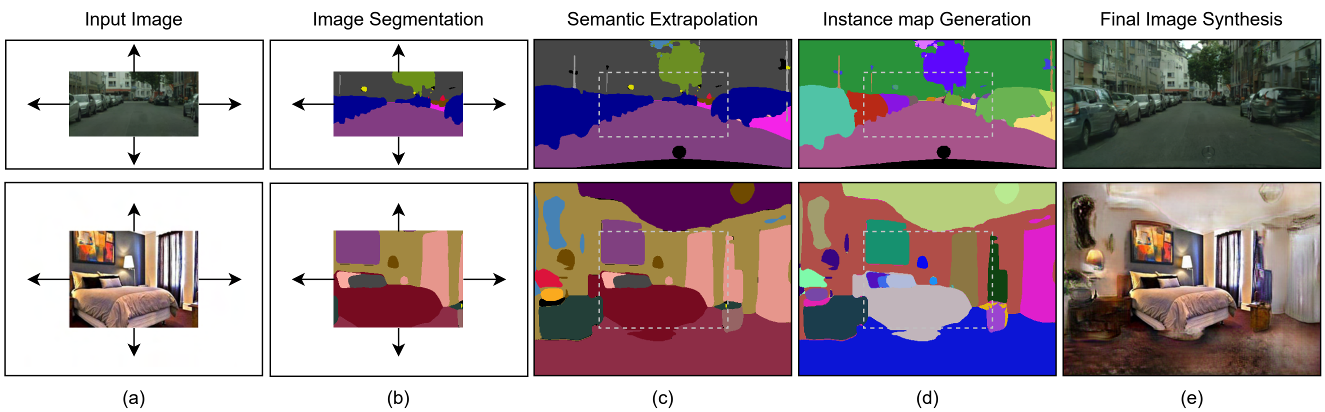


Figure 3. **Stage-wise results:** The input (cropped) image is converted to semantic label map in stage-1, which is then extrapolated in stage-2 to form the outpainted semantic label map. Panoptic label maps are generated from this semantic label map in stage-3. The input image, the (outpainted) semantic label map and the panoptic label map are used to synthesize the final image in stage-4.

obtained using the proposed Instance-aware Context Normalization. The output is an RGB image $\mathbf{Y} \in \mathbb{R}^{h_1 \times w_1 \times 3}$.

This is different from prior conditional GANs problems [15, 26, 31, 37] since they synthesize RGB images from semantic label maps, but we have to synthesize RGB images from semantic label maps, given some pixel information of the to-be-synthesized RGB image, which is the cropped image in our case. Here, we have to take care of texture consistency in the synthesized image while maintaining an identity mapping from the cropped image to the final image. To maintain this texture consistency for the extrapolated instances, we concatenate the feature maps to the input, which are generated using Instance-aware Context Normalisation module.

Generator

We use SPADE [31] normalization residual blocks for each of the layers in the generator. We use similar learning objective functions, as used in SPADE [31] and pix2pixHD [37]: GAN loss with hinge-term [24, 29, 45] ($\mathcal{L}_{GAN}$), Feature matching loss [37] based on the discriminator ($\mathcal{L}_{FM}$) and VGGNet [33] for perceptual losses [6, 16] ($\mathcal{L}_{VGG}$)

Instance-aware Context Normalization (IaCN)

Outpainting-SRN [38] proposed Context Normalization (CN) to maintain texture consistency between the inside (cropped) region and the outside (outpainted) region. It involves transferring the mean feature or color from the inside region to the outside region. However, we believe that transferring this input mean color directly to the outside region is not suitable for images which have very diverse object instances (like outdoor images, street images).

