Can Vision Transformers Learn without Natural Images?

Kodai Nakashima[1]* Hirokatsu Kataoka[1]*
Asato Matsumoto[1] Kenji Iwata[1] Nakamasa Inoue[2]
National Institute of Advanced Industrial Science and Technology (AIST)[1]
Tokyo Institute of Technology[2]
{nakashima.kodai, hirokatsu.kataoka, matsumoto-a, kenji.iwata}@aist.go.jp
inoue@c.titech.ac.jp

Abstract

Can we complete pre-training of Vision Transformers (ViT) without natural images and human-annotated labels? Although a pre-trained ViT seems to heavily rely on a large-scale dataset and human-annotated labels, recent large-scale datasets contain several problems in terms of privacy violations, inadequate fairness protection, and labor-intensive annotation. In the present paper, we pre-train ViT without any image collections and annotation labor. We experimentally verify that our proposed framework partially outperforms sophisticated Self-Supervised Learning (SSL) methods like SimCLRv2 and MoCov2 without using any natural images in the pre-training phase. Moreover, although the ViT pre-trained without natural images produces some different visualizations from ImageNet pre-trained ViT, it can interpret natural image datasets to a large extent. For example, the performance rates on the CIFAR-10 dataset are as follows: our proposal 97.6 vs. SimCLRv2 97.4 vs. ImageNet 98.0. The codes, datasets, and pre-trained models will be publicly available[1]

1. Introduction

In contemporary visual recognition, a transformer architecture [35] is gradually replacing the usage of convolutional neural networks (CNNs). The latter have been considered as central to the field of computer vision (CV). For example, the Residual Network (ResNet) [15] is one of the de-facto-standard models in a wide range of visual tasks including image classification. The current high-standard scores on ImageNet are ResNet-based architectures, e.g., EfficientNet [32, 11, 26], BiT [18].

Transformer architecture, which consists of a self-attention mechanism, was initially employed in natural language processing (NLP) tasks such as machine translation and semantic analysis. We have witnessed the development of epoch-making methods, e.g., BERT [8], GPT-{1, 2, 3} [27, 28, 2] with transformer modules. The trend is gradually shifting from NLP to CV. One of the most active topics is undoubtedly Vision Transformers (ViTs) for image classification [10]. ViTs effectively process and recognize an image based on transformers with minimum modifications. Even though the re-implementation is reasonably straightforward, it has been shown that ViTs often perform at least as well as state-of-the-art transfer learning. However, it is noteworthy that ViT architectures tend to require a large amount of data in the pre-training phase. Dosovitskiy *et al.* [10] reported that unless ViT is pre-trained with more than 100 million images, the accuracy is inferior to CNN. The pre-training problem is somewhat alleviated by virtue of the Data-efficient image Transformer (DeiT) [34].

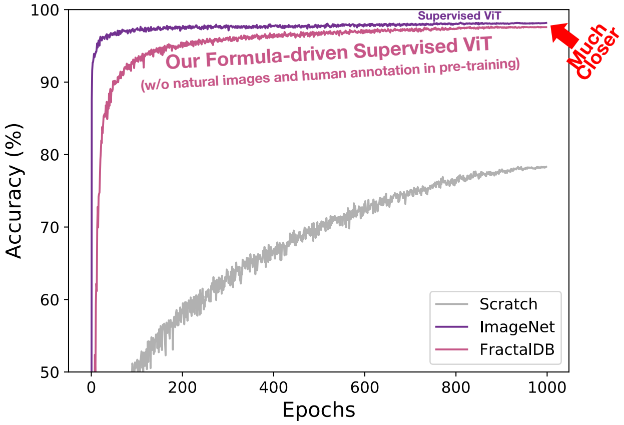
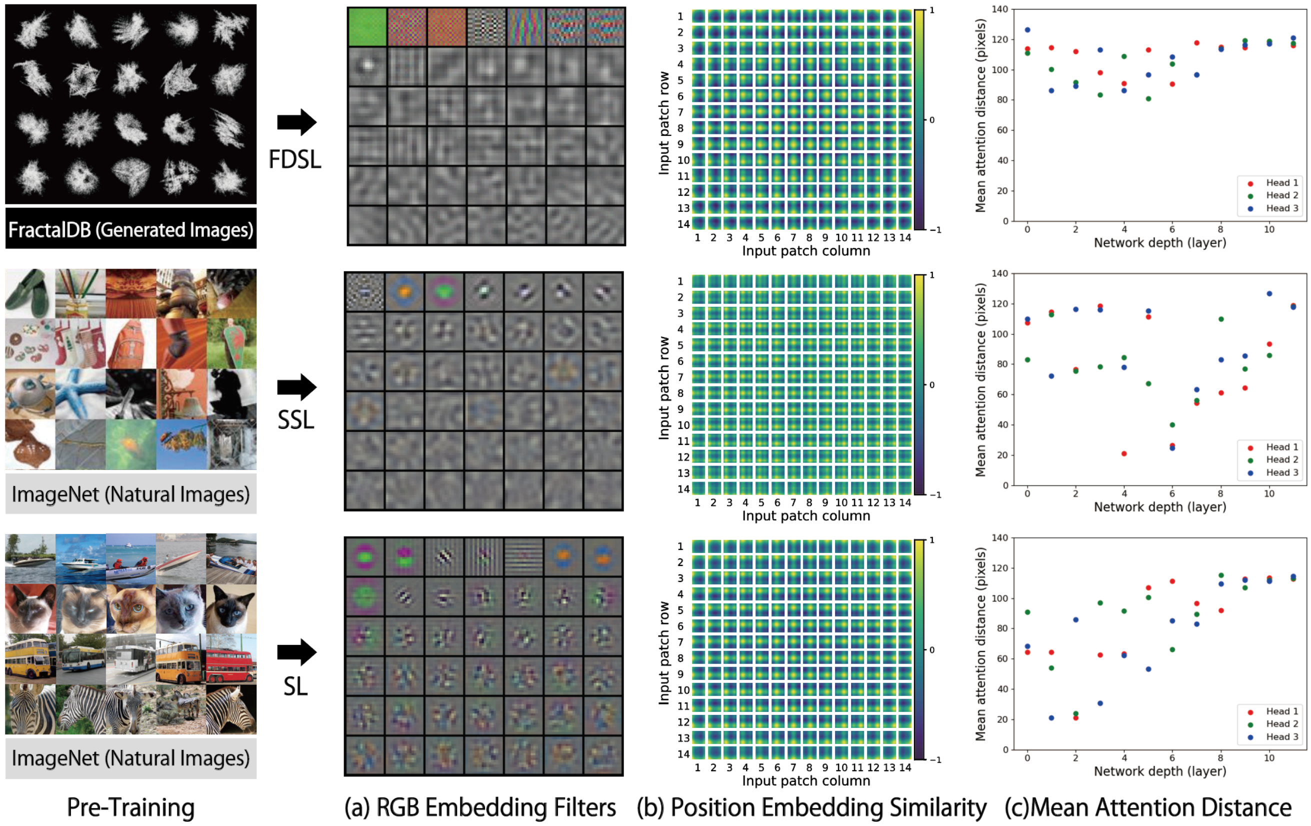


