Wearable ImageNet: Synthesizing Tileable Textures via Dataset Distillation

George Cazenavette[1] Tongzhou Wang[2] Antonio Torralba[2] Alexei A. Efros[3] Jun-Yan Zhu[1]

[1]Carnegie Mellon University [2]Massachusetts Institute of Technology [3]UC Berkeley

georgecazenavette.github.io/mtt-distillation

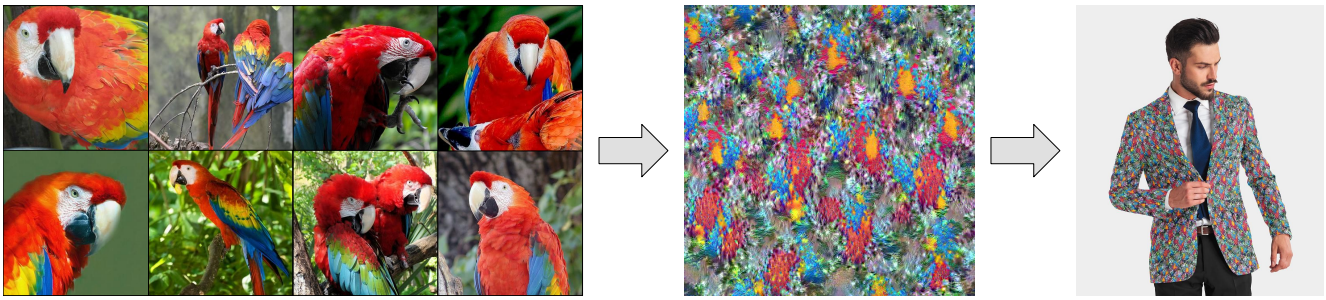


Figure 1. Our new method builds upon recent dataset distillation work [2] and lets us distill classes into *tilable textures*. These textures can then be used for downstream tasks, such as pattern swatches for clothing. (Visualizations made with FAB3D: https://fabric.tri3d.in/)

Abstract

Recent methods for Dataset Distillation are able to take in a large set of images of a specific class (e.g., from ImageNet) and synthesize a single image, such that a classifier trained on that image could perform similarly to one trained on the original dataset. It was noticed that the resulting "distilled images" are often quite visually pleasing. In this paper, we describe a simple method for generating tileable distilled textures by sampling random crops from a toroidal canvas of synthetic pixels while enforcing that all such crops serve as effective distilled training data. Such distilled textures not only summarize a given image category in a visually interesting way, but also allow for generation of infinite texture patterns suitable for printing on fabric, clothing, etc. This paper might be just the first step in making the ImageNet dataset into a fashion statement.

1. Introduction

The task of dataset distillation [2, 2, 8, 9, 10, 11, 12, 13, 14] involves creating a small synthetic dataset (as little as one image per class) such that a model trained on this synthetic dataset will have similar test-time performance as a model trained on the full original training set. This problem is particularly challenging, as a good solution must strike a delicate balance between compressing the visual information into just a few images, while still preserving the learnable discriminative features of each class. Unexpectedly, the results of the most recent method [2] have shown an additional property of being visually intriguing (see Figure 4B). In this short paper, we further explore dataset distillation as a tool for synthesizing visually interesting textures.

Until now, all existing methods of dataset distillation have focused on the synthesis of standard-sized distilled image(s) corresponding to each class. Here, we propose an alternative method that yields a more artistic yet still functional result. We consider a "canvas" of pixels for each class, which is twice as large as the input training images. At distillation time, we apply circular padding and take random crops from this canvas. By optimizing all such random crops to serve as effective training data, we see the emergence of category-based textures that preserve continuity across their borders. While being aesthetically pleasing in their own right, the resulting images can also be applied to practical tasks, such as pattern swatches used to create clothing or wallpaper. We call our method Tileable Texture Distillation.

Our method can be thought of as a type of texture synthesis, but while classic patch-based synthesis methods, e.g. [5], create a texture from a single source image, our method produces a result that captures the "essence" of an entire dataset class. Our way of synthesizing tileable textures is inspired by Wang tiles work of Cohen et al. [3] (and their algorithm can easily be used to extend our work to non-periodic tiled textures). Our other source of inspiration is the classic Image Epitomes work [7], but while epitomes capture the statistics necessary for *reconstruction* of an image, our method instead focuses on data needed for discrimination.