To this end, we propose Instance-aware Context Normalization (IaCN) (Figure 2), which takes as input the cropped image and the instance map. IaCN module computes the mean color using the input (cropped) image for all the partial instances. Partial instances refer to the instances which get extrapolated in the final image. Since the problem with

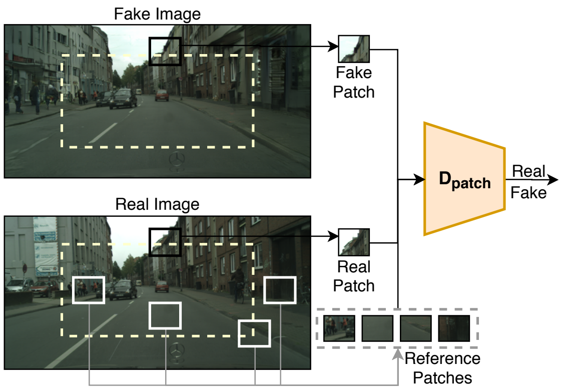


Figure 4. **Patch Discriminator**: D_{patch} takes in input 4 reference patches, a fake patch and a real patch. The reference patches are randomly selected from the real image. The fake patch and real patch are the same patches, randomly selected but made sure that some part of them is inside while other part is outside, from fake image and real image respectively. The discriminator tries to distinguish between fake patch and the real patch, making use of the reference patches. All the patches are of size 64×64.

texture consistency occurs only for partial instances, therefore we compute features only for them. These computed feature maps are then concatenated to the input.

Discriminators

We propose to use two discriminators, i) a traditional image discriminator (multi-scale discriminator) that attempts to differentiate between the real and the fake image, ii) a patch co-occurrence discriminator similar to [32], which employed a patch co-occurrence discriminator to ensure texture transfer [17, 40] from an input image to the target image to be edited. We employ a similar idea wherein the region in the image that needs to be extrapolated takes the role of the target image (equation 2). This facilitates consistent texture transfer from the inside region (source) to the extrapolated region (target) (illustrated in Figure 4).

$$\begin{aligned}&\mathcal{L}_{CooccurGAN}(G, D_{patch}) = \\ &\mathbb{E}_{x,y}[-log(D_{patch}(crop(G(x)), crop(y), crops(y)))]\end{aligned} \tag{2}$$

Here x is the input and y is the corresponding ground-truth image. $crop(y)$ function takes a random patch from image y and $crops(y)$ takes 4 random patches from image y, which serve as the reference patches.

The details of the network architectures for all generators and discriminators for the various stages are provided in the supplementary material.

Variational Autoencoder

To ensure appropriate style transfer, we use an encoder that processes the cropped image, which is then fed to the generator. We use the encoder used in [31]. This encoder forms a VAE [19] with the generator. In the objective function, we add a KL-Divergence Loss term [19] ($\mathcal{L}_{KLD}$).

Final Objective

The training objective is as shown below in equation 3:

$$\begin{aligned}\min_G\{&\mathcal{L}_{GAN} + \lambda_{FM}\mathcal{L}_{FM} + \lambda_{VGG}\mathcal{L}_{VGG} \\ &+ \lambda_{KLD}\mathcal{L}_{KLD} + \mathcal{L}_{CooccurGAN}\}\end{aligned} \tag{3}$$

4. Experiments

We evaluate the proposed approach on two different datasets which have a sufficient disparity between each other to show that our approach is fairly robust and is applicable to diverse scenes. We utilize the publicly available Cityscapes [5] and ADE20K-bedroom subset [49] datasets both of which contain large variety of distinct object categories. While Cityscapes comprises of outdoor street images, ADE20K bedroom subset consists of bedroom scenes. The ADE20K processed subset[1] consists of 31 classes including bed, lamp, wall, floor and table. Cityscapes consists of 2975 training images and 500 validation images. Each image has its corresponding semantic label map and instance label map along with the original image. The bedroom subset of ADE20K [49] has 1389 images in the training set and 139 in the validation set. In order to limit the size of our model, we downsample the images in Cityscapes to a resolution of 256×512 and the ADE20K bedroom by resizing all its images to a standard size of 384×512 while training. For both the datasets, the input image is taken as centre crop of resized image with half the height and width.

Implementation details

We train PSPNet [48] on Cityscapes as well as ADE20K bedroom subset at the resolution discussed earlier and use them to generate segmentation maps of the input (cropped) images. We adopt cGAN based generator for stage1, stage2 and stage4 models similar to SPADE [31]. In stage2 we replace $tanh$ with $sigmoid$ activation in the final layer to produce one hot encodings and semantic label map. For the training of stage2, in our final objective (Eq. 1), we use $\lambda_{FL} = 5$, $\lambda_{CE} = 5$ and $\gamma = 5$. For the training of stage4, we use the same weights for loss terms as [31], i.e. $\lambda_{FM} = 10, \lambda_{VGG} = 10$ and $\lambda_{KLD} = 0.05$ in Eq 3. We use ADAM solver [18] with $\beta_1 = 0$ and $\beta_2 = 0.9$ for both the stages. The training is done for 200 epochs.

