

# Internal Distribution Matching for Natural Image Retargeting

Assaf Shocher\*

Shai Bagon\*

Phillip Isola<sup>†</sup>

Michal Irani\*

\*Dept. of Computer Science and Applied Math, The Weizmann Institute of Science

<sup>†</sup>Computer Science and Artificial Intelligence Lab Massachusetts Institute of Technology

**Project Website:** <http://www.wisdom.weizmann.ac.il/~vision/ingan/>

## Abstract

*Good Visual Retargeting changes the global size and aspect ratio of a natural image, while preserving the size and aspect ratio of all its local elements. We propose formulating this principle by requiring that the distribution of patches in the input matches the distribution of patches in the output. We introduce a Deep-Learning approach for retargeting, based on an Internal GAN (InGAN). InGAN is an image-specific GAN. It incorporates the Internal statistics of a single natural image in a GAN. It is trained on a single input image and learns the distribution of its patches. It is then able to synthesize natural looking target images composed from the input image patch-distribution. InGAN is totally unsupervised, and requires no additional data other than the input image itself. Moreover, once trained on the input image, it can generate target images of any specified size or aspect ratio in real-time.*<sup>1</sup>

## 1. Introduction

The ubiquity of digital displays of various sizes and aspect ratios poses a great challenge for digital media: any image should be readily retargeted to fit any size and aspect ratio. Good visual retargeting changes the global size and aspect ratio of a natural image, while preserving the size and aspect ratio of all its local elements, see Fig. 1 for examples. Simakov et al. formalized this idea via the notion of *bidirectional similarity* [19]. The output of a retargeting algorithm should exhibit *coherence* – the output image should contain only patches that are found in the input image – and, vice versa, *completeness* – the input image should contain only patches that are found in the output. Thus, no artifacts are introduced to the retargeted image and no critical visual information is lost in the process.

One way to satisfy these criteria is to require that the distribution of patches in the input should match the distribution of patches in the output. We therefore propose *distribution matching* as a new objective for visual retargeting,

which goes beyond bidirectional similarity to require not only that all input patches can be found in the output, and vice versa, but that the frequency of finding these patches should also match.

Reframing the problem as distribution matching has the advantage that we can immediately apply recent advances in adversarial learning. In particular, generative adversarial networks (GANs) can be understood as a tool for distribution matching [7]. A GAN maps data sampled from one distribution to transformed data that is indistinguishable from a target distribution,  $G : x \rightarrow y$  with  $x \sim p_x$ , and  $G(x) \sim p_y$ .

An image can be viewed as a set of samples from a distribution over patches, just like an image dataset can be viewed as a set of samples from a distribution over images. In the same way as we can learn a generative model of images in a dataset, we can learn a generative model of patches in a single image. Retargeting, framed as distribution matching, can therefore be achieved by training a GAN to map from an input image to an output image whose patch distribution is indistinguishable from the input's. Unlike most GANs, which map between two different distributions, ours is an automorphism at the patch-level,  $G : x \rightarrow x$ , with  $p_x$  being the distribution of patches in the input image. We call such a GAN an “internal GAN” (*InGAN*), because it is trained to match the internal statistics of a single image [23]. Retargeting is achieved by modifying the size and aspect ratio of the output tensor, which changes the arrangement of patches, but not the distribution of patches.

Although this formulation is sufficient in theory to encourage both coherence and completeness, in practice we observe that completeness is often not achieved – many patches from the input image are omitted in the output. To ameliorate this, we introduce a second mechanism for encouraging completeness: it should be possible to reconstruct (“decode”) the input image from the output, i.e.  $\|F(G(x)) - x\|$  should be small, where  $F$  is a second network trained to perform the reverse mapping. This objective encourages the mapping between input and retargeted output to be cycle-consistent [22], a desideratum that has

<sup>1</sup>Code will be made publicly available.



Figure 1: Examples for retargeting results generated by InGAN for scales and change of aspect ratios ranging from  $\times 0.33$  to  $\times 1.75$  of the original size. Input is marked with a **red frame**. Note that the only data the network is trained on is the input image. Once trained on a single image in a fully-unsupervised manner, InGAN can retarget to any size in real-time. The reader is encouraged to observe these and more examples in the supplementary material, where smooth videos of the transformations can be found.