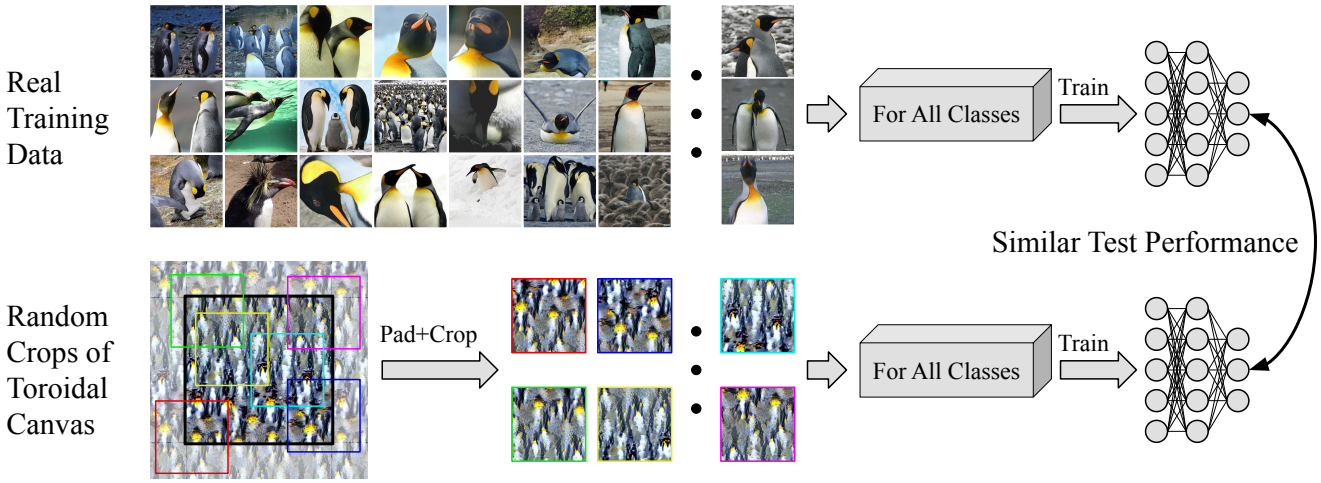


Figure 2. Our method takes random crops of a padded distilled *canvas* instead of discrete images. Note: Our method distills all classes simultaneously, although only one (penguin) is shown in detail in this figure for simplicity.

