# Formula-driven Supervised Learning with Recursive Tiling Patterns

Hirokatsu Kataoka, Asato Matsumoto, Ryosuke Yamada, Yutaka Satoh
National Institute of Advanced Industrial Science and Technology (AIST)
Tsukuba, Ibaraki, Japan
{hirokatsu.kataoka}@aist.go.jp

Eisuke Yamagata, Nakamasa Inoue
Tokyo Institute of Technology
Yokohama, Kanagawa, Japan

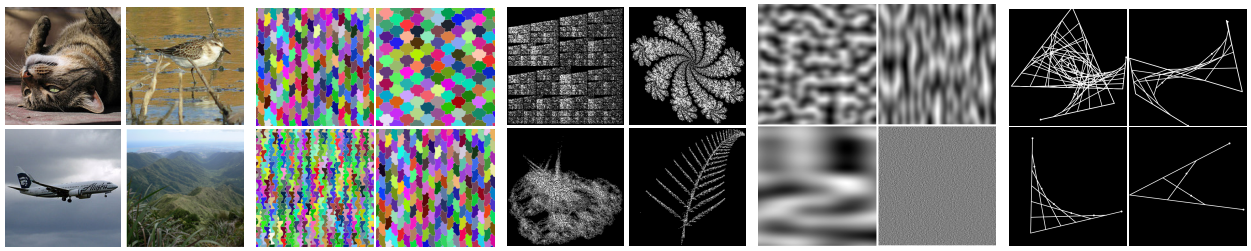
## Abstract

*Can convolutional neural networks pre-trained without natural images be used to assist natural image understanding? Formula-Driven Supervised Learning (FDSL) automatically generates image patterns and their category labels by assigning a well-organized formula. Due to the characteristics of not using natural images in pre-training phase, FDSL is expected to develop a trustworthy vision-based system in terms of human-annotation-free, fairer and more transparent datasets. In this paper, we propose TileDB which consists of recursive tiling patterns in the whole image and evaluates the family of FDSL such as the datasets consist of Perlin noise and Bezier curves. Experimental results show that our proposed TileDB pre-trained model performs much better than models trained from scratch, surpasses a similar self-supervised learning (SSL), and performs similarly to the models pre-trained with 100k-order natural image datasets such as ImageNet-100 and Places-30. By comparing to the FractalDB pre-trained model, the TileDB pre-trained model achieves better performances in a compact dataset (< 1,000 categories). Moreover, the image representation trained on TileDB can extract similar features to the ImageNet pre-trained model even though the training images are non-trivially different.*

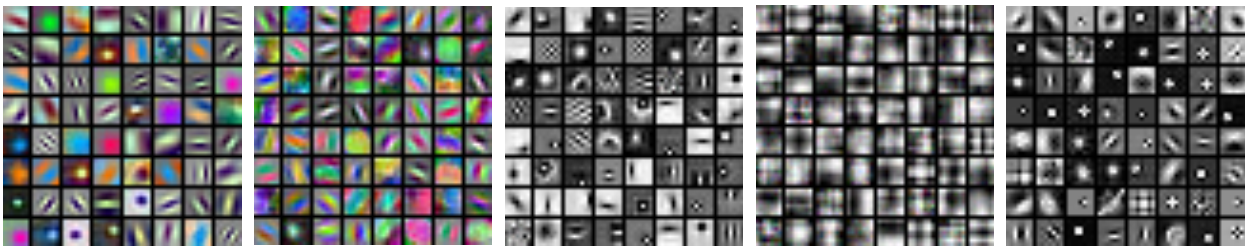
## 1. Introduction

The potential of image recognition has been greatly expanded with the introduction of sophisticated pre-training image representation. Undoubtedly, in the field of computer vision, an image representation with, for example, the ImageNet/Places pre-trained convolutional neural network (CNN), has become the most important breakthrough [6, 38]. We have gained much from the ImageNet project for constructing large scale image datasets. However, recent discussions have revealed that these datasets have problems, namely privacy-violating and ethics-related labels [36]. In other words, the ImageNet dataset is limited only to non-commercial usage because the images involved in ImageNet cannot mitigate problems related to copyrights. We believe that this aspect of pre-trained models restricts the prospects of vision-based recognition.