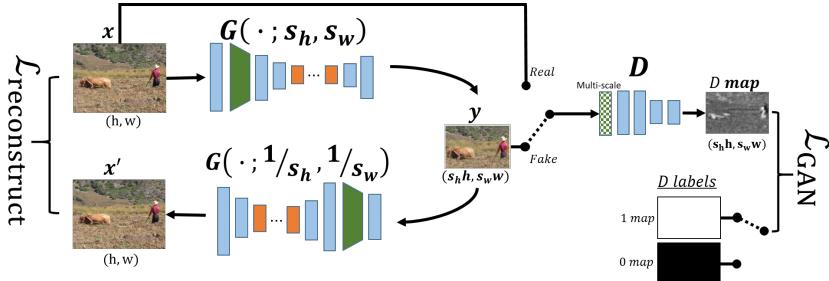


Figure 2: InGAN consists of a Generator  $G$  that retargets input  $x$  to output  $y$  with size  $(s_h \cdot h, s_w \cdot w)$  (top left). A multiscale discriminator  $D$  learns to discriminate the patch statistics of the fake retarget  $y$  from the true patch statistics of the input image (right). Additionally, we take advantage of  $G$ 's automorphism to reconstruct the input back from  $y$  using  $G$  (bottom left).

recently come into widespread use and often improves the results of distribution matching problems. Since our proposed InGAN is an automorphism, we use  $G$  itself to perform the decoding, that is  $\|G(G(x)) - x\|$  resulting with a novel Encoder-Encoder architecture.

Our contributions are therefore several-fold:

- To the best of our knowledge, this is the first work that performs natural image retargeting using a GAN.
- Introducing a GAN that is trained only internally i.e. trained on a single natural image.
- Defining distribution matching of patches as a criterion for retargeting, that is more general than bidirectional similarity [19].
- Exploiting the inherent symmetry of the challenge, and recognizing the wanted solution as an Automorphism gives rise to requiring an invertible Generator by employing an Encoder-Encoder network.
- Unlike many retargeting and texture-synthesis methods, we propose a network that once trained on a single image can retarget it to any size or aspect ratio in a single feed-forward pass.
- As oppose to existing GAN based methods, InGAN can capture multiple textures and elements at various scales, as is often the case in natural images.

## 2. Related Work

**Image retargeting:** Seam-carving [1] removes or adds “seams”, which are single-pixel-wide paths from one side of the image to another. Seams are selected such that when removed or added yield the minimal change to image gradients. Seam carving uses very local information, thus it is fast but unable to preserve global image structures.

Simakov *et al.* [19] introduced the aforementioned notion of bidirectional similarity for patches in several scales of the image. Looking at image patches at various scales allows for their method to outperform seam carving on nat-

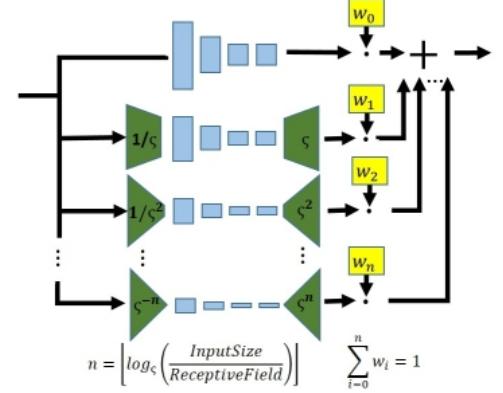


Figure 3: Adaptive Multi-Scale Discriminator.

ural images containing challenging textures and structures. However, their method requires to minimize the bidirectional similarity from scratch for each desired output size and aspect ratio.

Recently, retargeting using deep learning was introduced by [2]. Their method is based on a learned context-aware shift map which is a pixel-wise mapping from source to target grid. The learned shift map preserves salient objects while background regions are seamlessly discarded. It mostly does smart resizing while preserving the main object centered and in its original size, and most likely struggle when there are several dominant objects present.

In contrast to these methods, our InGAN is trained *once* for each input image to match multiscale patch statistics resulting in both coherent and complete retargeted outputs.

**Texture synthesis:** Texture synthesis is the task of generating a texture image given a small sample of the texture. This can be viewed as retargeting the input sample into the target output image. Gatys *et al.* [5] and [4] used a fully convolutional net to synthesize textures. More closely related methods to InGAN are spatial GAN [12] and non-stationary texture synthesis [21]. These methods use a patch-based GAN in a fully convolutional manner to produce high quality single texture images. However, despite their high quality results on single texture synthesis, these methods perform poorly when retargeting natural images because multiple textures at various scales are encountered.

Our InGAN, built upon the same principles of fully convolutional patch based GAN, on the other hand, is able to handle the complexity of the patch distribution of natural images thanks to our multiscale discriminator, and the automorphism built into the generator.