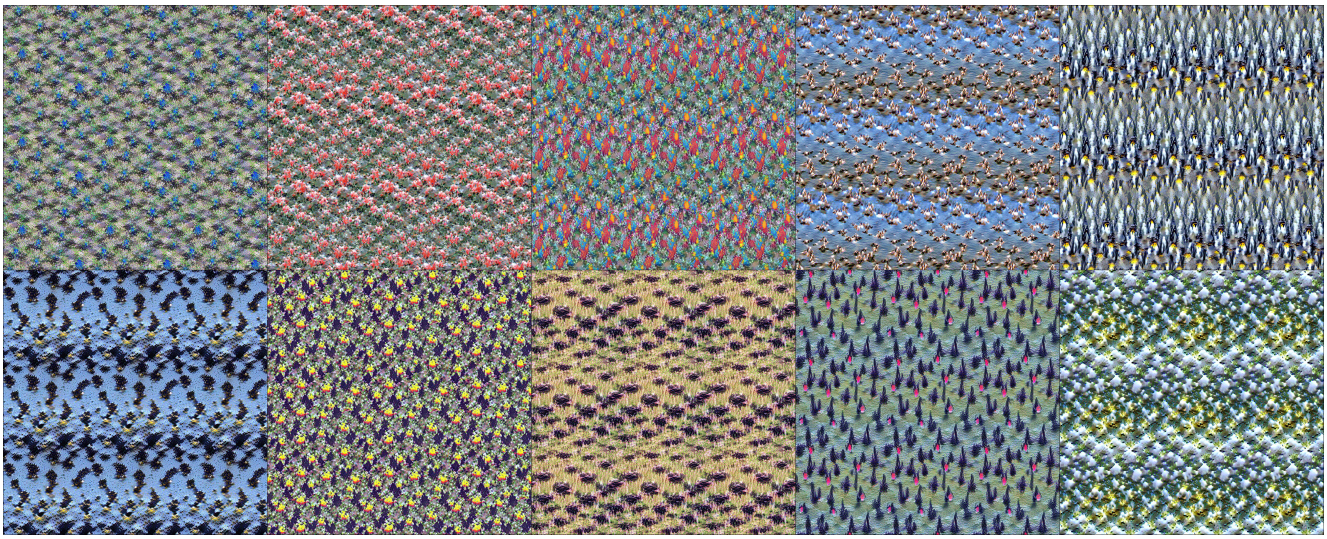


Figure 3. ImageSquawk Distilled Textures Tiled 3x3

2. Preliminaries

Here, we briefly review our underlying distillation method: Dataset Distillation by Matching Training Trajectories (`MTT`) [2]. Similar to previous methods, `MTT` seeks to distill synthetic data that can still train a well-performing model.

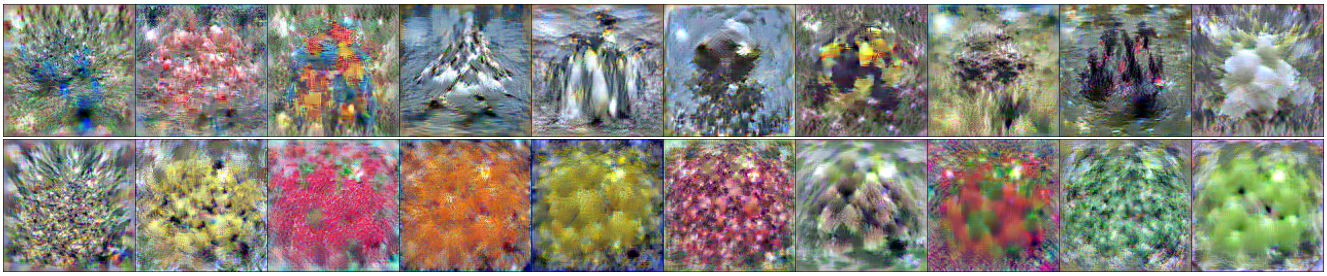
While other methods attempt to optimize the synthetic data with respect to either a single gradient step [12, 14] or the entire training process [8, 9, 11], `MTT` considers many-step trajectories starting from varying points along pre-computed *expert trajectories* (those obtained by training on the full, real training set). On a high-level, `MTT` encourages models trained on synthetic data to mimic the behavior of the expert trajectory.

Specifically, `MTT` initialize initializes a student network at some point along an expert trajectory. The student network is then trained for many ($N \approx 50$) iterations on the synthetic data, and the "distillation loss" is calculated as the relative error between the student network's current parameters and those of a future timestep of the expert trajectory. The motivation behind this method is an effort to *directly* optimize the synthetic data such that it induces a similar training trajectory as the real training set, resulting in a final model that lies at a similar point in parameter space. For further details on the distillation method from which ours was adapted, we refer the reader to [2].

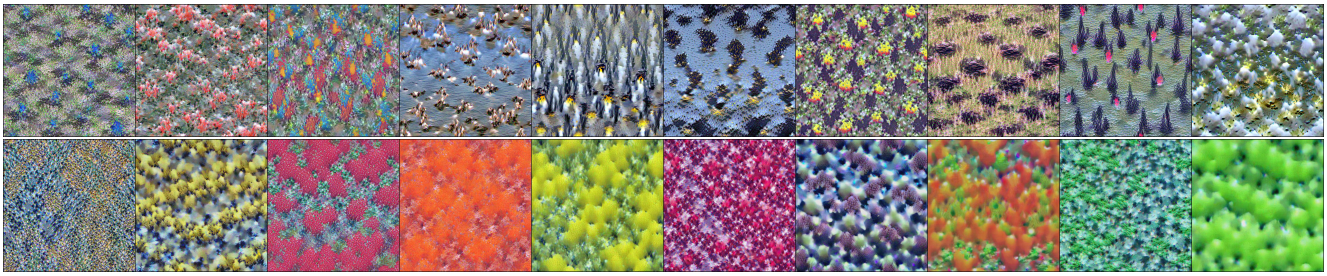
3. Tilable Texture Distillation

Our method, Tilable Texture Distillation, builds upon `MTT` and offers a new mode of distillation. Instead of distilling individual, disjoint training images, we instead optimize a toroidal canvas of pixels such that any random crop of the canvas is a good training sample. As illustrated in Figure 2, our optimization objective is to ensure that a model trained