Figure 1. Accuracy transition in the fine-tuning phase. The graph illustrates that FractalDB-1k pre-trained ViT exhibits much higher training accuracy in early training epochs. The accuracy of FractalDB-1k is similar to that of ImageNet-1k pre-training.

On the other hand, using a large-scale image dataset may be problematic from the perspective of privacy preserva-

*indicates equal contribution

[1] https://hirokatsukataoka16.github.io/Vision-Transformers-without-Natural-Images/.



Pre-Training (a) RGB Embedding Filters (b) Position Embedding Similarity (c)Mean Attention Distance

Figure 2. Overview: We consider whether Vision Transformer (ViT) can pre-train without natural images. Following the previous paper [10], we list the (a) RGB embedding filters, (b) position embedding similarity, and (c) mean attention distance in the frameworks of Formula-Driven Supervised Learning (FDSL) with FractalDB, Self-Supervised Learning (SSL) with SimCLRv2, and Supervised Learning (SL) with human-annotated ImageNet. We replace pre-training based on human-labeled image datasets with FDSL. Compared to SSL and SL, FDSL pre-trained ViT enables the acquisition of slightly different filters, the same position embedding, and a wider receptive field.

tion, annotation labor, and AI ethics. In fact, representative datasets including natural images taken by a camera are limited to academic or educational usage. The problem cannot be solved even if we assign Self-Supervised Learning (SSL), e.g., MoCo [13, 6], SimCLR [4, 5], and SwAV [3], for automatic labeling of natural images. Training on natural image datasets still raises concerns in the context of privacy-violation and fairness-protection. Other large-scale datasets (e.g., Human-related images in ImageNet [37] and 80M Tiny Images [33][2]) are being deleted due to issues regarding AI ethics. To date, huge datasets such as JFT-300M and Instagram-3.5B [21] are not publicly available. Additionally, YFCC-100M has apparently withdrawn dataset access rights[3]. These dataset-related problems significantly limit the opportunities for research in this domain. The research community must carefully consider large-scale datasets in terms of availability and reliability while overcoming dataset-related problems.

In this context, Formula-Driven Supervised Learning (FDSL) was proposed in late 2020 [17]. The concept involves automatically generating image patterns and their labels based on a mathematical 'formula' which includes rendering functions and real-world rules. In the original paper, Kataoka *et al.* clarified that fractal geometry [22, 1] was the best way to construct a dataset in the framework of FDSL. Therefore, in the present paper, we consider whether ViTs can be pre-trained with FractalDB in FDSL. Although disadvantages of FractalDB pre-trained CNN have hitherto been pointed out[4], we believe that the vision transformers can successfully be pre-trained with the FDSL framework because the self-attention mechanism enables elimination of the background effects between fractal and natural images, and can understand entire fractal shapes which consist of iteratively recursive patterns. Figure 1, 2 illustrate the accuracy transition and characteristics of training properties.

The contributions of the paper are as follows: We clarify that the FractalDB under the FDSL framework is

[2] https://groups.csail.mit.edu/vision/TinyImages/

[3] 'Core Dataset' in http://mmcommons.org/

[4] https://www.technologyreview.com/2021/02/04/1017486/

more effective for ViT compared to CNN. The performance of FractalDB-10k pre-trained ViT is similar to those approaches with supervised learning (see Table 7), and slightly surpasses the self-supervised ImageNet with SimCLRv2 pre-trained ViT (see 'Average' in Table 8). Here, on the CIFAR-10 dataset the scores are as follows: FractalDB 97.6 vs. SimCLRv2 97.4 vs. ImageNet 98.0. Importantly, the FractalDB pre-trained ViT does not require any natural images and human annotation in the pre-training.

2. Related work

We would like to discuss a couple of topics in visual transformers and pre-training datasets. We mainly focus on architectures and large-scale datasets for image classification.