**Image-to-Image Translation using GANs:** Image retargeting can be seen as a form of image-to-image translation, where an input view of the world is transformed into an output representation that preserves the semantics, or *content*,

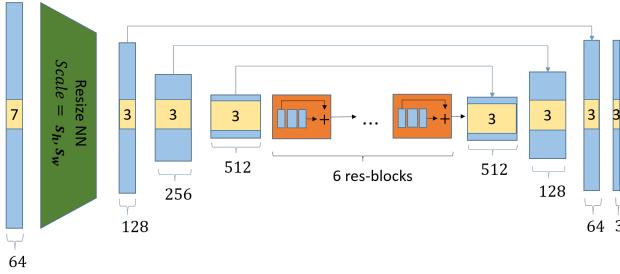


Figure 4: Generator architecture:  $G$  receives as input an image  $x$  and the retargeting scale  $s_h$  and  $s_w$ .

of the input but uses a different sort of “syntax” – the aspect ratio – to depict it. Recently, GANs have become a popular choice for solving translation problems like this [11]. However, whereas most previous work has translated from an input domain  $X$  to an output domain  $Y$ , our model is an automorphism that translates from  $X$  back to itself. Methods that translate between domains  $X$  and  $Y$  generally require supervision in the form of at least domain-level labels [22, 9], and often also require paired examples of the input-output relationship [11, 20]. Our method, on the other hand, requires no labels whatsoever – the only input to the training is a single photo.

**Internal Learning:** The fact that natural images have strong internal data repetition is well known. For example, small image patches (e.g.,  $5 \times 5$ ,  $7 \times 7$ ) repeat many times both within and across different image scales. This observation was empirically verified by [6, 23]. Recently, Shocher *et al.* [18] took advantage of this notion of internal statistics to train a fully convolutional network on a *single* image, and demonstrated zero-shot super-resolution utilizing Deep Internal Learning. Our framework uses internal GAN to model this multiscale patch repetitions within a single image, and utilizes this to match the patch distribution of the input and retargeted images.

### 3. Method

#### 3.1. Overview

Our InGAN uses a generator,  $G$ , a discriminator,  $D$ , and re-uses  $G$  for decoding/reconstructing, as depicted in Fig. 2. Given an input image  $x$  of size  $(h, w)$ ,  $G$  generates a retargeted image  $y$ , of size  $(s_h \cdot h, s_w \cdot w)$ , that maintains two requirements: The first, **matching distributions**: The distribution of patches, across scales, in the retargeted image, must match that distribution in the original input image. This requirement is a generalization of both the *Coherence* and *Completeness* requirements of [19]. The second requirement of  $G$  is **localization**: The elements’ locations in the generated image should generally match their relative locations in the original input image.

The discriminator  $D$  and  $\mathcal{L}_{\text{GAN}}$  encourage matching the patch distribution of  $y = G(x; s_h, s_w)$  to that of  $x$ .  $D$  is fully convolutional: each output pixel of  $D_{\text{map}}$  depends only on the receptive field of  $D$  [3], thus it has all the patches of the original input  $x$  to train on. We use a *multiscale*  $D$ , enforcing patch distribution matching at each scale separately.

In practice using only  $\mathcal{L}_{\text{GAN}}$  may result in mode collapse, i.e. the retargeted image consists of only a subset of patches of the original image (it is *coherent*) but many patches are missing (it is not *complete*). To ameliorate this mode collapse we take advantage of the automorphism of  $G$  and re-use  $G$  to reconstruct  $x$  back from the retargeted image  $y$ . The  $\ell_1$  reconstruction loss  $\mathcal{L}_{\text{reconstruct}} = \|G(y; 1/s_h, 1/s_w) - x\|_1$  encourages  $G$  to avoid mode collapse and maintain *completeness*. The overall loss function of InGAN is

$$\mathcal{L}_{\text{InGAN}} = \mathcal{L}_{\text{GAN}} + \lambda \cdot \mathcal{L}_{\text{reconstruct}}$$

The localization requirement is enforced by two elements; The first one is the fact that  $G$  is fully-convolutional. That is, the decoding is done in a position-preserving fashion. The second element is the simplicity of the locally matching solution. Since this is an image-to-image transformation rather than a classic GAN where the input is random noise, the solution of locally mapping elements from input to output is easy and natural for the generator to learn.

#### 3.2. Output-size dependant Generator

Fig. 4 shows the architecture of the generator  $G$ . The retarget output size  $(s_h \cdot h, s_w \cdot w)$  is treated as an additional input that is fed to  $G$  for every forward pass. A *parameter-free* resizing layer (green layer in Fig. 4) scales the feature map by  $(s_h, s_w)$  making  $G$  output  $y$  of the desired retarget size. Making the resize layer parameter-free allows training  $G$  once to retarget  $x$  to any size and aspect ratio at test time.