The frontier in vision-based learning has been shifting to self-supervised learning (SSL), which labels images without human intervention. More recent approaches (e.g., DeepCluster [4], MoCo [12], and SimCLR [5]) are closer to a human-based supervision on ImageNet. Automatic annotation based on SSL is a highly promising method for replacing human supervision in the pre-training phase. In the context of SSL, large-scale image datasets, such as ImageNet, are usually employed. However, entire large-scale image datasets (80M Tiny Images [34] as well as human-related labels in ImageNet [36]) were recently removed due to the previously mentioned reasons regarding ethical concerns. In terms of supervised and self-supervised learning, we cannot overlook the fact that most image datasets consist of natural images. Hence, we must consider how to replace the natural image datasets in order to create more ethical models. How can we get a good representation to improve recognition with a natural image dataset through pre-training without any natural images or human annotations? Unlike the conventional frameworks with supervised, self-supervised, and other learning approaches, Kataoka *et al.* introduced Formula-Driven Supervised Learning (FDSL) [17]. The learning framework creates a sophisticated pre-trained CNN model without using any natural images or human annotations. In relation to synthetic image datasets, FDSL is also different from Domain Randomization (e.g., [33, 31]). Though most synthetic image datasets are defined by properties such as object models, background images, light conditions and view-



(a) ImageNet. (b) Tiles. (c) Fractals. (d) Perlin noise. (e) Bezier curves.



(f) PT on ImageNet. (g) PT on tiles. (h) PT on fractals. (i) PT on Perlin noise. (j) PT on Bezier curves.

Figure 1. Comparisons of feature representations between pre-training with supervised dataset (a) and FDSL datasets (b)–(e). According to our experimental results, the models pre-trained with FDSL can be a close representation to a model pre-trained with human-annotated dataset. Surprisingly, pre-training (PT) representations on tiles [27] are "obviously" similar to the representations of PT on ImageNet, while the two image sets (a) and (b) are quite different. Also, the performance rates with the TileDB pre-trained model are relatively close to those from the ImageNet/Places pre-trained models (e.g., TileDB 78.0 vs. ImageNet 79.5 on Places-30).

points, FDSL does not require manual definition of, for example, object category and background. In light of these discussion, Baradad *et al.* proposed the concept of 'learning to see by looking at noise' which supports the FDSL framework. They employed external training labels with SSL in addition to the usage of a large amount of synthetic images [2].

If we could improve the framework of pre-training models without natural images, then human-annotated dataset pre-trained models may be replaced to preserve privacy and decrease annotation labor. The original FDSL paper [17] explained how to automatically construct a synthetics dataset and create pre-trained CNN models with fractal geometry. We believe that the concept has great potential. However, on the other hand, the FractalDB pre-trained model heavily relied on a large amount of parameter tuning in the original paper. We highly require a knowledge how to generate a compact FDSL dataset without heavy parameter tunings, and with a fewer parameters. Therefore, we propose a new principle for FDSL which does not involve fractal geometry. As shown in Figure 1, we implemented and compared three types of formulas, namely Tile [27] (ours), PerlinNoise [28] and BezierCurves [9]. We assigned automatic texture- and edge-renderers in the proposed methods [17]. In the present paper, we propose Tile DataBase (TileDB) which simultaneously implements both characteristics (texture and edge). Perlin noise and Bezier curves then correspond to texture and edge features, respectively. Here, the pre-training feature representation on TileDB is clearly similar to that of the ImageNet pre-trained model (see Figure 1(f) and 1(g)), even though the two input images are non-trivially different (see Figure 1(a) and 1(b)). This allows us to construct ImageNet-like basis feature (Convolutional layer 1 in ResNet-50) without any natural images in the pre-training phase.

The present paper makes the following contributions.

(1) We propose a new FDSL family, namely, TileDB, which can be constructed from fewer hyperparameters compared to FractalDB, the conventional FDSL. The proposal of FDSL family is worthwhile to propose such a new pretext task in SSL. (2) Through experimental results, we confirmed that TileDB performs similarly to a related labeling method based on DeepCluster and 100k-order supervised training with ImageNet-100 and Places-30 (see Table 2). Further, TileDB is superior to FractalDB in relatively small dataset configurations (see Figure 5(a)). (3) Our TileDB acquires a good feature representation through pre-training without any natural images or human annotations. In the visualization of the first convolutional filter, TileDB pre-trained features are similar to ImageNet pre-trained features on ResNet-50.

## 2. Related Work

**Learning Frameworks.** In image classification, the most promising framework is supervised learning with human-

annotated datasets, such as ImageNet [6], Places [38], OpenImages [18], Pascal VOC [8], and MSCOCO [22]. On the one hand, CNN models have contributed to the extraction of well-defined feature representations in a large-scale image dataset [20, 30, 32, 13, 35, 15, 29, 14]. The research community is considering how to replace the human annotations with self-generated labels, namely the SSL research sub-field. The learning framework can be used to automatically self-generate labels for unlabeled images. The research community already has simple yet effective methods for performing pre-text tasks [7, 24, 26, 37, 25, 11]. More recent SSL methods, such as DeepCluster [4], MoCo [12], and SimCLR [5], have been improving the pre-text tasks to make them closer to pre-training with human-annotated datasets.