Baselines

We compare our method with various baselines both in quantitative (with FID and Similarity in Object Co-Occurrence metrics) and qualitative terms. We compare the proposed approach with five baselines 'Outpainting-SRN' [38], 'Boundless' [36], 'SPGNet' [34], 'SPGNet++' and partial convolutions ('PConv') [25]. 'PConv' [25] was originally proposed for image inpainting, like in [36], we adapt it for the task of image out-painting in our setting. We also

[1] To obtain the processed subset, we contacted the authors of [21].

Figure 5. **Cityscapes dataset:** Our method is able to generate new objects in the extrapolated region leading to realistic image extrapolation. Except ours and SPGNet, all other methods fail to generate new objects in the extrapolated region.

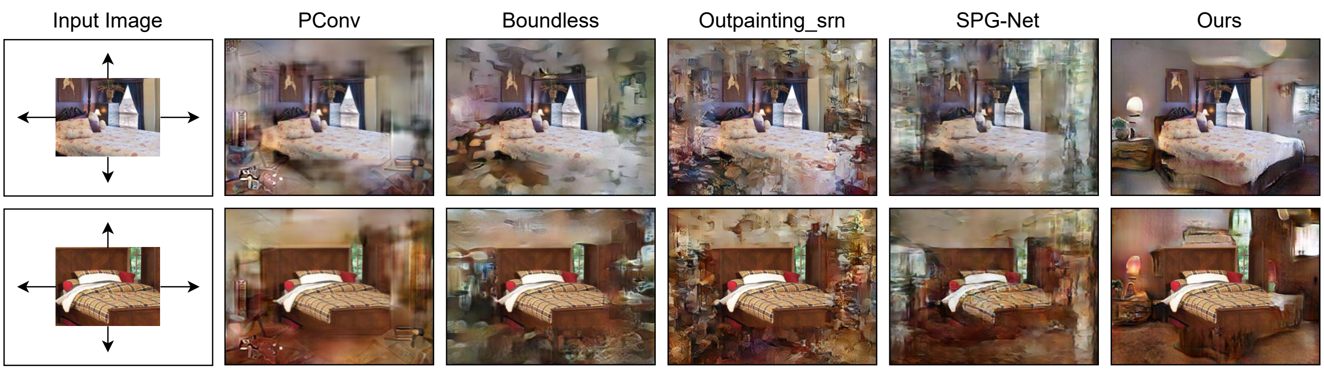


Figure 6. **ADE20K dataset:** Our method is able to generate new objects in the extrapolated region leading to realistic image extrapolation. Only our, all other methods try to copy texture patches from inside region in the extrapolated region.

use SPGNet [34] as a baseline since it also operates in semantic label space but for image inpainting task; but we adapt it for out-painting task. We create a modified version of SPGNet, SPGNet++ using [31] base generator and multiscale discriminator used in our method while retaining the exact training procedure and loss functions used in [34] and use it to compare with our method. We train these baselines on our kind of input-crop (25% of the original image).

Evaluation Metrics

To compare the perpetual quality of the generated RGB image we use Frêchet Inception distance (FID) [13] metric. However, since we additionally focus on generation of new objects in the extrapolated region, we also evaluate the results in semantic label space using similarity in object co-occurrence (SOCC) statistics [23].

FID: It is a standard metric used to calculate the fidelity of GAN generated images and provides a measure of the distance between the generated images and the real images.

SOCC: The co-occurrence measure for two classes c_a and c_b can be calculated as the ratio of the number of times they occur together to the total number of times one of them occurs in the entire dataset. Let N_{c_a} represent the frequency of a class c_a in the input image, and $N_{c_{ab}}$ be the number of times there is atleast one instance of class c_b present in the extrapolated region, given c_a is present in the input. The probability of co-occurrence $p(c_a, c_b)$ of the two classes can be calculated as $\frac{N_{c_{ab}}}{N_{c_a}}$. The similarity in co-occurrence probability of a pair of classes between generated outputs and the training set, therefore, reflects the extent of faithful emulation of scene distribution. The similarity in co-occurrence for class c_2 in the output to the training set, given c_1 is present in the output, is defined as $s(c_a, c_b) = 1 - |p_{train}(c_a, c_b) - p_{gen}(c_a, c_b)|$. The closer is this score to 1, the greater is the similarity between the outputs of the model and the training set images.