The generator is fully-convolutional with an hourglass architecture and skip connections (U-net [17] architecture). The bottleneck consists of residual-blocks [8]. Downscaling is done by max pooling. Upscaling is done by nearest-neighbor resizing followed by a convolutional layer [16].

#### 3.3. Multi-scale Patch Discriminator

We use a fully-convolutional patch discriminator  $D$  (Fig. 3), as introduced in [11]. The labels for the discriminator are maps of size  $(s_h \cdot h, s_w \cdot w)$ . This enforces grading each patch of the size of the receptive field for how well it matches the patch distribution, rather than grading the entire retargeted image.

InGAN uses a multi-scale  $D$  (similar to [20]). This feature is significant: A single scale discriminator can only capture patch statistics of specific size. Using multiscale  $D$  matches the patch distribution over a range of patch sizes

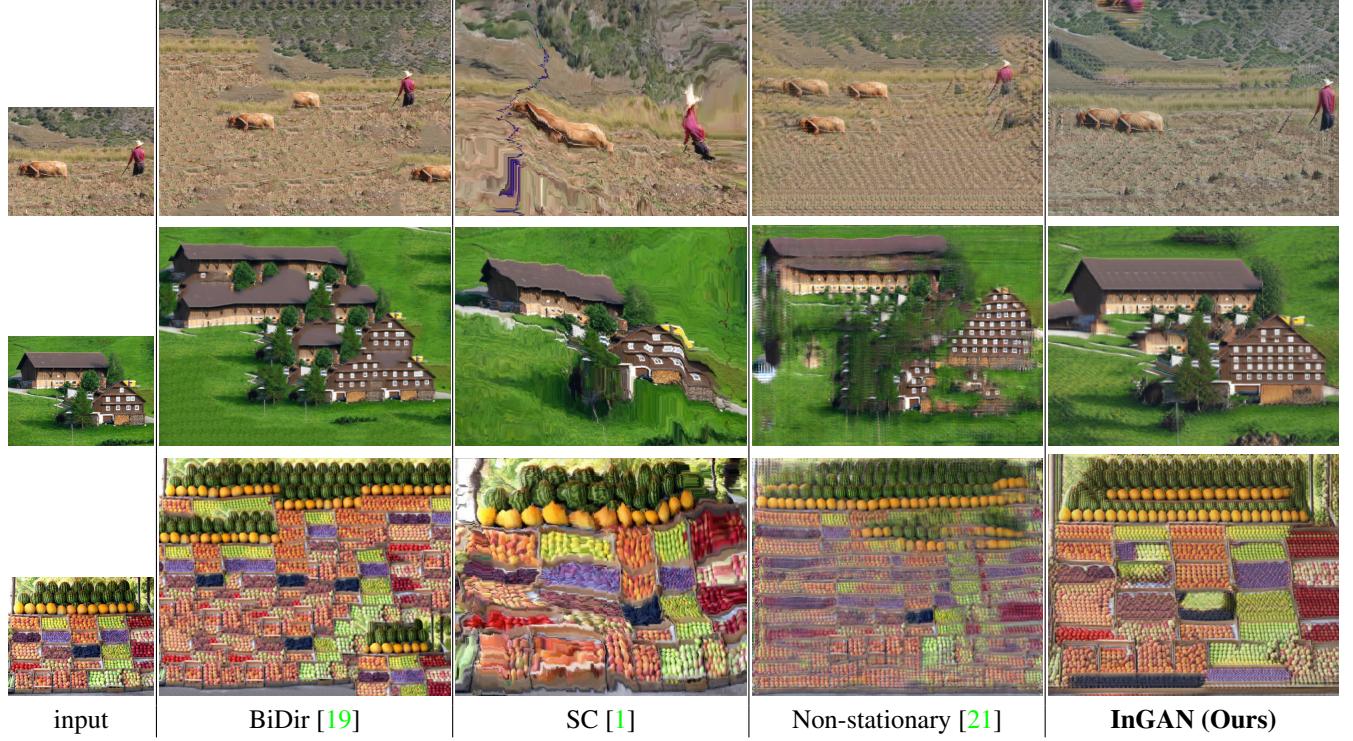


Figure 5: Comparison to other methods enlarging an image by factor of  $\times 2$  in both dimensions.

capturing both fine-grained details as well as coarse structures in the image. At each scale, the discriminator is rather simple: it consists of just four conv-layers with the first one strided. Weights are *not shared* between different scale discriminators. The downsampling factor from one scale to the next is set to  $\varsigma = \sqrt{2}$ .