Unlike the studies on SSL, the proposed learning framework can be done without any (pre-)training images and labels. A more recent study discussed SSL with single images [1]; therefore, we believe that the pre-training (pretext task in SSL) can be done without any natural images. In addition to the self-generated labels that SSL creates, our training on FDSL enables the automatic rendering of training images based on a mathematical formula.

Moreover, in the experimental section, we compare our proposed method with ImageNet and Places pre-trained models as a representation for human-annotated datasets. We additionally assign ImageNet-100 and Places-30 which are randomly selected 100/30 categories on ImageNet and Places, respectively. We let the dataset contain over 100k-order images in order to evaluate the pre-training effects. On one hand, we mainly compare our proposed method with DeepCluster in the context of SSL. Our method is similar to DeepCluster in terms of automatic labeling under certain rules. Our TileDB follows the tiling parameters, and DeepCluster is categorized based on K-means clustering in features of training images.

**Formula-Driven Supervised Learning (FDSL).** To promote a learning framework without any human supervision, self-supervision, or natural images, we replaced the method of annotation and image representation with well-defined mathematical formulas. There are two representative work in this topic [17, 2]. Kataoka *et al.* originally proposed the concept of 'pre-training without natural images' in order to replace the conventional annotatioin and image representation. Their work achieved that a CNN was allowed to pre-train with a large number of synthetic fractal images and their corresponding labels. The more recent study has reported the architecture can be effectively used by vision transformers [23]. Baradad *et al.* have followed the work with several types of noise images (e.g. Dead Leaves [21], StyleGAN images [16]) in order to train visual representations from synthetic images [2]. The research tried to implement the combination of synthetic noise images and SSL labels with MoCoV2.

In the present paper, though detailed descriptions are given for dataset generation (Section 3), we here introduce the three types of mathematical formulas. One of the well-organized FDSL is tiling patterns with various colors [27] (see Figure 1(b)). For dataset creation with tiling patterns, we use images with clear edges, colors, and textures. The image patterns are flexibly changed after starting from basic hexagon patterns. The other well-known FDSL are Perlin noise [28] and Bezier curves [9], both of which are employed to render any pattern in computer graphics. We represent a textured feature with Perlin noise and an edge feature with Bezier curves.

The success of studies that relied on FDSL supports our assumption that tiles, Bezier curves, and Perlin noise can help make learning image representations for recognizing natural scenes and objects without any natural images or human annotations. Our idea in FDSL is that the intervals of parameters are equivalent to the image categories for training an image classifier with a CNN.

## 3. Proposed method

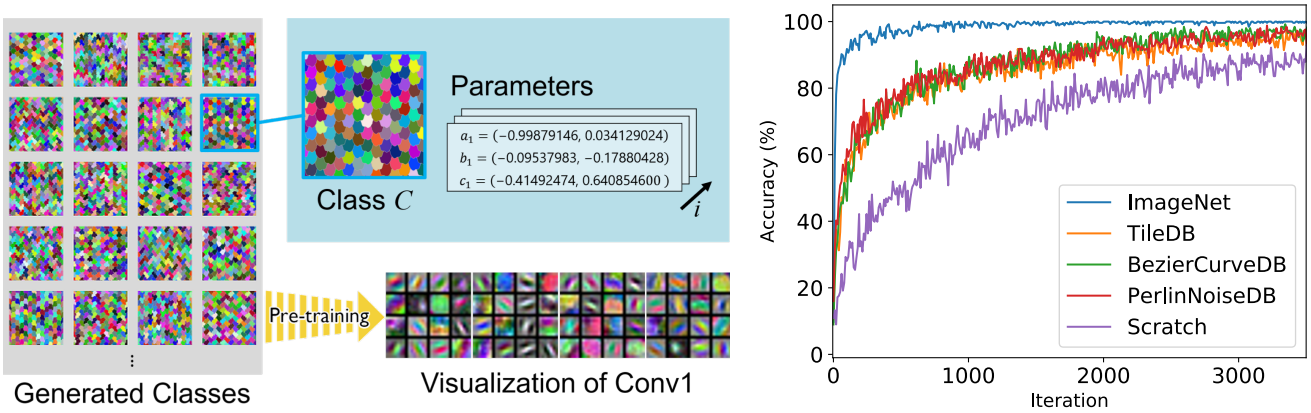
### 3.1. Overview

In this section, we introduce how to construct TileDB which consists of recursive tiling patterns. An overview of our method is shown in Figure 2(a). Both training images and their categories are automatically generated with a mathematical formula. In the example in Figure 2(a), basic hexagon patterns and their randomly changing points (parameters $a$, $b$, and $c$) generate image patterns and their categories as a TileDB. Based on the parameter intervals, the various (pre-)training categories are assigned in the mathematical formula. Moreover, the effects of FDSL are shown in Figure 2(b). We compare the TileDB pre-trained model with the other FDSL approaches, scratch training and ImageNet pre-trained model. Similar to the conventional work [17], the accuracy is higher than scratch training and it becomes to be a close accuracy in a longer training.