(A) Real Training Data (showing 1 image per class)



(B) Cazenavette et al. [2]



(C) Tilable Texture Distillation (Ours)

Figure 4. Traditional distillation methods take the real training data (A) and create disjoint synthetic training images (B). Our new method builds upon recent dataset distillation work [2] and allows for the synthesis of class-based tileable textures (C).

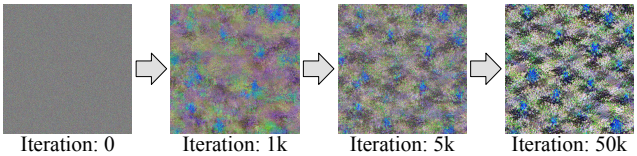


Figure 5. We initialize our synthetic canvases from Gaussian noise. Structure emerges after several hundred iterations, but it takes tens of thousands until convergence.

on random crops of our distilled canvases will perform similarly to a model trained on the real training set.

To achieve this effect, we first apply *circular padding* in both the x and y directions. By always considering the image to be circular padded, we effectively induce a toroidal topology on our synthetic canvas. Where `MTT` would use the disjoint synthetic samples to train the student trajectory, we instead train our student trajectory on *random crops* of our padded synthetic canvases, including such crops that would span the seams of the torus.

While the random cropping alone induces a texturized appearance in our synthetic canvas, the circular padding ensures that they will also be *tilable*. Our loss function naturally encourages each "image" fed through the student network to be continuous. By using circular padding, we enforce that the patches that span the seams of the torus (i.e. the edges of the image) to also be continuous. This has the corollary effect of making our images tileable, as seen in Figure 3. Note that both padding and cropping are differentiable operations.

Please refer to our Appendix (Alg. 1) for more details.

4. Experiments

Since we are focusing on visual (qualitative) results in this work instead of classification performance, we chose to distill higher-resolution imagery than in the `MTT` paper [2]. Specifically, for each class, we distilled a 512×512 toroidal canvas wherein each 256×256 patch is optimized to be a good training sample. For comparison, the largest images distilled in `MTT` were 128×128.

We stay with the precedent set by other dataset distillation work and use ConvNet models [6]. Since our patches are 256×256, we use a depth-6 ConvNet for our distillation, following the pattern seen in `DM` [13] and `MTT` [2].

We distill the ImageSquawk subset of ImageNet [4] as

Figure 6. By distilling classes into tileable textures, we can apply our synthetic data to cases where such tileability is required, such as fabric pattern swatches. From left to right, we see our flamingo, eagle, macaw, and penguin textures applied to clothing using FAB3D [1]

introduced in [2]. We chose this dataset with the thought that the various colors and patterns of the birds' plumage would make for visually appealing textures. Specifically, ImageSquawk is composed of the peacock, flamingo, macaw, pelican, penguin, eagle, toucan, ostrich, black swan, and cockatoo classes. We also distill ImageFruit, which contains pineapple, banana, strawberry, orange, lemon, pomegranate, fig, bell pepper, cucumber, and granny smith.

4.1. Results

We present the result of our main experiment in Figure 1. Our method (b) can synthesize texture images with no centering bias while still resembling the distinct features of their respective classes (a), which makes them potentially useful in designing new clothes (c).

In Figure 3, we see that our distilled textures seamlessly tile together into a larger, continuous image. It is possible to create larger tiles without increasing memory cost simply by distilling a larger canvas while keeping the underlying patch size the same. However, this method would significantly increase training time since each patch would be sampled less frequently, requiring more total iterations for convergence.

As far as training (distillation) dynamics, we see coarse structure emerge relatively early in the optimization, within the first several hundred iterations. However, it takes many more iterations, on the order of tens of thousands, to synthesize a more defined structure in the texture and remove the high-frequency noise components, as seen in Figure 5.