The multiscale  $D$  outputs  $n$  discrimination maps that are summed via global weighted mean pooling to yield  $D$ 's output. These weights are updated during the optimization process in a coarse-to-fine manner. Initially, the weights are such that most the contribution to  $\mathcal{L}_{\text{GAN}}$  is from the coarsest scale. As the training progresses, the contribution gradually shifts to the finer scales.

### 3.4. Generator Invertibility

training  $D$  with  $\mathcal{L}_{\text{GAN}}$  often leads to mode collapse where retargeted  $y$  are *coherent* – the multiscale patches of  $y$  are drawn from the input image's distribution – but not *complete* – i.e. important visual information is missing from the retargeted  $y$ . To that end InGAN reconstructs the input image  $x$  from the retargeted image  $y$  ensuring no visual information was lost in  $y$ . Taking advantage of  $G$ 's automorphism allows us to re-use  $G$  to reconstruct  $x$  back from  $y$  without training any additional decoder, yielding an “Encoder-Decoder” architecture. This is advantageous: knowing the exact output to be reconstructed en-

ables  $\mathcal{L}_{\text{reconstruct}}$  to be a regular  $\ell_1$  loss, which provides a strong signal that influences the Generator's behavior at all scales and also acts as a stabilizer for the entire system.

## 4. Implementation Details

We use the ADAM optimizer [13] and a linearly decaying learning rate. We train over crops, ranging from  $192 \times 192$  to  $256 \times 256$  px, with a batch-size of 1. The default weighting of  $\mathcal{L}_{\text{reconstruct}}$  loss is  $\lambda = 0.1$ . At each iteration a non-uniform retarget scale  $(s_h, s_w)$  is randomly sampled, resulting in different output size and aspect ratio. We employ a form of curriculum-learning so that  $s_h$  and  $s_w$  are initially very close to 1. As the training progresses the allowed range of scales gradually grows through the curriculum period (10k iterations) until it finally covers the entire desired range.

We employ several mechanisms for encouraging stability; spectral normalization [15] is used in the discriminator and the generator for all layers except the last one. Batch normalization [10] is used in most conv-blocks. For  $\mathcal{L}_{\text{GAN}}$  we use the LSGAN [14] loss rather than cross-entropy. We also encountered a degenerate case where  $D$  was able to discriminate real patches from generated ones due to the fact that all values of the real patches were quantized to values  $n/255$ . To avoid this we add uniform noise in the range of  $[0, 1/255]$  to the real examples before feeding them to the

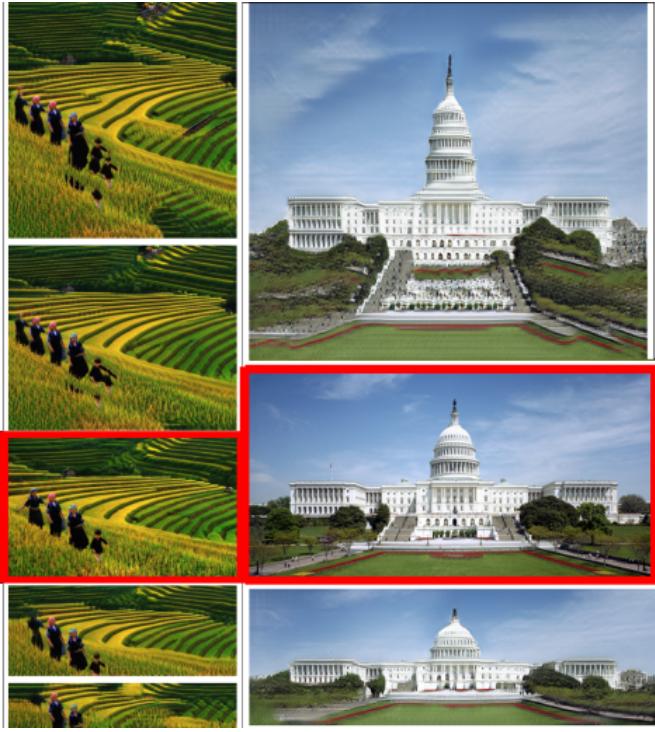


Figure 6: Examples of retargeting images. Input is marked with a red frame.

discriminator.

InGAN requires around 20k-50k iterations of gradient descent in order to obtain appealing results. Training takes 1.5-4 Hrs on a single V-100 GPU, regardless of the size of the input image. Once trained, an InGAN can retarget to VGA size ( $640 \times 480$ px) in about 40 ms (equivalent to 25 fps).