### 3.2. Dataset generation

The TileDB shown in Figure 1(b) comprises patterns created with tiles [27]. A tile is a wallpaper group that is a mathematical classification of a two-dimensional repetitive pattern and complex textures are created by adding three operations to regular hexagonal tiles: moving vertices, deforming edges, and moving symmetrically in a specular direction. Using this simple method we created a dataset with 1,000 categories and 1,071 images per category.

Patterns are created based on arranged regular hexagonal clusters. More complex textures can be expressed by specular inverting and edge deforming to these regular hexagons. Specular inverting is horizontally flipping for each column




(a) Overview of FDSL. We automatically generate a large-scale labeled image dataset based on mathematical formulas.

(b) The pre-training effect under the supervision of FDSL. The figure indicates the accuracy transitions on CIFAR-10 dataset with ImageNet, TileDB pre-training models and training from scratch.

Figure 2. FDSL and the effects of pre-training.

of tiles in Figure 3(a). We define each side of an ordinary regular hexagon ($F$) as $a \sim f$ clockwise, and it can be seen that the order is reversed for the inverted regular hexagon ($\exists$). As shown in Figure 3(a), the six parameters make the tiling shapes with the combination of curves in a hexagon. The parameters indicate the amount of shifting in an angle.

Next, these three pairs of sides are deformed. The deformation is performed using Bezier curves, which changes from a straight line to a cubic curve. In Bezier curves, the shape of a curve is determined by the position of points. The actual transformations were performed using four points to generate a cubic curve. Therefore, since two rotation angles are required for each of the three transformations, the pattern of the tile is determined by 6 values, which is the class in TileDB.

To create a dataset from FDSL, we define the categories of images and the way to create instances within each of the categories. The categories in TileDB were determined by the shape patterns of the tile edges. We assign categories in TileDB based on the parameters set ($a$-$f$). In TileDB pre-training phase, a CNN solves a classification task to categorize the tiling categories. TileDB has 1,000 categories to determine the parameters randomly. To create instances within each of the categories, we moved vertices of hexagons horizontally and vertically. To maintain the shape of the tiles, if a vertex is moved, it is moved in the opposite direction by the same amount as the vertex located diagonally. In the actual dataset, 51 horizontal and 21 vertical movements were determined, and each of them was combined to produce 1,071 images per category.

### 3.3. Training configuration

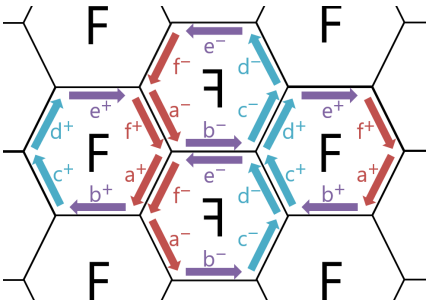
We assign a standard training and validation step in image classification (training: 90 epochs / validation: 90 epochs). That is, we simply pre-train TileDB containing automatically labeled images. In the fine-tuning step, we additionally train with the pre-trained parameters on target datasets. The flow is based on the conventional transfer learning in CNNs, such as ImageNet / Places pre-trained models.

Our FDSL is different from the standard SSL testing in terms of freezing layers. Since our proposed method is not pre-trained by natural images, full-layer fine-tuning is required in contrast to the setting of the standard SSL testing. At the same time, in our experiments, we assign the full-layer fine-tuning even if the approach is SSL method.

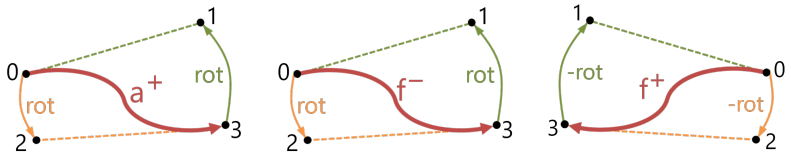
To confirm the properties of TileDB and compare the pre-trained models with results of previous studies, we used the ResNet-50 [13]. We simply replaced the pre-training phase with our FDSL without changing the fine-tuning step. For pre-training and fine-tuning, we assigned a value of 0.9 to the momentum of the stochastic gradient descent [3], and a basic batch size of 256. The initial learning rate is set to 0.1 for pre-training and 0.01 for fine-tuning. Both learning rates were multiplied by 0.1 when the learning epoch reached 30 and 60. Training was performed up to epoch 90. Moreover, the input image size was cropped to $224 \times 224$ pixels from an input image with $256 \times 256$ pixels.