2.1. Network Architectures for Image Recognition

Convolutional Neural Networks (CNN) are popular in visual recognition. Several well-defined structures have emerged through a large number of trials in this decade [19, 30, 31, 15, 36, 16, 32]. Very recently, at the end of 2020, the architecture shifted to transformers [35] originating from natural language processing. Transformers basically consist of several modules with multi-head self-attention layers and Multi-Layer Perceptron blocks. The mechanism has enabled the construction of revolutionary models (e.g., BERT [8], GPT-$\{1, 2, 3\}$ [27, 28, 2]). Thus, the computer vision community is focusing on replacing the defacto-standard convolutions with a transformer-based architecture. One of the most insightful architectures is the Vision Transformer (ViT) [10]. Though the ViT is a basic transformer architecture in terms of image input, the model performs comparably to state-of-the-art alternative approaches on several datasets. However, ViT requires over ten-million-order labeled images in representation learning. The JFT-300M/ImageNet-21k pre-trained ViT was verified by experiments to perform well in terms of accuracy. The issue of learning with large-scale datasets was alleviated with the introduction of the Data-efficient image Transformer (DeiT) [20]. However, the pre-training problem still remains in image classification.

2.2. Image dataset and training framework

It is said that the deep learning era started from ILSVRC [29]. Undoubtedly, transfer learning with large-scale image datasets has contributed to accelerating visual training [14]. Initially, the ImageNet [7] and Places [40] pre-trained models were widely used for diverse tasks, not limited to image classification. However, even in million-scale datasets, there exist several concerns such as AI ethics and copyright problems, e.g., fairness protection, privacy violations, and offensive labels. Due to these sensitive issues, as mentioned above, human-related labels in ImageNet [37] and 80M Tiny Images [33] were deleted. We must pay attention to the terms of use in large-scale image datasets and create pre-trained models accordingly.

On one hand, to alleviate the image labeling labor required of human annotators, Self-Supervised Learning (SSL) progressed significantly in recent years. The early methods created pseudo labels based on semantic concepts [9, 23, 25, 24, 12] and trained feature representations through image reconstruction [39]. By contrast, the SSL methods are closer to supervised learning with human annotations in terms of performance rates (e.g., MoCo [13, 6], SimCLR [4, 5], SwAV [3]). In this context, Formula-Driven Supervised Learning (FDSL) [17] was proposed to overcome the problems of AI ethics and copyrights in addition to annotation labor. The framework is similar to self-supervised learning. However, FDSL methods do not require any natural images taken by a camera. The framework simultaneously and automatically generates image patterns and the paired labels for pre-training image representations. We would like to investigate whether the formula-driven image dataset can sufficiently optimize a vision transformer in the pre-training phase. At the same time, we will compare the pre-trained ViT through the FDSL framework with supervised and self-supervised pre-training. If the supervised/self-supervised pre-training can be replaced by FDSL, vision transformers may be pre-trained without using any natural images in the future.

3. Vision transformer (ViT)

As mentioned in the previous sections, we believe that FractalDB pre-training can replace pre-training with natural image datasets by combining with ViT architecture. The FDSL framework including FractalDB enables automatic generation of an infinite number of training categories and their image labels with a mathematical formula. In the ViT characteristics, the FractalDB pre-trained ViT must be better than CNN. Additionally, although FractalDB does not contain a background area inside of the image, the self-attention mechanism effectively focuses on the fractal patterns while ignoring background areas. Moreover, the FractalDB pre-trained ViT is also better than the pre-training with natural image datasets in terms of privacy protection, AI ethics, and annotation labor. Here, we explore the potential of the transformer in visual tasks.

The basic transformer requires a 1D sequence of tokens in the input layer. To process 2D images, the image $x \in \mathbb{R}^{H \times W \times C}$ is reshaped into flattened image patches $x_p \in \mathbb{R}^{N \times (P^2 \cdot C)}$, where (H, W) is the original image size, C is the number of channels, (P, P) is the size of each image patch, and $N = HW/P^2$ is the number of patches. Flattened image patches are converted into D-dimensional vectors by a trainable linear projection and processed in a fixed dimension through all layers. After the linear projection, adding trainable 1D position embeddings to the patch representation and concatenate classification token similar to BERT becomes the input sequence of the transformer encoder.

The transformer encoder block consists of the multi-head self-attention layer and the Multi-Layer Perceptron (MLP). The self-attention, called scaled dot-product attention, firstly computes a set of queries $Q = XW_Q$, a set of keys $K = XW_K$, and a set of values $V = XW_V$, in which $X \in \mathbb{R}^{(N+1)\times D}$ is an input sequence, $W_Q \in \mathbb{R}^{D\times d}$, $W_K \in \mathbb{R}^{D\times d}$, $W_V \in \mathbb{R}^{D\times d}$ are trainable weights, and d is vector size. The self-attention is computed as follows:

$$\mathrm{Attention}(Q,K,V) = \mathrm{softmax}\left(\frac{QK^{\mathrm{T}}}{\sqrt{d}}\right)V \tag{1}$$

where the softmax function is applied over each row of the matrix. In multi-head self-attention (MSA), h heads are added to self-attention as follows:

$$\begin{aligned} \mathrm{MSA}(Q,K,V) &= \mathrm{concat}(head_1, ..., head_h)W \qquad (2) \\ head_h &= \mathrm{Attention}(Q_h, K_h, V_h) \qquad (3) \end{aligned}$$

Each head provides a sequence of size $(N+1)\times d$. These h sequences are rearranged into an $(N+1)\times dh$ sequence that is reprojected by an MLP into $(N+1)\times D$. In summary, the transformer encoder processes the image as follows:

$$\begin{aligned} z_0 &= \left[x_{\mathrm{class}}; \mathrm{MLP}(x_p^1); ...; \mathrm{MLP}(x_p^N)\right] + \mathrm{E}_{pos} && (4) \\ z'_l &= \mathrm{MSA}(\mathrm{Norm}(z_{l-1})) + z_{l-1} && (5) \\ z_l &= \mathrm{MLP}(\mathrm{Norm}(z'_l)) + z'_l && (6) \\ y &= \mathrm{Norm}(z_L^0) && (7) \end{aligned}$$

We conduct a characteristic evaluation on mechanisms in the ViT architecture such as linear embeddings, positional embeddings, and attention maps. Though the detailed characteristics are described and visualized in the experimental section, we generated interesting results. For example, the filters of the first linear embedding in FractalDB pre-training are different from the ImageNet pre-trained model (see Figure 2(a)). However, the positional embeddings (see Figure 2(b)) are similar. Moreover, the FractalDB pre-trained ViT focuses on an object-specific area to understand the object in an image because of the rendering process without any background area (see Figure 4). In the next section, we will explain data structure and generation for better understanding of FractalDB.