## 5. Experimental Results

We used InGAN to retarget a variety of images to various target scales and aspect ratios. Figures 1, 5, 6, 7 and 8 show some qualitative results. We encourage the reader to view additional results in the supplementary material.

### 5.1. Analysis of results

**Capturing multiscale patch distribution:** Unlike single-texture images, a natural image exhibits both fine grained details as well as coarse level structures. For example, look at the corn image in Fig. 1: small patches capture the minute details of the corn seed while coarse patches capture the structure of an entire corn-cob. InGAN is able to capture this multiscale patch distribution: when retargeting the corn image into a thinner/wider output entire corn-cobs are removed or added. Also when changing the image height, corn seeds are added or removed from each corn-cob. How-



Figure 7: Downsizing of images: We compared InGAN (our) to bidirectional similarity (BiDir) [19] and seam carving (SC) [1]. Retargeting image to  $\times 1/3$  of its original height and width (top), and to fit  $\times 1.35$  of the original height but only  $\times 1/3$  of the width (bottom).

ever, at no point neither the corn cobs nor the corn seeds are distorted by InGAN despite the radical change in aspect ratio between input and output. This multiscale patch distribution is a fundamental characteristic of natural images. InGAN's ability to capture it well can be observed also in other examples. For instance look at the fruit stand of Fig. 5 and the birds of Fig. 7. A good visualization of this capability of InGAN is shown in the video clips submitted in the supplementary material and we encourage the reader to view them.

**Element generalization:** Looking at the rice field image of Fig. 6 (left), it depicts three adults and one child. When retargeting the image to twice its height (top most result) InGAN adds one adult but also generates one young person, bigger than the smallest child but smaller than the adults. InGAN is able to generalize new elements from the learned distribution. Note that InGAN does not duplicate elements. Having more than one example, it can generalize and synthesize novel instances. Looking again at the corn example (Fig. 1): when new corn cobs are generated, no two cobs are alike, their coarse structure is synthesized from the distribu-

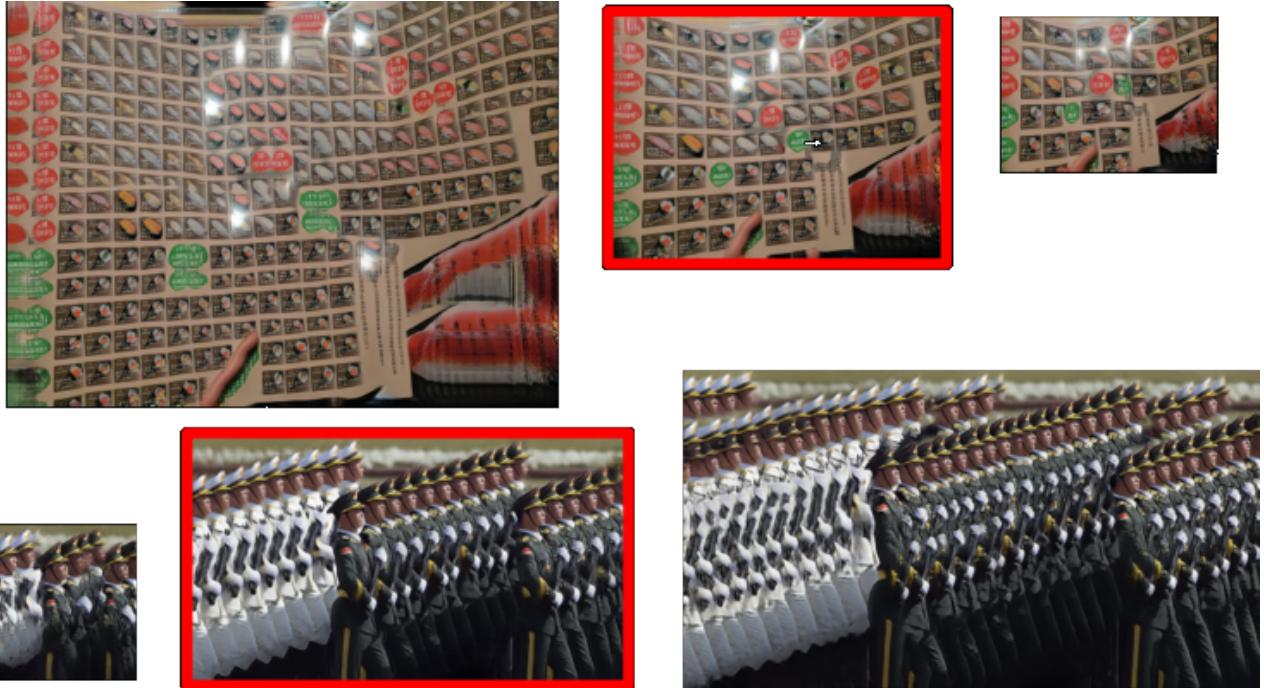


Figure 8: Examples for retargeting results emphasizing local distribution matching property of InGAN. Input is marked with a **red frame**. See sec. 5.1 for more details.

tion of coarse patches, while the corn seeds are generated from the distribution of small patches. Nonetheless it is important to stress that InGAN has no semantic information about “objects” or “scenes”, it only models the multiscale patch distribution of the input image.