## 4. Experiment

We evaluated the pre-training effectiveness of the TileDB pre-trained model. The performance rates during the fine-tuning phase are shown in Tab. 1 (in FDSL), and 2 (comparisons). With these experimental results, we verified the effectiveness of our FDSL without natural images, SSL, and human annotations for (natural) image recognition tasks. We also show the feature representations with




(a) Deformation of a regular hexagonal tiling.




(b) Curve on two corresponding edges by tiling. In this tile, Edge $a^+$ and $f^-$ correspond, and the inverted pair is $f^+$ after rotation.

Figure 3. Method for making tile images.

Table 1. Classification accuracies of TileDB (proposed), PerlinNoiseDB, BezierCurveDB pre-trained models with respect to CIFAR-10/100 (C10/C100) [19], Places30 (P30) [38] and Pascal VOC 2012 (VOC12) [8] datasets.

| Pre-training | C10 | C100 | P30 | VOC12 |
|---|---|---|---|---|
| From scratch | 87.6 | 62.7 | 70.7 | 58.9 |
| BezierCurveDB [17] | 89.7 | 68.1 | 73.6 | 65.4 |
| PerlinNoiseDB [17] | 90.9 | 70.4 | 74.2 | **69.9** |
| TileDB (proposed) | **92.3** | **73.5** | **75.0** | 69.4 |

initial convolutional maps (see Figure 1).

Through the experiments, we assign representative datasets (CIFAR-10/100 [19] [1], Places-30 [38], and Pascal VOC 2012 [8]]) in image recognition. Places-30 is a subset of Places dataset which consists of 30 randomly selected categories out of 365 categories.

### 4.1. Exploration study

First, we compare the proposed TileDB pre-trained model with from-scratch training from random parameters in Tab. 1. Regarding PerlinNoiseDB (Figure 1(d)) and BezierCurveDB (Figure 1(e)), these two datasets have different characteristics, namely, texture- and edge-based projection with mathematical formulas. The BezierCurveDB pre-trained model outperformed training from scratch. Here, we can see the PerlinNoiseDB pre-trained model is better than the BezierCurveDB pre-trained model. As discussed in [10], the texture-based representation tends to be advantageous in image recognition tasks.

The TileDB pre-trained model tends to achieve better rates than that of the PerlinNoiseDB. A reason why TileDB is more accurate than PerlinNoiseDB is the representation with Gabor-like and color-based features in the Conv1 maps. From the ImageNet map (Figure 1(f), the accuracy tends to increase with textured patterns [10]. Moreover, the Gabor-like features are said to be good representation for natural image recognition. Although the BezierCurveDB has Gabor-like features, the representation is quite partial and grayscale. TileDB also utilizes color representation in addition to the textured patterns. These characteristics allow us to acquire representations similar to those of the ImageNet pre-trained model. Indeed, imitating the representation in the basis filter of the ImageNet pre-trained model is one of our objectives.

Figure 4 illustrates the relationships between accuracy and category/instance in the configuration of TileDB on CIFAR-10 / 100 datasets. We confirm that the constant #category (1,000 categories) and varied #instance {16, 32, 64, 128, 256, 512, 1000} is better than vice versa. That is, #category is more effective for improving the TileDB. We further evaluate larger categories of 3,000 and 5,000. Although the accuracies were partially improved with 3,000 categories in TileDB, reaching 92.54 on CIFAR-10, the accuracies were saturated with 5,000 categories at 91.6.

### 4.2. Comparisons

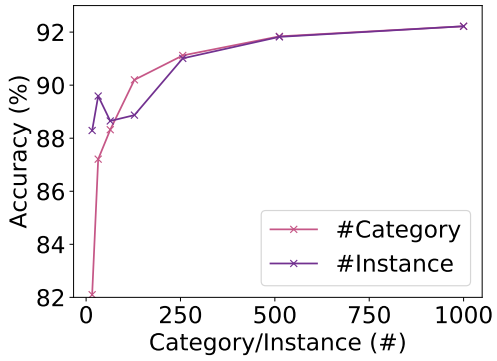
**Comparisons with conventional work.** We compared our proposed method with representative pre-trained models in supervised and self-supervised learning. The models compared are from-scratch training using random parameters, Places-30/365 [38], ImageNet-100/1k (ILSVRC'12) [6], DeepCluster [4], and TileDB. ImageNet-100 is a model trained on a random selection of 100 categories from ImageNet-1k. For DeepCluster, we pre-trained the model with both natural images of ImageNet and formula-generated images of TileDB with 1,000-means clustering. The properties of pre-training and fine-tuning followed Section 3.3. Note that the accuracy of the fine-tuning for the from-scratch and ImageNet pre-trained models could be different from the related papers since the training was performed under identical conditions for comparison.