4. Formula-Driven Supervised Learning

This section presents Formula-Driven Supervised Learning (FDSL) for vision transformers (ViT). We begin from a brief review of FractalDB [17] under the framework of FDSL. We also describe how to apply the auto-generated pre-training dataset for ViT.

4.1. Definition

The goal of FDSL is to accomplish pre-training without any natural images. The framework automatically creates paired image patterns and their labels by following a mathematical formula. Unlike the pre-training framework in supervised learning, the FDSL does not require any natural images and human annotated labels. More specifically, FDSL is formulated as follows:

$$\underset{M}{\mathrm{argmax}}\, \mathbb{E}_{y,s}[\ell(M(x), y)] \text{ s.t. } x = F(\theta, s),\ y = \theta, \tag{8}$$

where M is a network to be pre-trained, ℓ is a loss function, x is a generated image pattern, and y is a label in the image. The image patterns are generated by a mathematical formula F, whose inputs are a parameter θ and a random seed s. The network learns to predict the parameter θ used to generate x. For simplicity, we assume that y follows a uniform distribution over a pre-defined discrete set of parameters $\Theta = \{\theta_k\}_k^K$. This allows us to introduce K-class classification loss such as cross-entropy loss for ℓ.

4.2. FractalDB

One of the most successful approaches in FDSL relies on fractals. FractalDB consists of 1k to 10k pairs of fractal images generated with the iterated function system (IFS) [1]. The reason that fractal geometry is chosen to generate the dataset is that the function can render complex patterns and different shapes for each parameter set.

In Equation 8, F and θ_i correspond to IFS and $(a_i, b_i, c_i, d_i, e_i, f_i, p_i)$, respectively. The parameters are randomly searched and will be adopted when the image patterns generated from the parameters exceed the threshold of the filling rate which is calculated by dividing the number of pixels of the fractal dot by the total number of pixels of the image. The intra-category instances are expansively generated by three methods for considering category configurations to maintain the shape in the category: varying the parameters slightly, rotation, and drawing with patch. Varying the parameters is the process of multiplying one of the 6 parameters of IFS by weights. We can generate the image from this parameter, which changes the detailed representation while maintaining the general shape of the category. By multiplying one of each parameter by 4 weights, 25 (original 1 + params 6 × weights 4) different variations of the image were generated. In the second method, rotation, we manipulate the flipping operation in the image. There are 4 rotations {none, horizontal flip, vertical flip, horizontal vertical flip}. Drawing with patch is the process of rendering the fractal image with patch instead of point. In FractalDB, 10 different 3 × 3 [pixel] patches were used to generate the fractal image. Finally, adopting all three methods can create 1,000 (25 × 4 × 10) intra-category instances.

The basic FractalDB consists of 1,000 or 10,000 different fractal categories and 1,000 instances. In experiments, the ResNet-50 as CNN model pre-trained with FractalDB partially outperformed models pre-trained with human-annotated datasets such as ImageNet and Places.

4.3. FractalDB for Vision Transformers

We introduce two modifications to FractalDB pre-trained models according to the architecture specifications: (i) Colored fractal images and (ii) Training epoch. We investigate whether the performance rate of FractalDB pre-trained ViT improves or not with these configurations by following the success of self-supervised learning with natural images. We believe that it is necessary to utilize colored images for pre-trainig to recognize natural images in a longer training time.

Colored fractal image. Images of conventional FractalDB were drawn by moving dots or patches in grayscale. However, the natural images for common pre-training are not only grayscale, but also various color combinations. The model pre-trained with the datasets constructed natural images has representations related to color distribution in nature [38]. Therefore, we generated the FractalDB in color. The generating procedure was to draw points or patches colored randomly each iteration time. By pre-training with a dataset of colored fractal images, the model acquires feature representations related to color.

Training epoch. The recent Self-Supervised Learning (SSL) methods consider a longer training. For example, SimCLR tried a longer training epochs up to 1k [epoch] [4]. Therefore, although the first work in FDSL [17] conducted with only 90 [epoch], we further verify a suitable training term. Therefore, we also plan to implement a longer training epoch with reference to the recent SSL methods. Here, in the experimental section, we evaluate up to 300 epochs for a further improvement in the FractalDB pre-trained ViT.

4.4. Explore parameters

In FractalDB, the parameters related to the configuration of the dataset and image generation methods were experimentally investigated. Kataoka *et al.* [17] explored #category and #instance, patch vs. point, filling rate, weight of intra-category fractals, #dot, and image size. Here, we only investigate effective parameters for exploration study in ViT architecture. According to their study, we further carry out the experiments in terms of #category/#instance (see Figure 3), 1k/10k categories (see Table 2), patch vs. point (see Table 3), in addition to above-mentioned grayscale vs. color (see Table 4) and training epochs (see Table 5). At the same time, we first compare FractalDB pre-training with other FDSL frameworks and training from scratch (see Table 1) and evaluate patch size which is one of the important parameters in ViT (see Table 6).

Table 1. Comparisons of ViT pre-training among FractalDB and other formula-driven image datasets with Bezier curves (BezierCurveDB) and Perlin noise (PerlinNoiseDB).