**Local distribution matching:** Fig. 8 demonstrates the principle of local distribution matching: When retargeting menu items (top) or soldiers (bottom) InGAN matches the local distribution of elements in the images. New sushi dishes are synthesized at the relative location of sushi dishes, while new nigiri dishes are synthesized at their respective location. The same goes for the marching soldiers: InGAN synthesizes new soldiers with the appropriate uniform color relative to their local distribution. This local distribution matching property of InGAN gives rise to stability of retargeting results as the target size is being gradually changed. This stability is best observed in video clips attached in supplementary material.

## 5.2. Comparison to other methods

Fig. 5 visually compares InGAN to bidirectional similarity (BiDir) [19], seam carving (SC) [1] and non-stationary texture synthesis [21]<sup>2</sup> on the challenging task of retargeting images to twice their input size. BiDir exhibits very

<sup>2</sup>We were unable to compare to [2]: code or models were not made publicly available. Moreover, there is no project webpage or any other reliable source for the input images used in their study.

high quality locally, however, it struggles with maintaining intact global structures in the image, e.g., the roof of the farm house is bent, the crates of the fruit stand are shuffled. SC makes very local decision per added seam, thus when applied for such extreme retargeting it completely distorts the image. The method of [21], despite its similarity to InGAN, is only capable of  $\times 2$  enlarging of a *single* texture. When applied to natural images with many, multiscale textures it fails to produce visually pleasing results. In contrast to these methods, InGAN is capable of preserving both locality and multiscale textures and produce visually pleasing results even in these extreme retargeting settings.

Fig. 7 shows further comparisons to BiDir and SC on the task of retargeting to smaller size than input size (note that we cannot compare to [21] in these settings as it is restricted to  $\times 2$  retargeting). BiDir consistently produces very pleasing results locally, but struggles with very global structures: it crops the tips of the carrots and completely remove the dominant bird. SC distorts the output images in these retargeting scenarios. Again, InGAN is able to produce visually pleasing results by showing an entire carrot as well as preserving all three birds.

Additional comparison results as well as video clips showing comparisons of gradual retargeting of images can be found in the supplementary material.

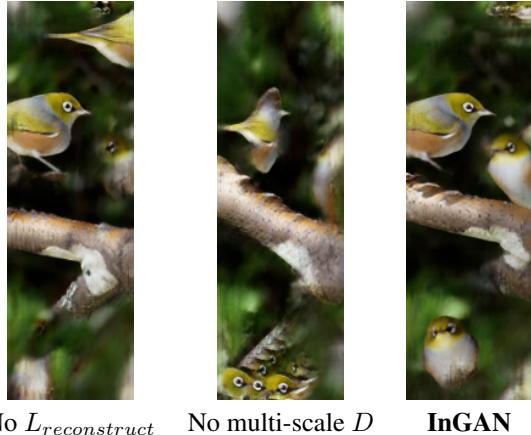


Figure 9: Ablation study example. Input can be found in fig 7. See sec. 5.3 for more details.



Figure 10: Failure examples. Please see sec. 5.4 for detailed explanation

### 5.3. Ablation Study

Fig. 9 shows an example of ablation study we conducted to verify the importance of (1) the “encoder-encoder” architecture and the  $\ell_1$  reconstruction loss, and (2) the importance of multiscales in the discriminator  $D$ . We used the birds input image (Fig. 7). Training InGAN without  $\mathcal{L}_{\text{reconstruct}}$  (left most result) shows unstructured output: two birds are completely missing and the dominant bird is split into two. Using single scale  $D$  (middle result) makes  $G$  generate a result that is locally coherent, but lacks large scale structures. The birds were completely destroyed. In contrast, full InGAN (right most result) with  $\mathcal{L}_{\text{reconstruct}}$  and multiscale  $D$  maintains both fine details and coarse structures in the retargeted output.