4.1. Qualitative performance

In figure 3, we show the various stage-wise results of our pipeline. In figures 5 and figure 6, we compare our results with the baselines for the Cityscapes and ADE20K dataset respectively. It can be seen that our method not only extrapolates the existing objects, ensuring texture and structural consistency but is also capable of adding very precise novel objects, which the baselines fail to do. Almost all the baseline methods that operate on RGB space (except [34]) have trivial block like patches in the extrapolated region which is more noticeable in ADE20K dataset results.

Method	(Bed, Lamp)	(Wall, Window)	(Bed, Curtain)	(Floor, Table)	(Wall, Painting)
Outpainting-SRN	0.66	0.82	**0.94**	0.77	0.64
Boundless	0.79	0.82	0.87	0.75	0.76
Pconv	0.75	0.85	0.83	0.77	0.83
SPGNet	0.77	0.53	0.51	0.84	0.82
SPGNet++	0.79	0.87	0.85	0.81	0.83
Ours	**0.82**	**0.90**	0.84	**0.87**	**0.84**

Table 1. **Results**: Similarity in object co-occurrence scores (higher is better) for our method vs the baselines on ADE20K-bedroom dataset.

Method	(Parking, Car)	(Person, Person)	(Pole, Traffic Light)	(Person, Rider)	(Car,Sidewalk)
Outpainting-SRN	0.85	0.89	0.68	0.91	0.85
Boundless	0.82	0.89	**0.99**	0.94	0.82
Pconv	0.83	0.88	0.57	0.9	0.83
SPGNet	0.91	0.84	0.87	0.86	0.91
SPGNet++	0.94	0.87	0.93	0.94	0.93
Ours	**0.96**	**0.92**	0.96	**0.96**	**0.94**

Table 2. **Results**: Similarity in object co-occurrence scores (higher is better) for our method vs the baselines on Cityscapes dataset.

Method	Cityscapes	ADE20K
Pconv	86.82	147.14
Boundless	77.36	136.98
Outpainting-SRN	66.89	140.98
SPGNet	83.84	197.69
SPGNet++	52.14	97.23
Ours	**47.67**	**90.45**

Table 3. **Results**: FID scores (lower is better) for our method vs the baselines on Cityscapes and ADE20K-bedroom dataset.

4.2. Quantitative performance

Table 3 shows the FID scores of our method compared to the baselines on the two datasets. Note that all these scores are on the validation split of the two datasets. We outperform all the baselines by very significant margins.

Table 2, 1 show the SOCC scores for different pairs of object classes in both the datasets. Our method is able to generate results that consistently resemble the object co-occurrence statistics for most class pairs in the datasets.

5. Ablation study

In this section, we discuss the importance of individual components of the proposed approaches in our pipeline. Table 4 shows the FID scores for variants of our method on cityscapes dataset.

5.1. Use of Boundary maps as an extra channel for semantic extrapolation

As discussed in Section 3.2, we use the semantic class boundary map to enforce the object shape information into the network during the training time. In Figure 7, while other approaches (SPG-Net and SPG-Net++) resulted in blobs representing newly generated instances, the shapes of those instances are much better when enforced with boundary maps during training.

Method	FID
SPGNet++	52.14
Our (base w/o D_{patch} w/o IaCN)	48.77
Our (w/ D_{patch} w/o IaCN)	48.72
Our (w/o D_{patch} w/ IaCN)	47.76
Our (final)	**47.67**

Table 4. **Ablation**: FID ablation study on cityscapes dataset

5.2. Use of Panoptic label maps (Stage3)

We use the synthesized panoptic label maps to obtain (i) boundary maps and (ii) the feature maps for IaCN module. The boundary maps are used to generate crisp boundaries between instances of the same class (Figure 8).