	C10	C100	Cars	Flowers
Scratch	78.3	57.7	11.6	77.1
PerlinNoiseDB	94.5	77.8	62.3	96.1
BezierCurveDB	96.7	80.3	82.8	**98.5**
FractalDB-1k	**96.8**	**81.6**	**86.0**	98.3

Table 2. Exploration of larger categories on FractalDB. We compare FractalDB-1k pre-training with FractalDB-10k pre-training.

Pre-train #cat	C10	C100	Cars	Flowers
1k	96.8	81.7	86.0	98.3
10k	**97.6**	**83.5**	**87.7**	**98.8**

5. Experiments

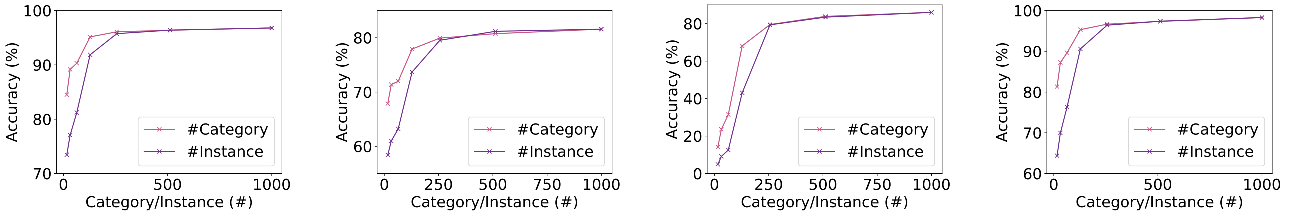
We verify the effectiveness of FractalDB pre-trained ViT in multiple respects. First, we explore a better configuration of FractalDB for ViT. Then we evaluate the best configuration in FractalDB pre-trained ViT on several image datasets, namely CIFAR-10/100 (C10/C100), Stanford Cars (Cars), and Flowers-102 (Flowers), by following the paper [20]. Moreover, we quantitatively compare the FractalDB pre-trained ViT with the pre-training with representative large-scale image datasets (e.g., ImageNet-1k, Places-365) and architectures (e.g., ResNet-50).

Here, to confirm the properties of the FractalDB pre-trained model, we simply use the original ViT model (more specifically, we assign DeiT; hereafter, we assign DeiT for the experiments in the paper) without any modification. We investigate the pre-training method with various parameters. For example, we explore different learning rates in the pre-training phase since DeiT is known to be an architecture parameter-sensitive to different training datasets. The fine-tuning setting is the same as that of [20].

5.1. Exploration study

We explore an effective FractalDB configuration for DeiT under the reference in Kataoka *et al.* [17]. According to the previous work, a full exploration would be very time-consuming. Therefore, we seek to implement the most influential parameters which are described in Section 4.4.

Comparison with other formula-driven image datasets (see Table 1). In addition to FractalDB, Kataoka et al. [17] also proposed datasets based on Perlin noise (PerlinNoiseDB) and Bezier curves (BezierCurveDB). We try to pre-train and fine-tune with these formula-driven image datasets on the DeiT architecture. This allows us to determine whether the proposed FractalDB performs the best through pre-training. From Table 1, we can confirm the existence of higher accuracies compared to scratch training with all of the formula-driven image datasets. The improve-



(a) C10 (b) C100 (c) Cars (d) Flowers

Figure 3. Effects of #category and #instance. The other parameter is fixed at 1,000, e.g., #Category is fixed at 1,000 as #Instance varies among {16, 32, 64, 128, 256, 512, 1,000}.

Table 3. Patch vs. point.

	C10	C100	Cars	Flowers
Point	94.2	77.3	65.4	95.1
Patch	**96.8**	**81.6**	**86.0**	**98.3**

Table 4. Grayscale vs. color.

	C10	C100	Cars	Flowers
Grayscale	**97.1**	**82.6**	**87.1**	**98.3**
Color	96.8	81.6	86.0	**98.3**

Table 5. Training epoch.

#Epoch	C10	C100	Cars	Flowers
100	96.1	81.1	82.0	96.5
200	**96.8**	**82.1**	85.3	98.2
300	**96.8**	81.6	**86.0**	**98.3**

Table 6. Patch size.

	Size	C10	C100	Cars	Flowers
ImageNet-1k	16	**98.0**	**85.5**	**89.9**	**99.4**
	32	97.5	84.7	86.4	98.0
FractalDB-1k	16	**96.8**	**81.6**	**86.0**	**98.3**
	32	95.5	78.4	76.0	95.7

ments are up to {+18.5, +23.9, +74.4, +21.2} higher accuracies with FractalDB-1k on {C10, C100, Cars, Flowers} datasets. Note that the configuration is based on the original and standard FractalDB-1k which contains 1,000 [category] × 1,000 [instance]. In formula-driven image datasets, the FractalDB pre-trained DeiT outperforms the other pre-trained models. The accuracies are {+0.1, +1.3, +3.2, -0.2} from BezierCurveDB pre-trained DeiT. According to this result, we conduct the following experiments by FractalDB.

#category and #instance (see Figures 3(a), 3(b), 3(c), 3(d)). Figure 3 indicates the effects of increase for category and instance on FractalDB pre-training. We set category and instance as variables, fixing one of them at 1000 and changing the others to {16, 32, 64, 128, 256, 512, 1000}. From the experimental results, a larger category and instance tend to lead to higher accuracy on a fine-tuning dataset. Especially in FractalDB pre-training, the category increase is more effective for transfer learning on an image dataset. This result is intuitive because the task is easier for datasets with fewer categories and more instances.