### 5.4. Limitations and Failure Cases

During our experiments we noticed two types of failure cases. Fig. 10 shows an example of each type:

**Local Orientation:** InGAN requires all patches of the output image to match the distribution of patches in the input. Transformations/deformations over these patches can

be generalized, but only if sufficient diversity in the input patches exist. When the input image contains certain elements oriented only in one direction, e.g., the contrails of the airplanes, the learned distribution cannot generalized to more possible orientations leading to severe artifacts in the retargeted image.

**Object partition ambiguity:** InGAN is unsupervisedly trained on a single image thus lacks semantic understanding. This fact prevents modeling of what is a full object that can only exist as a complete unit and what can be partitioned into standalone sub-objects. This phenomenon is visible in the soldiers example: when retargeting to larger image height, InGAN stops treating the soldiers as a single “unit” and duplicates their legs only.

### 5.5. Future Directions

Capturing internal statistics of a natural image has broader potential beyond just natural image retargeting. Fig. 11 shows an initial example of applying InGAN to image inpainting.

### 6. Take Home Message

Capturing the internal *multiscale* distribution of patches in an image is crucial to make the step from single-texture synthesis to natural image retargeting. InGAN tackles this challenge by using a multiscale discriminator trained as part of a fully convolutional GAN on a *single* image. Once trained, the generator of InGAN can retarget image  $x$  to any size  $(s_h \cdot h, s_w \cdot w)$  in a single feed forward pass. Training InGAN to capture the internal multiscale statistics of an image has a nice by product: the generator is invertible, it can retarget an image to new size/aspect ratio but also transform the retargeted image back. This “encoder-encoder” behavior helps InGAN training escape mode collapse and improves stability.



Figure 11: Inpainting using InGAN: Training InGAN on the full image allow to in-paint the missing part using the learned patch statistics of the entire image.

## References

- [1] S. Avidan and A. Shamir. Seam carving for content-aware image resizing. *ACM Trans. Graph.*, 26(3), July 2007. [3](#), [5](#), [6](#), [7](#)
- [2] D. Cho, J. Park, T.-H. Oh, Y.-W. Tai, and I. S. Kweon. Weakly and self-supervised learning for content-aware deep image retargeting. In *The IEEE International Conference on Computer Vision (ICCV)*, 2017. [3](#), [7](#)
- [3] V. Dumoulin and F. Visin. A guide to convolution arithmetic for deep learning. *arXiv preprint arXiv:1603.07285*, 2016. [4](#)
- [4] L. A. Gatys, A. S. Ecker, and M. Bethge. A neural algorithm of artistic style, 2015. [3](#)
- [5] L. A. Gatys, A. S. Ecker, and M. Bethge. Texture synthesis using convolutional neural networks, 2015. [3](#)
- [6] D. Glasner, S. Bagon, and M. Irani. Super-resolution from a single image. In *International Conference on Computer Vision (ICCV)*, 2009. [4](#)
- [7] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014. [1](#)
- [8] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015. [4](#)
- [9] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz. Multimodal unsupervised image-to-image translation. In *ECCV*, 2018. [4](#)
- [10] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ICML’15, pages 448–456. JMLR.org, 2015. [5](#)
- [11] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. [4](#)
- [12] N. Jetchev, U. Bergmann, and R. Vollgraf. Texture synthesis with spatial generative adversarial networks, 2016. [3](#)
- [13] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. [5](#)
- [14] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, and Z. Wang. Least squares generative adversarial networks. In *Computer Vision (ICCV), IEEE International Conference on*, 2017. [5](#)
- [15] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*, 2018. [5](#)
- [16] A. Odena, V. Dumoulin, and C. Olah. Deconvolution and checkerboard artifacts. *Distill*, 2016. [4](#)
- [17] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. *Medical Image Computing and Computer-Assisted Intervention MICCAI 2015*, page 234241, 2015. [4](#)
- [18] A. Shocher, N. Cohen, and M. Irani. Zero-shot super-resolution using deep internal learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. [4](#)
- [19] D. Simakov, Y. Caspi, E. Shechtman, and M. Irani. Summarizing visual data using bidirectional similarity. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008. [1](#), [3](#), [4](#), [5](#), [6](#), [7](#)
- [20] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. [4](#)
- [21] Y. Zhou, Z. Zhu, X. Bai, D. Lischinski, D. Cohen-Or, and H. Huang. Non-stationary texture synthesis by adversarial expansion. *ACM Trans. Graph.*, 37(4):49:1–49:13, July 2018. [3](#), [5](#), [7](#)
- [22] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Computer Vision (ICCV), IEEE International Conference on*, 2017. [1](#), [4](#)
- [23] M. Zontak and M. Irani. Internal statistics of a single natural image. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011. [1](#), [4](#)