Use of IaCN module

As discussed in Section 3.4, we use the feature maps generated by IaCN to preserve the texture of the extrapolated instances. Figure 9 shows that the relative texture of the extrapolated instances for both 'things' (van) and 'stuff' (tree) class are maintained when IaCN is used.

5.3. Use of patch co-occurrence discriminator

Figure 10 shows that sharper images are produced when patch co-occurrence discriminator is used. It also ensures much better consistent textures at the boundary of input and extrapolated regions.

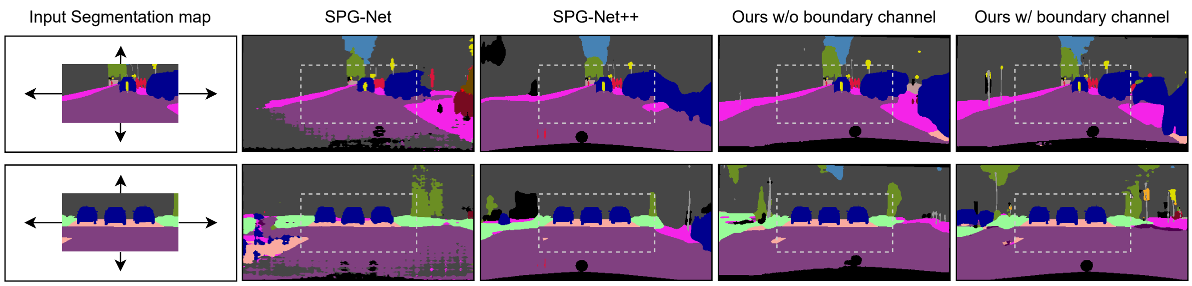
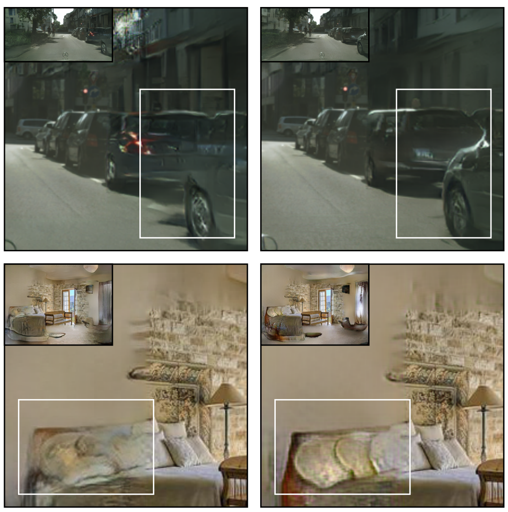


Figure 7. **Semantic label extrapolation ablation**: Shapes of the extrapolated or newly synthesized instances are more realistic when boundary channel is used.

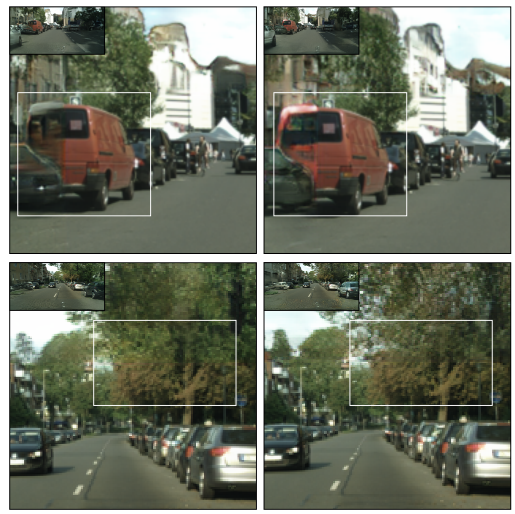


(a) Ours w/o instance boundary map (b) Ours w/ instance boundary map

Figure 8. **Instance boundary map ablation**: Crisp boundaries between instances are clearly visible in the region highlighted with white box between (a) 2 cars (Cityscapes) (b) pillows on the bed (ADE20K), when boundary maps derived from the estimated panoptic label maps are used. We show the complete extrapolated image at the top-left of each image.