Once we have our converged toroidal textures, we can apply them to domain-relevant tasks, such as pattern swatches for fabrics. Using FAB3D [1], we visualize our textures as applied to clothing. FAB3D has knowledge of the clothing article's underlying topology, so it makes full use of our textures' tileability, as seen in Figure 6.

5. Discussion

In this work, we introduced an extension to a recent dataset distillation algorithm that allows us to distill tileable, class-based textures. While all distillation methods to date have solely focused on classification performance [8, 9, 10, 11, 12, 13, 14], our new method is the first to change the overall objective to synthesize something other than disjoint training samples.

Tilable Texture Distillation takes random crops across an induced toroidal canvas and enforces them to all be good training examples for discriminating the respective class. It gives us a unique way of synthesizing tileable class-based textures for down-stream use. We also include code allowing users to easily distill textures for their own classes of choice.

By presenting a dataset distillation method with an end goal other than training discriminative models, we hope to encourage the creative use of dataset distillation methods for other novel tasks.

Limitations. As we continue to distill higher-resolution images, the process becomes more computationally costly, both spatially and temporally. For `MTT` specifically, training larger models to obtain the expert trajectories (i.e., depth-6 ConvNet versus depth-3 ConvNet) takes much longer, and the checkpoints take up significantly more space when stored on disk. Furthermore, distilling quadratically more pixels (i.e., doubling the resolution) while keeping the patch size the same requires *many* more iterations before convergence, especially when performing our new random crop method.

Acknowledgements. We are grateful to the artist Danielle Baskin for her suggestion to that we apply our distillation method to creating fabric patterns. This work is supported, in part, by the NSF Graduate Research Fellowship under Grant No. DGE1745016 and grants from J.P. Morgan Chase, IBM, and SAP.

References

[1] Tri3d fabric visualiser. https://fabric.tri3d.in/. 4

[2] George Cazenavette, Tongzhou Wang, Alexei A. Efros, Antonio Torralba, and Jun-Yan Zhu. Dataset distillation by matching training trajectories. In *CVPR*, 2022. 1, 2, 3, 4

[3] Michael F Cohen, Jonathan Shade, Stefan Hiller, and Oliver Deussen. Wang tiles for image and texture generation. *ACM TOG*, 22(3):287–294, 2003. 1

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 3

[5] Alexei A Efros and William T Freeman. Image quilting for texture synthesis and transfer. In *SIGGRAPH*, pages 341–346, 2001. 1

[6] Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting. In *CVPR*, 2018. 3

[7] Nebojsa Jojic, Brendan J Frey, and Anitha Kannan. Epitomic analysis of appearance and shape. In *ICCV*, volume 2, pages 34–34. IEEE Computer Society, 2003. 1

[8] Timothy Nguyen, Zhourong Chen, and Jaehoon Lee. Dataset meta-learning from kernel ridge-regression. In *ICLR*, 2020. 1, 2, 4

[9] Timothy Nguyen, Roman Novak, Lechao Xiao, and Jaehoon Lee. Dataset distillation with infinitely wide convolutional networks. *NeurIPS*, 2021. 1, 2, 4

[10] Kai Wang, Bo Zhao, Xiangyu Peng, Zheng Zhu, Shuo Yang, Shuo Wang, Guan Huang, Hakan Bilen, Xinchao Wang, and Yang You. Cafe: Learning to condense dataset by aligning features. In *CVPR*, 2022. 1, 4

[11] Tongzhou Wang, Jun-Yan Zhu, Antonio Torralba, and Alexei A Efros. Dataset distillation. *arXiv preprint arXiv:1811.10959*, 2018. 1, 2, 4

[12] Bo Zhao and Hakan Bilen. Dataset condensation with differentiable siamese augmentation. In *ICML*. PMLR, 2021. 1, 2, 4

[13] Bo Zhao and Hakan Bilen. Dataset condensation with distribution matching. *arXiv preprint arXiv:2110.04181*, 2021. 1, 3, 4

[14] Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. Dataset condensation with gradient matching. In *ICLR*, 2020. 1, 2, 4