Table 2 lists the comparisons between these pre-trained

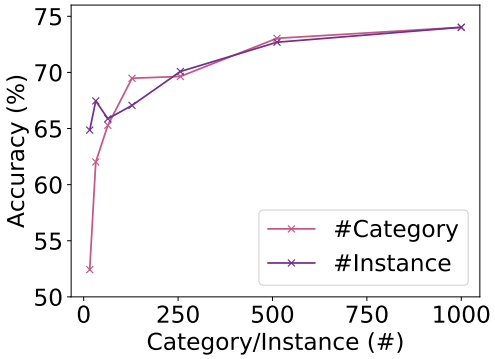
[1] For the experimental setting on CIFAR-10/100, our implementation is different in terms of input, kernel, stride, and padding sizes. We aligned all architectures and their hyperparameters for a fair comparison. The experimental settings is following to the previous paper [17].

Table 2. Classification accuracies of TileDB (proposed), from-scratch, DeepCluster, ImageNet-100/1k and Places-30/365 pre-trained models. We show the types of pre-trained image (Image) and annotated label (Label).

| Pre-training dataset | Image | Label | C10 | C100 | P30 | VOC12 |
|---|---|---|---|---|---|---|
| From scratch | – | – | 87.6 | 62.7 | 70.7 | 58.9 |
| DeepCluster | Natural Image | Self-supervision | 89.9 | 66.9 | 75.1 | 67.5 |
| DeepCluster | Formula | Self-supervision | 83.1 | 57.0 | 72.8 | 60.4 |
| Places-30 | Natural Image | Human-supervision | 90.1 | 67.8 | – | 69.5 |
| Places-365 | Natural Image | Human-supervision | 94.2 | 76.9 | – | 78.6 |
| ImageNet-100 | Natural Image | Human-supervision | 91.3 | 70.6 | – | 72.0 |
| ImageNet-1k | Natural Image | Human-supervision | **96.8** | **84.6** | **79.5** | **85.8** |
| TileDB (proposed) | Formula | Formula-supervision | **92.5** | **73.7** | **78.0** | **71.4** |

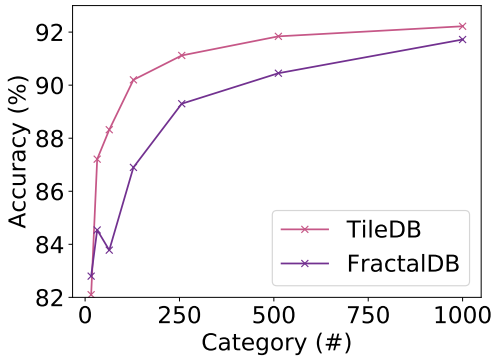



(a) CIFAR-10.

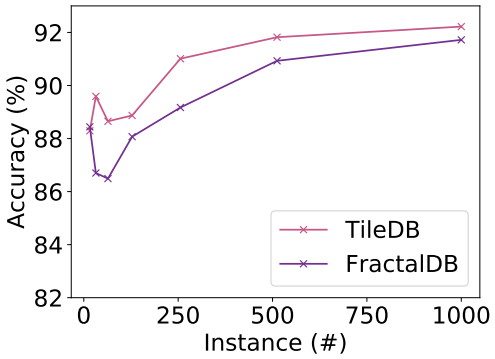



(b) CIFAR-100.

Figure 4. Parameter tuning in #category/#instance on CIFAR




(a) TileDB vs. FractalDB in category.




(b) TileDB vs. FractalDB in instance.

Figure 5. Relationship between accuracy and category/instance on CIFAR-10. We compare the TileDB and FractalDB pre-trained models.

models. The proposed pre-trained model tends to be more accurate than ImageNet-100/Places-30 pre-trained models, for example, TileDB 92.5 vs. ImageNet-100 91.3/Places-30 90.1 on CIFAR-10 and TileDB 73.7 vs. ImageNet-100 70.6/Places-30 67.8 on CIFAR-100. Based on these results, the performance of the model pre-trained with TileDB reaches that of a model pre-trained with 100k-order natural images (ImageNet-100 and Places30 pre-trained models).

Pre-training with TileDB achieved good scores in some respects. The TileDB pre-trained model achieved similar values with the Places-365 pre-trained model on CIFAR-10 (TileDB 92.5 vs. Places 94.2), the ImageNet pre-trained model on Places-30 (TileDB 78.0 vs. ImageNet 79.5). In this way, natural image recognition can be enhanced by TileDB pre-trained model.