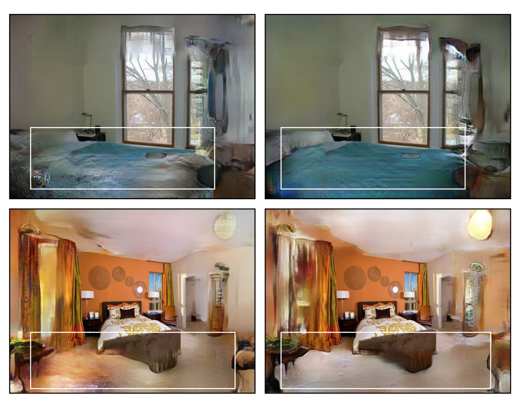
6. Discussions and Conclusion

We propose a new solution for image extrapolation that is amenable for adding novel objects as well extending the existing objects and textures. Our solution distinguishes itself from all previous works in the image extrapolation by extrapolating the image in semantic label space. We show in the paper that this helps us achieve our objective of adding new objects. We also propose the generation of panoptic label maps from just segmentation maps, which enables us to create multiple instances of the same classes and as well allow us to have control over the instances thus created. We show in our supplementary video how our method can be recursively applied to generate image extrapolations of arbitrary dimensions. We hope our work encourages researchers to develop solutions for image editing in semantic label space.



(a) Ours w/o IaCN (b) Ours w/ IaCN

Figure 9. **IaCN ablation**: Texture is consistently transferred for extrapolated instances in the region highlighted with white box in (a) red van ('things' class) (Cityscapes) (b) yellowish-green tree ('stuff' class) (Cityscapes), when IaCN is used. The complete extrapolated image is present at top-left of each image.



(a) Ours w/o D_{patch} (b) Ours w/ D_{patch}

Figure 10. D_{patch} **ablation**: Sharper images with consistent texture are produced in the region highlighted with white box in (a) blue bed-sheet (ADE20K) (b) off-white floor (ADE20K) when patch co-occurrence discriminator is used.

References

[1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR, 2017. 2

[2] Samaneh Azadi, Michael Tschannen, Eric Tzeng, Sylvain Gelly, Trevor Darrell, and Mario Lucic. Semantic bottleneck scene generation. *arXiv preprint arXiv:1911.11357*, 2019. 2

[3] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Trans. Graph.*, 28(3):24, 2009. 2

[4] Bowen Cheng, Maxwell D Collins, Yukun Zhu, Ting Liu, Thomas S Huang, Hartwig Adam, and Liang-Chieh Chen. Panoptic-deeplab: A simple, strong, and fast baseline for bottom-up panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12475–12485, 2020. 2, 3

[5] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016. 5

[6] A. Dosovitskiy and T. Brox. Generating images with perceptual similarity metrics based on deep networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2016. 4

[7] Ishan Durugkar, Ian Gemp, and Sridhar Mahadevan. Generative multi-adversarial networks. *arXiv preprint arXiv:1611.01673*, 2016. 2

[8] Alexei A Efros and William T Freeman. Image quilting for texture synthesis and transfer. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 341–346, 2001. 2

[9] A. A. Efros and T. K. Leung. Texture synthesis by non-parametric sampling. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pages 1033–1038 vol.2, 1999. 2

[10] 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. 2, 3

[11] 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*, pages 701–717. Springer, 2020. 1

[12] Dongsheng Guo, Hongzhi Liu, Haoru Zhao, Yunhao Cheng, Qingwei Song, Zhaorui Gu, Haiyong Zheng, and Bing Zheng. Spiral generative network for image extrapolation. In *The European Conference on Computer Vision (ECCV)*, pages 701–717, 2020. 2

[13] 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 *Advances in neural information processing systems*, pages 6626–6637, 2017. 6

[14] Seunghoon Hong, Xinchen Yan, Thomas Huang, and Honglak Lee. Learning hierarchical semantic image manipulation through structured representations. *arXiv preprint arXiv:1808.07535*, 2018. 2

[15] 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. 2, 4

[16] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016. 4

[17] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8110–8119, 2020. 5

[18] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[19] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 5

[20] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9404–9413, 2019. 2

[21] Kuldeep Kulkarni, Tejas Gokhale, Rajhans Singh, Pavan Turaga, and Aswin Sankaranarayanan. Halluci-net: Scene completion by exploiting object co-occurrence relationships. *arXiv preprint arXiv:2004.08614*, 2020. 5