Hereafter, we use 1,000 [category] × 1,000 [instance] as a basic setting of FractalDB. Due to the effectiveness of increasing the number of categories for improving the accuracy, we try to optimize 10,000 [category] × 1,000 [instance] as well. It is said to be better when the transformer's pre-training dataset is larger. We'd like to confirm whether the tendency is applicable in image classification tasks.

Larger categories (see Table 2). We conduct an experiment in larger categories on FractalDB. The table indicates a larger FractalDB pre-training enhances the transformer in image classification. As a matter of fact, the accuracies are improved as {96.8, 97.6} by FractalDB-{1k, 10k} pre-training on CIFAR-10.

Patch vs. point (see Table 3). Table 3 indicates a comparison between 3 × 3 [pixel] patch rendering and 1 × 1 [pixel] point rendering. We execute the experiment to find a better way of fractal rendering. Though the point rendering represents a detailed pattern in a fractal image, the patch rendering augments the instances inside of the category. We can confirm that patch rendering increases performance with {+2.6, +4.7, +20.6, +3.2} on {C10, C100, Cars, Flowers}. We assign the patch rendering in FractalDB through the experiments.

Grayscale vs. color (see Table 4). The table shows pre-training on FractalDB works better with grayscale images than colored images. Especially, in C100 and Cars datasets, the improvement gaps are +1.0 pt and +1.1 from the pre-training with colored fractal images. We confirmed that the colored representation is not required in DeiT architecture.

Training epoch (see Table 5). In FractalDB pre-training, a longer training epoch tends to achieve better performance rates, similarly to SSL methods. The accuracies in 300 epoch pre-training recorded the best scores in three out of four different datasets.

Patch size in DeiT (see Table 6). To input an image to DeiT, an image is divided with multiple patches. Though the patch size in DeiT is verified, we also seek a suitable patch size by comparing between ImageNet-1k and

Table 7. Comparison of pre-training DeiT-Ti on several datasets. The optimization setting is based on [34]. We show types of pre-trained image (PT img), which includes {Natural Image (Natural), Formula-driven image dataset (Formula)}; and pre-training type (PT Type), which includes {Supervised learning (Supervision), Formula-driven supervised learning (Formula-supervision)}. We employed CIFAR-10 (C10), CIFAR-100 (C100), Stanford Cars (Cars), and Flowers-102 (Flowers), Pascal VOC 2012 (VOC12), Places-30 (P30), ImageNet-100 (IN100) datasets. The **Underlined bold** and **bold** scores show the best and second best values, respectively.

PT	PT Img	PT Type	C10	C100	Cars	Flowers	VOC12	P30	IN100
Scratch	–	–	78.3	57.7	11.6	77.1	64.8	75.7	73.2
Places-30	Natural	Supervision	95.2	78.5	69.4	96.7	77.6	–	86.5
Places-365	Natural	Supervision	**97.6**	**83.9**	**89.2**	**99.3**	84.6	–	**89.4**
ImageNet-100	Natural	Supervision	94.7	77.8	67.4	97.2	78.8	78.1	–
ImageNet-1k	Natural	Supervision	**98.0**	**85.5**	**89.9**	**99.4**	**88.7**	**80.0**	–
FractalDB-1k	Formula	Formula-supervision	96.8	81.6	86.0	98.3	84.5	78.0	87.3
FractalDB-10k	Formula	Formula-supervision	**97.6**	83.5	87.7	98.8	**86.9**	**78.5**	**88.1**

FractalDB-1k pre-training. As shown in Table 6, we calculated DeiT with different patch sizes, {16×16, 32×32} [pixel], at each pre-training setting. From the table, the 16×16 patch size in both ImageNet-1k and FractalDB-1k is the better configuration in three out of four datasets.

5.2. Comparisons

We compare the performance of FractalDB pre-trained DeiT with {ImageNet-1k, ImageNet-100, Places-365, Places-30} pre-trained DeiT on representative datasets in addition to training from scrach with additional fine-tuning datasets. ImageNet-100 and Places-30 are randomly selected categories from ImageNet-1k and Places-365 presented in [17]. Moreover, we also evaluate SSL methods with {Jigsaw, Rotation, MoCov2, SimCLRv2} on ImageNet-1k. Here, we show the effectiveness of the proposed method in compared properties, namely human supervision with natural images (Table 7) and self supervision with natural images (Table 8).

FDSL vs. Supervised Learning. We compare natural image datasets and the FractalDB in the pre-training phase. Table 7 describes the detailed settings in pre-training (PT), architecture (Arch.), and pre-training images (PT img) and their performance in terms of accuracy. At the beginning, the FractalDB-1k/10k pre-trained DeiTs outperformed the pre-trained models on 100k-order labeled datasets (ImageNet-100 and Places-30). Although the FractalDB-10k pre-trained DeiT did not exceed the performance with million-order labeled datasets (ImageNet-1k and Places-365), the scores are similar to the ImageNet-1k pre-trained model.

FDSL vs. SSL. Through the comparisons with SimCLRv2, we clarify that the FractalDB-10k pre-trained DeiT performs slightly higher (FractalDB-10k 88.8 vs. SimCLRv2 88.5) in average accuracy on representative datasets. The FractalDB-10k pre-training outperformed the SimCLRv2 pre-training on C10 (97.6 vs. 97.4), Cars (87.7 vs. 84.9), and VOC12 (86.9 vs. 86.2); the accuracy was lower on C100 (83.5 vs. 84.1), Flowers (98.8 vs. 98.9), and P30 (78.5 vs. 80.0). In addition to SimCLRv2, we implemented Jigsaw, Rotation, MoCov2 to compare with our FractalDB-10k. Although the FractalDB-10k pre-trained model performs similarly to SimCLRv2, the proposed method recorded higher accuracies than other SSL methods including MoCov2, Rotation, and Jigsaw, except for P30 dataset. Further comparisons with other SSL methods are shown in Table 8.