We assigned the DeepCluster since the automatic categorization is similar to our method which is based on a simple mathematical rule. TileDB varies the parameters to create tiling patterns. On one hand, DeepCluster assigns categories under K-means clustering. The TileDB pre-trained

models outperform DeepCluster with natural and formula images. Therefore, the way that categories are defined in TileDB is more effective than clustering.

**Comparisons with FractalDB.** Figure 5 illustrates the relationship between performance rates on CIFAR-10 and #category/#instance configurations in pre-training. Note that the we made these figures in our implementation. The final accuracy in FractalDB-10k (94.1%) [17] is better than the TileDB pre-trained model, however, TileDB pre-training with fewer #category/#instance surpasses the FractalDB pre-training on CIFAR-10 dataset. This shows that TileDB does not require a relatively large dataset like FractalDB to create a pre-trained model. According to the results, we confirmed the TileDB pre-trained model can be made without any difficult parameter tunings like a FractalDB pre-trained model. There contains only three main parameters in TileDB (see Figure 3). Moreover, the graphs describe that the TileDB tends to be compact dataset configuration in pre-training phase. It is better way to easily construct a pre-trained model.

**Visualization of initial convolutional layer.** Moreover, we compared both FDSL and ImageNet in terms of the initial convolutional maps. In TileDB, PelinNoiseDB, and BezierCurveDB, these representations are different from each other. This means that FDSL is capable of being activated by different features in an image, depending on the type of mathematical formula. This feature has significant potential for flexibly changing feature representations in the future. In this paper, our focus is on reproducing an ImageNet-like feature representation; therefore, the TileDB pre-training is visualizing very similar patterns (see Figure 1(f) and 1(g)). In fact, the TileDB pre-trained model performed better than a model pre-trained on FractalDB in a relatively compact ($<$ 1,000 categories) dataset configuration. On the other hand, in a different way from the ImageNet-like features, FractalDB enabled to a better image representation than TileDB pre-training in a larger 10,000 categories $\times$ 1,000 instances dataset.

## 5. Conclusion

We proposed TileDB, a Formula-Driven Supervised Learning (FDSL) that consists of tiling patterns. Our TileDB pre-trained model performed much better than a model trained from scratch and performed similarly to pre-trained models with 100k-order supervised datasets (ImageNet-100 and Places-30 pre-trained models). Moreover, visualizations of the first convolutional maps between TileDB and ImageNet pre-trained models are similar to each other even though the pre-training images are non-trivially different.

The framework of FDSL enables to construct a trustworthy pre-trained model in an annotation-free, fairer and more transparent dataset. The framework of FDSL is defined as "automatically generate image patterns and their category labels by assigning mathematical formula", therefore, we do not require any human annotation in addition to natural images. A fairness problem must be alleviated in pre-training phase since any human-related labels do not appear in a dataset. Obviously, a dataset transparancy is clear due to the creation can be done by mathematical formula.

## Acknowledgements


- This paper is based on results obtained from a project, JPNP20006, commissioned by the New Energy and Industrial Technology Development Organization (NEDO).
- This work was supported by JSPS KAKENHI Grant Number JP19H01134.
- AI Bridging Cloud Infrastructure (ABCI) provided by the National Institute of Advanced Industrial Science and Technology (AIST) was used.


## References


[1] Y. M. Asano, C. Rupprecht, and A. Vedaldi. A critical analysis of self-supervision, or what we can learn from a single image. In *International Conference on Learning Representations (ICLR)*, 2020.

[2] Manel Baradad, Jonas Wulff, Tongzhou Wang, Phillip Isola, and Antonio Torralba. Learning to see by looking at noise, 2021.

[3] L. Bottou. Large-Scale Machine Learning with Stochastic Gradient Descent. In *19th International Conference on Computational Statistics (COMPSTAT)*, 2010.

[4] M. Caron, P. Bojanowski, A. Joulin, and M. Douze. Deep Clustering for Unsupervised Learning of Visual Features. In *European Conference on Computer Vision (ECCV)*, 2018.

[5] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A Simple Framework for Contrastive Learning of Visual Representations. In *International Conference on Machine Learning (ICML)*, 2020.

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *The IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[7] C. Doersch and A. A. Gupta, A.and Efros. Unsupervised Visual Representation Learning by Context Prediction. In *The IEEE International Conference on Computer Vision (ICCV)*, 2015.

[8] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The Pascal Visual Object Classes Challenge: A Retrospective. *International Journal of Computer Vision (IJCV)*, 2015.

[9] G. Farin. Curves and surfaces for computer aided geometric design: A practical guide. *Academic Press*, 1993.

[10] R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. A Wichmann, and W. Brendel. Imagenet-trained CNNs are biased

towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representation (ICLR)*, 2019.

