Visual Atoms: Pre-training Vision Transformers with Sinusoidal Waves –Supplementary Material–

Sora Takashima[1,2], Ryo Hayamizu[1], Nakamasa Inoue[1,2], Hirokatsu Kataoka[1], Rio Yokota[1,2]

[1]National Institute of Advanced Industrial Science and Technology (AIST)
[2]Tokyo Institute of Technology

1. Details of VisualAtom

Example images of VisualAtom (Figure 1). We list example VisualAtom images of randomly selected 50 classes with one instance per class in Figure 1a. These images are selected from VisualAtom generated with our baseline parameters setting. Also, we list example images of randomly selected 50 instances in one randomly selected class in Figure 1b. In Figure 2, with the similar format, we also list example images of RCDB [3], one of the previous FDSL datasets, for comparison. As you can see, VisualAtom has more variation in contour outlines between categories than RCDB.

Details on generating VisualAtom. We include separate scripts named 'visual_atomic_renderer' to generate VisualAtom in the supplementary material. Please see 'visual_atomic_renderer/README.md' for details on how to execute these scripts. We confirmed generating VisualAtom-1k took 40 minutes using parallel execution with 40 threads on 20 CPU cores.

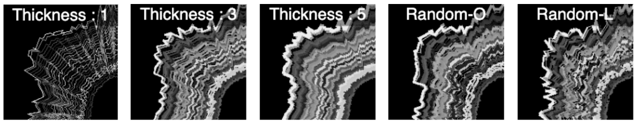
2. Additional experimental results

Here, we describe our findings and insights into VisualAtom parameters that are relatively unimportant in contours. The findings with these parameters could not be included in the main paper due to space limitations. As in the main paper, we pre-train ViT-Tiny [2] on VisualAtom-1k when varying parameters, and compare of fine-tuning accuracy on three datasets: CIFAR10 (C10) [4], CIFAR100 (C100) [4] and ImageNet-100 (IN100)[1]

Orbit-Thickness Parameter (Table 1). In Table 1, we show the effects of the line thickness parameter on rendering. Here, we used the following configurations: fixed at 1, 3, or 5 pixel, and Random (orbit) or Random (line).

Table 1. Fine-tuning accuracy when varying line thickness parameter l. Random-O and -L uses randomly sampled line thickness l at each orbit and line, respectively, where l is uniformly sampled from $\{1, 2, 3, 4, 5\}$ pixels.

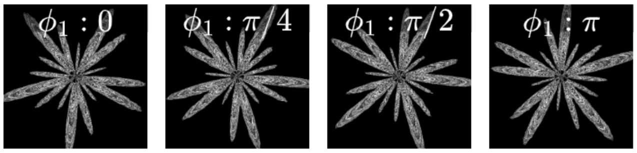
Value of l	C10	C100	IN100
1 pixel	97.6	84.9	**90.3**
3 pixel	97.6	**85.7**	90.1
5 pixel	97.6	84.5	89.9
Random-O	**97.7**	85.1	90.0
Random-L	97.6	85.4	89.8



The first three configurations fix the line-thickness parameter at $l = 1, 3, and\ 5$ pixel, respectively. The 4th/5th configurations randomly select l from $\{1, 2, 3, 4, 5\}$ per orbit/line, respectively. The findings show that the three-pixel line thickness greatly improved the performance on C100, but that it was not always best. For example, for IN100, the one-pixel line thickness was best. For simplicity, we used the one-pixel configuration for the baseline.

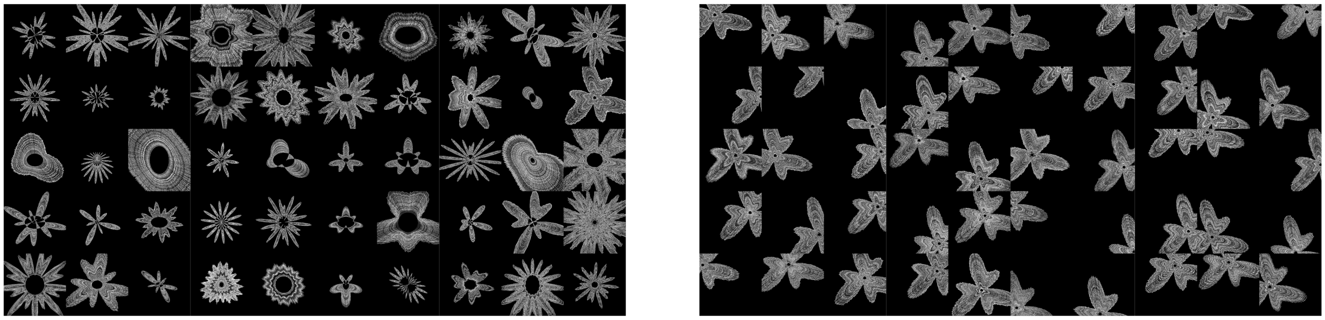
Table 2. Fine-tuning accuracy when using randomly phase difference parameters ϕ_1 and ϕ_2 at each categories, where ϕ_1 and ϕ_2 are uniformly sampled from the range of $[0, \pi]$.