5.3. Additional experiments

We conduct additional experiments in DeiT vs. ResNet (see Table 9) by using more parameters. We also show the visualization through first linear embeddings, positional embeddings, mean attention distance (Figure 2), and attention map (Figure 4).

DeiT vs. ResNet (see Table 9). We additionally verify DeiT and ResNet with different architecture sizes. We tested ResNet-{18, 34, 50} and DeiT-{Ti, B} with 16×16 patch. We assigned data augmentation in conjunction with the DeiT's setting. The performance rates of ResNets and DeiTs are listed in Table 9. At the beginning, different from the paper [17], the accuracies of ResNet-50 are better than previous ones (e.g., from 94.1 to 96.1 on C10). However, the FractalDB pre-trained DeiTs are still better than FractalDB pre-trained ResNets on fine-tuning datasets.

Visualization (see Figure 2 and 4). For DeiT, the filters of the first linear embedding, similarity of positional embedding, and mean attention distance can be visualized by following the previous work [10]. We list the filters as representations of ImageNet-1k and FractalDB-1k pre-trained models. Figure 2(a) shows trained filters with ImageNet-1k and FractalDB-1k. Though both DeiT pre-trained on ImageNet-1K and FractalDB-1K acquire similar filters, the FractalDB-1k pre-trained DeiT tends to spread in wide-ranged areas of these filters. On one hand, the filters of ImageNet-1k pre-trained DeiT seem to concentrate on center areas. Figure 2(b) illustrates the cosine similarity of positional embedding corresponding to the input patch at each row and column. From the visualized figures, the

Table 8. Detailed results with FDSL (FractalDB-10k) vs. SSL (Jigsaw, Rotation, MoCov2, SimCLRv2). Addition to columns also in Table 6, 'Use Natural Images?' shows whether the natural images was used or not in the pre-training phase. 'Average' indicates the average accuracy of all datasets in the table. ImageNet-100 is eliminated from the table because the listed SSL methods are trained by images on ImageNet-1k. The **Underlined bold** and **bold** scores show the best and second best values, respectively.

Method	Use Natural Images?	C10	C100	Cars	Flowers	VOC12	P30	Average
Jigsaw	YES	96.4	82.3	55.7	98.2	82.1	**80.6**	82.5
Rotation	YES	95.8	81.2	70.0	96.8	81.1	79.8	84.1
MoCov2	YES	96.9	83.2	78.0	98.5	85.3	**80.8**	87.1
SimCLRv2	YES	**97.4**	**84.1**	**84.9**	**98.9**	**86.2**	80.0	**88.5**
FractalDB-10k	NO	**97.6**	**83.5**	**87.7**	**98.8**	**86.9**	78.5	**88.8**

Table 9. DeiT vs. ResNet with FractalDB-1k pre-training.

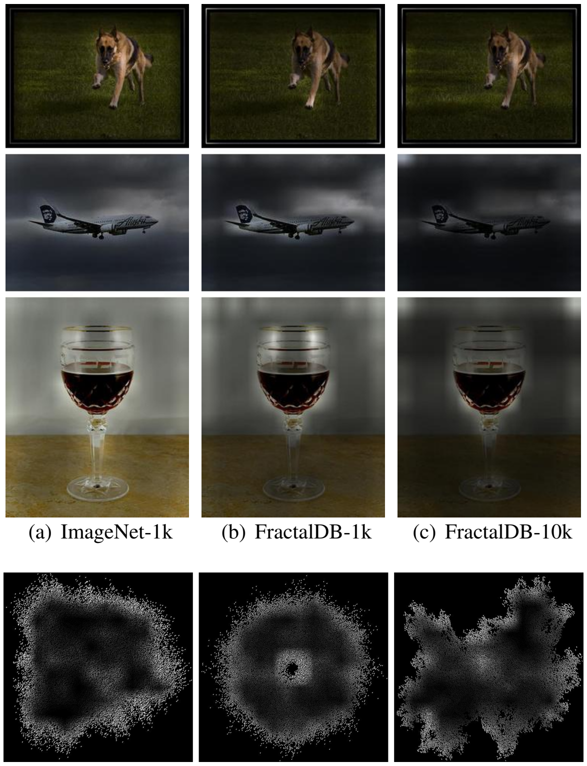
Arch.	Params (M)	C10	C100	Cars	Flowers
ResNet-18	11	94.8	77.6	65.2	96.3
ResNet-34	21	95.9	79.4	79.8	84.9
ResNet-50	25	96.1	80.0	82.5	98.2
DeiT-Ti/16	5	96.8	81.6	86.0	98.3
DeiT-B/16	86	97.1	83.2	86.5	97.9

FractalDB-1k pre-trained DeiT acquired similar position embeddings at each row and column to ImageNet-1k pre-trained DeiT. These pre-training datasets, ImageNet-1k and FractalDB-1k allow us to grab a feature from the same image position. Figure 2(c) shows mean attention distance as in the original DeiT [10]. By comparing to the ImageNet-1k pre-training, the FractalDB-1k pre-trained DeiT tends to look at wide-spread areas in an image. The indicator is similar to the size of receptive fields in CNN.

Figure 4 illustrates attention maps in DeiT with different pre-training datasets. The FractalDB-1k pre-trained ViT focuses on the object areas (Figure 4(b)) as well as ImageNet pre-training (Figure 4(a)). Moreover, the FractalDB-10k pre-trained DeiT looks at more specific areas (Figure 4(c)) compared to FractalDB-1k pre-training. Figure 4(d) shows attention maps in fractal images. From the figures, the FractalDB pre-training seems to recognized by observing contour lines. In relation to Figure 2(c), we believe that the recognition of complex and distant contour lines enabled the extraction of features from a wide area.