[11] S. Gidaris, P. Singh, and N. Komodakis. Unsupervised Representation Learning by Predicting Image Rotations. In *International Conference on Learning Representation (ICLR)*, 2018.

[12] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. Momentum Contrast for Unsupervised Visual Representation Learning. In *The IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[13] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. In *The IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[14] A. G. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, W. Tan, M.and Wang, Y. Zhu, R. Pang, V. Vasudevan, Q. V. Le, and H. Adam. Searching for MobileNetV3. In *The IEEE International Conference on Computer Vision (ICCV)*, 2019.

[15] A. G. Howard, Chen B. Zhu, M., D. Kalenichenko, W. Wang, T. Weyand, M. An-dreetto, and H. Adam. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. In *CoRR abs/1704.04861*, 2017.

[16] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[17] H. Kataoka, K. Okayasu, A. Matsumoto, E. Yamagata, R. Yamada, N. Inoue, A. Nakamura, and Y. Satoh. Pre-training without natural images. In *Asian Conference on Computer Vision (ACCV)*, 2020.

[18] I. Krasin, T. Duerig, N. Alldrin, V. Ferrari, S. Abu-El-Haija, A. Kuznetsova, H. Rom, J. Uijlings, S. Popov, S Kamali, M. Malloci, J. Pont-Tuset, A. Veit, S. Belongie, V. Gomes, A. Gupta, C. Sun, G. Chechik, D. Cai, Z. Feng, D. Narayanan, and K. Murphy. OpenImages: A public dataset for large-scale multi-label and multi-class image classification. *Dataset available from https://storage.googleapis.com/openimages/web/index.html*, 2017.

[19] A. Krizhevsky. Learning Multiple Layers of Features from Tiny Images. *University of Toronto*, 2009.

[20] A. Krizhevsky, Ilya Sutskever, and G E Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems (NIPS) 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[21] A. B Lee, D. Mumford, and J. Huang. Occlusion models for natural images: A statistical study of a scale-invariant dead leaves model. *International Journal of Computer Vision (IJCV)*, 1(41), 2001.

[22] T. Lin, M. Maire, S. J. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C. L. Zitnick. Microsoft COCO: common objects in context. In *European Conference on Computer Vision (ECCV)*, pages 740–755, 2014.

[23] Kodai Nakashima, Hirokatsu Kataoka, Asato Matsumoto, Kenji Iwata, and Nakamasa Inoue. Can vision transformers learn without natural images? In *CoRR:2103.13023*, 2021.

[24] M. Noroozi and P. Favaro. Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles. In *European Conference on Computer Vision (ECCV)*, 2016.

[25] M. Noroozi, H. Pirsiavash, and P. Favaro. Representation Learning by Learning to Count. In *The IEEE International Conference on Computer Vision (ICCV)*, 2017.

[26] M. Noroozi, A. Vinjimoor, P. Favaro, and H. Pirsiavash. Boosting Self-Supervised Learning via Knowledge Transfer. In *The IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[27] M.L. A. D. L. Penas and A. F. Guzon. Tilings, patterns and technology. In *Asian Technology Conference in Mathematics (ATCM)*, 2011.

[28] K. Perlin. Improving Noise. *ACM Transactions on Graphics (TOG)*, 2002.

[29] M. Sandler, A. G. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen. MobileNetv2: Inverted Residuals and Linear Bottlenecks. Mobile Networks for Classification, Detection and Segmentation. In *CoRR abs/1801.04381*, 2018.

[30] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *International Conference on Learning Representations (ICLR)*, 2015.

[31] M. Sundermeyer, Z.-C. Marton, M. Durner, M. Brucker, and R. Triebel. Implicit 3D Orientation Learning for 6D Object Detection from RGB Images. In *European Conference on Computer Vision (ECCV)*, 2018.

[32] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going Deeper with Convolutions. In *The IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[33] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel. Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World. In *International Conference on Intelligent Robots and Systems (IROS)*, 2017.

[34] A. Torralba, R. Fergus, and W. T. Freeman. 80 million tiny images: a large dataset fornon-parametric object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2008.

[35] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He. Aggregated Residual Transformations for Deep Neural Networks. In *The IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[36] K. Yang, K. Qinami, L. Fei-Fei, J. Deng, and Olga Russakovsky. Towards Fairer Datasets: Filtering and Balancing the Distribution of the People Subtree in the ImageNet Hierarchy. In *ACM Conference on Fairness, Accountability, and Transparency (FAT*)*, 2020.

[37] R. Zhang, P. Isola, and A. A. Efros. Colorful Image Colorization. In *European Conference on Computer Vision (ECCV)*, 2016.

[38] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba. Places: A 10 million Image Database for Scene Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2017.