Value of ϕ_1 and ϕ_2	C10	C100	IN100
0, 0	97.6	84.9	**90.3**
Random	97.5	84.8	90.1



Phase Difference Parameters (Table 2). In Eq. (5) of the main paper, introducing phase difference parameters $\phi_1, \phi_2 \in \mathbb{R}$ varies the phase of the two waves as follow:

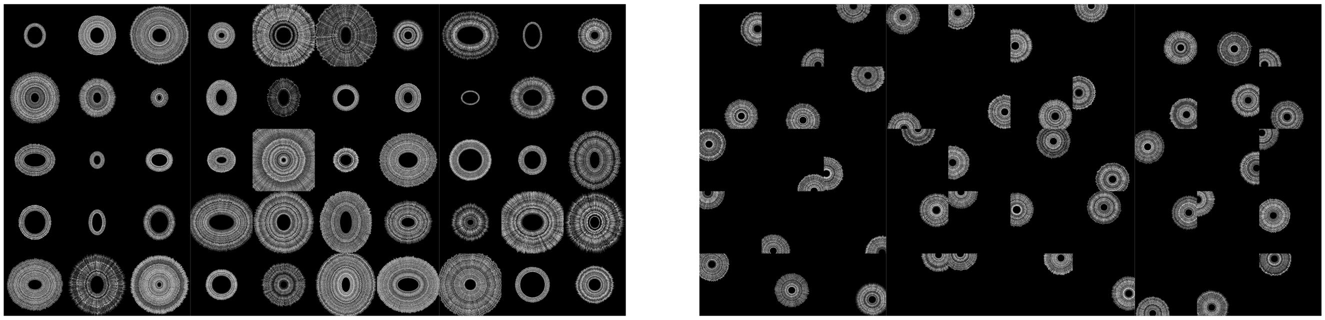
[1] This is a subset of ImageNet [1] with 100 object categories.



(a) VisualAtom Classes

(b) VisualAtom Instances

Figure 1. **Example images of VisualAtom.** (a) Example images of randomly selected 50 classes with one instance per class. (b) Example images of randomly selected 50 instances in one randomly selected class.



(a) RCDB Classes

(b) RCDB Instances

Figure 2. **Example images of RCDB [3].** (a) Example images of randomly selected 50 classes with one instance per class. (b) Example images of randomly selected 50 instances in one randomly selected class.

$$\Phi_k(\theta) = \lambda_1 \sin(n_1\theta + \phi_1) + \lambda_2 \sin(n_2\theta + \phi_2) + \eta\epsilon(\theta). \tag{5*}$$

We tried the phase difference parameters ϕ_1 and ϕ_2 defined for the two waves, randomly sampled from the range of $[0, \pi]$. Table 2 shows that VisualAtom has a robust pre-training effect on phase differences. For simplicity, we did not change the phase of two waves for the baseline.

3. Hyper-parameters in our experiments

For each experiments, hyper-parameters are based on the configuration used by Kataoka *et al.* [3]. More fundamentally, they are based on the paper proposing DeiT [6]. Table 3 shows hyper-parameters in our experiments.

We conducted our experiments using the training scripts used in previous works [3, 5] almost verbatim. The training scripts we used are available on the Github repository[2]. The scripts to generate the VisualAtom are published in the same Github repository. Also, for pre-training on the large dataset such as VisualAtom-21k, we used WebDataset[3] to accelerate IO processing. Note that we changed only the parameter of the Warmup interval to 5k steps from 5 epochs used in the previous work. This is to apply Warmup at a fixed iterations regardless of the size of dataset. It should be noted that the loss in pre-training was found to be sufficiently convergent with the number of epochs shown here.

References

[1] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 1

[2] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations*, 2021. 1

[3] Hirokatsu Kataoka, Ryo Hayamizu, Ryosuke Yamada, Kodai Nakashima, Sora Takashima, Xinyu Zhang, Edgar Josafat

[2] Our Github repository : https://github.com/masora1030/CVPR2023-FDSL-on-VisualAtom

[3] WebDataset library : https://github.com/webdataset/webdataset

Table 3. Hyper-parameters of pre-training and fine-tuning in our experiments. Basically, they are same as the configuration used by Kataoka *et al.* [3].

Training Step	Pre-training				Fine-tuning	
Model	ViT-T		ViT-B		ViT-T/B	
Dataset Category	1k	21k	1k	21k	1k	Others
Epochs	300	90	300	90	300	1000
Batch Size	1024	8192	1024	8192	1024	768
Optimizer	AdamW	AdamW	AdamW	AdamW	AdamW	SGD
LR	1.0e-3	8.0e-3	1.0e-3	1.0e-3	1.0e-3	1.0e-2
Weight Decay	0.05	0.05	0.05	0.05	0.05	1.0e-4
LR Scheduler	Cosine	Cosine	Cosine	Cosine	Cosine	Cosine
Warmup Steps	5k	5k	5k	5k	5 (epochs)	10 (epochs)
Resolution	224	224	224	224	224/384	224
Label Smoothing	0.1	0.1	0.1	0.1	0.1	0.1
Drop Path	0.1	0.1	0.1	0.1	0.1	0.1
Rand Augment	9/0.5	9/0.5	9/0.5	9/0.5	9/0.5	9/0.5
Mixup	0.8	0.8	0.8	0.8	0.8	0.8
Cutmix	1.0	1.0	1.0	1.0	1.0	1.0
Erasing	0.25	0.25	0.25	0.25	0.25	0.25

Martinez-Noriega, Nakamasa Inoue, and Rio Yokota. Replacing Labeled Real-Image Datasets With Auto-Generated Contours. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21232–21241, 2022. 1, 2, 3

[4] Alex Krizhevsky, Hinton, and Geoffrey. Learning Multiple Layers of Features from Tiny Images. 2009. 1

[5] Kodai Nakashima, Hirokatsu Kataoka, Asato Matsumoto, Kenji Iwata, Nakamasa Inoue, and Yutaka Satoh. Can Vision Transformers Learn without Natural Images? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 1990–1998, 2022. 2

[6] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herve Jegou. Training Data-efficient Image Transformers & Distillation through Attention. In *International Conference on Machine Learning*, volume 139, pages 10347–10357, July 2021. 2