[22] Donghoon Lee, Sifei Liu, Jinwei Gu, Ming-Yu Liu, Ming-Hsuan Yang, and Jan Kautz. Context-aware synthesis and placement of object instances. *arXiv preprint arXiv:1812.02350*, 2018. 2

[23] Manyi Li, Akshay Gadi Patil, Kai Xu, Siddhartha Chaudhuri, Owais Khan, Ariel Shamir, Changhe Tu, Baoquan Chen, Daniel Cohen-Or, and Hao Zhang. Grains: Generative recursive autoencoders for indoor scenes. *ACM Transactions on Graphics (TOG)*, 38(2):1–16, 2019. 6

[24] Jae Hyun Lim and Jong Chul Ye. Geometric gan. *arXiv preprint arXiv:1705.02894*, 2017. 4

[25] 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 *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 85–100, 2018. 2, 5

[26] Xihui Liu, Guojun Yin, Jing Shao, Xiaogang Wang, et al. Learning to predict layout-to-image conditional convolutions for semantic image synthesis. In *Advances in Neural Information Processing Systems*, pages 570–580, 2019. 2, 4

[27] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2794–2802, 2017. 3

[28] Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for gans do actually converge? In *International conference on machine learning*, pages 3481–3490. PMLR, 2018. 2

[29] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018. 4

[30] Evangelos Ntavelis, Andrés Romero, Iason Kastanis, Luc Van Gool, and Radu Timofte. Sesame: Semantic editing of scenes by adding, manipulating or erasing objects. In *European Conference on Computer Vision*, pages 394–411. Springer, 2020. 2

[31] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2337–2346, 2019. 2, 4, 5, 6

[32] Taesung Park, Jun-Yan Zhu, Oliver Wang, Jingwan Lu, Eli Shechtman, Alexei Efros, and Richard Zhang. Swapping autoencoder for deep image manipulation. *Advances in Neural Information Processing Systems*, 33, 2020. 2, 5

[33] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 4

[34] Yuhang Song, Chao Yang, Yeji Shen, Peng Wang, Qin Huang, and C-C Jay Kuo. Spg-net: Segmentation prediction and guidance network for image inpainting. *arXiv preprint arXiv:1805.03356*, 2018. 2, 3, 5, 6

[35] Andrew Tao, Karan Sapra, and Bryan Catanzaro. Hierarchical multi-scale attention for semantic segmentation. *arXiv preprint arXiv:2005.10821*, 2020. 3

[36] 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. 1, 2, 5

[37] 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8798–8807, 2018. 2, 3, 4

[38] 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. 1, 2, 4, 5

[39] Yaxiong Wang, Yunchao Wei, Xueming Qian, Li Zhu, and Yi Yang. Sketch-guided scenery image outpainting. *arXiv preprint arXiv:2006.09788*, 2020. 2

[40] Wenqi Xian, Patsorn Sangkloy, Varun Agrawal, Amit Raj, Jingwan Lu, Chen Fang, Fisher Yu, and James Hays. Texturegan: Controlling deep image synthesis with texture patches. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8456–8465, 2018. 5

[41] 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. 1, 2

[42] Raymond A. Yeh, Chen Chen, Teck Yian Lim, Alexander G. Schwing, Mark Hasegawa-Johnson, and Minh N. Do. Semantic image inpainting with deep generative models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 2

[43] Changqian Yu, Jingbo Wang, Changxin Gao, Gang Yu, Chunhua Shen, and Nong Sang. Context prior for scene segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12416–12425, 2020. 3

[44] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5505–5514, 2018. 2

[45] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *International Conference on Machine Learning*, pages 7354–7363. PMLR, 2019. 4

[46] Hang Zhang, Chongruo Wu, Zhongyue Zhang, Yi Zhu, Zhi Zhang, Haibin Lin, Yue Sun, Tong He, Jonas Mueller, R Manmatha, et al. Resnest: Split-attention networks. *arXiv preprint arXiv:2004.08955*, 2020. 3

[47] Xiaofeng Zhang, Feng Chen, Cailing Wang, Ming Tao, and Guo-Ping Jiang. Sienet: Siamese expansion network for image extrapolation. *IEEE Signal Processing Letters*, 27:1590–1594, 2020. 1, 2

[48] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017. 3, 5

[49] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 5