6. Conclusion and discussion

We successfully trained Vision Transformers (ViT) without any natural images and human-annotated labels through the framework of Formula-Driven Supervised Learning (FDSL). Our FractalDB pre-trained ViT achieved similar performance rates to the human-annotated ImageNet pre-trained model, partially outperformed SimCLRv2 self-supervised ImageNet pre-trained model, and surpassed other self-supervised pre-training methods, including MoCov2. According to the results of the experiments, the findings are as follows.



(a) ImageNet-1k (b) FractalDB-1k (c) FractalDB-10k

(d) Attention maps in fractal images with FractalDB-1k pre-trained DeiT. The brighter areas show more attentive areas.

Figure 4. Attention maps.

Feature representation with FractalDB pre-trained ViT. From the visualization results, the FractalDB pre-trained ViT acquired different the feature representations in the first linear embeddings (Figure 2(a)), and similar arranged position embeddings (Figure 2(b)) compared to the ImageNet-1k pretrained model. Moreover, Figure 4(d) illustrates that ViT tends to pay attention to the contour areas in pre-training phase. We believe that the pre-trained model enabled feature acquisition in an area covering a wider range than the ImageNet-1k pre-trained model (Figure 2(c)). We also understood the complex contour lines used to classify fractal categories in the pre-training phase.

Can we complete pre-training of ViT without natural images and human-annotated labels? According to the comparisons with SSL methods (Table 8), we showed that the performance of FractalDB-10k was comparative to the accuracy of SimCLRv2 pre-trained ViT, which is trained by 1.28M natural images on ImageNet. Although 10M images are used in FractalDB-10k, natural images are not used at all in the pre-training phase. Therefore, we can use a FDSL-based pre-training dataset to safely train ViT in terms of AI ethics and image copyright, if we can exceed the accuracy of supervised learning with human annotation (Table 7).

Acknowledgement

- This paper is based on results obtained from a project subsidized by the New Energy and Industrial Technology Development Organization (NEDO).
- Computational resource of AI Bridging Cloud Infrastructure (ABCI) provided by National Institute of Advanced Industrial Science and Technology (AIST) was used.
- We want to thank Yoshihiro Fukuhara, Munetaka Minoguchi, Eisuke Yamagata, Ryota Suzuki, Yutaka Satoh, Yue Qiu, and Shintaro Yamamoto for their helpful comments during research discussions.

References

[1] M. F. Barnsley. Fractals Everywhere. *Academic Press. New York*, 1988.

[2] T. B. Brown and et al. Language Models are Few-Shot Learners. In *Neural Information Processing Systems (NeurIPS)*, 2020.

[3] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin. Unsupervised Learning of Visual Features by Contrasting Cluster Assignments. In *Neural Information Processing Systems (NeurIPS)*, 2020.

[4] 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.

[5] T. Chen, S. Kornblith, K. Swersky, M. Norouzi, and G. Hinton. Big self-supervised models are strong semi-supervised learners. In *Neural Information Processing Systems (NeurIPS)*, 2020.

[6] X. Chen, H. Fan, R. Girshick, and K. He. Improved baselines with momentum contrastive learning. In *CoRR:2003.04297*, 2020.

[7] 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.

[8] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *North American Chapter of the Association for Computational Linguistics (NAACL)*, 2019.

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

[10] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representation (ICLR)*, 2021.

[11] P. Foret, A. Kleiner, H. Mobahi, and B. Neyshabur. Sharpness-aware Minimization for Efficiently Improving Generalization. In *International Conference on Learning Representation (ICLR)*, 2021.

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

[13] 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.

[14] K. He, R. Girshick, and P. Dollár. Rethinking ImageNet Pre-training. In *The IEEE International Conference on Computer Vision (ICCV)*, 2019.

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

[16] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger. Densely Connected Convolutional Networks. In *The IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[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] A. Kolesnikov, L. Beyer, X. Zhai, J. Puigcerver, J. Yung, S. Gelly, and N. Houlsby. Big Transfer (BiT): General Visual Representation Learning. In *European Conference on Computer Vision (ECCV)*, 2020.

[19] 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*. 2012.

[20] J. Laplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei. Scaling Laws for Neural Language Models. In *CoRR:2001.08361*, 2020.

[21] D. Mahajan, R. Girshick, V. Ranathan, K. He, M. Paluri, Y. Li, A. Bharambe, and L. van der Maaten. Exploring the Limits of Weakly Supervised Pretraining. In *European Conference on Computer Vision (ECCV)*, 2018.

[22] B. Mandelbrot. The fractal geometry of nature. *American Journal of Physics*, 51(3), 1983.

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

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

[25] 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.

[26] H. Pham, Z. Dai, Q. Xie, M.-T. Luong, and Q. V. Le. Meta Pseudo Labels. In *CoRR:2003.10580*, 2020.

[27] A. Radford, K. Narasimhan, T Salimans, and I. Sutskever. Improving language understanding by generative pre-training. In *Technical Report, OpenAI*, 2018.

[28] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. Language Models are Unsupervised Multitask Learners. In *International Conference on Machine Learning (ICML)*, 2018.

[29] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision (IJCV)*, 115(3), 2015.

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

[31] 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.

[32] M. Tan and Q. V. Le. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In *International Conference on Machine Learning (ICML)*, 2019.

[33] A. Torralba, R. Fergus, and W. T. Freeman. 80 million tiny images: A large data set for nonparametric object and scene recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, 2008.

[34] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou. Training data-efficient image transformers & distillation through attention. In *CoRR:2012.12877*, 2020.

[35] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, 2017.

[36] 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.

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

[38] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision (ECCV)*. Springer, 2014.

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

[40] 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)*, 40, 2017.