Exploring the Potential of Neural Dataset Search

Ryosuke Yamada[1,2]*, Risa Shinoda[1,3] *, Hirokatsu Kataoka[1]
[1]National Institute of Advanced Industrial Science and Technology,
[2]University of Tsukuba, [3]Kyoto University

Abstract

*Although we have witnessed Neural Architecture Search (NAS), which automatically explores architecture for best performance, the discussion has not advanced considering a dataset. We discuss the potential of **Neural Dataset Search (NDS)**, which explores the appropriate configuration in a pre-training dataset to achieve a better pre-training effect. The NDS is designed to train in order to find the optimal parameters in the pre-training dataset for a given network architecture and downstream tasks. This allows for predicting the optimal pre-training parameters for a new unseen task in one shot. Thus, the NDS has the potential to bottom up the effectiveness of the pre-training. Therefore, this paper focuses on formula-driven supervised learning, and as a first consideration, we verify the appropriate configuration in Residual Network (ResNet) and Fractal DataBase (FractalDB). From the experimental results, we confirmed that the FractalDB generation parameters that provide the best pre-training effect are different for each ResNet-{18, 50, 152}. These observations reveal that there is an adapted image representation or dataset structure (e.g., input size, parameter, category) for a particular architecture. We hope these results will encourage further research on NDS that fully exploits the pre-training of synthetic images.*

1. Introduction

How to construct deep neural networks (DNNs) is a crucial issue in computer vision. With reference to a hierarchical structure of the visual cortex and the extrastriate cortex, neural networks have been proposed. Initially, it was difficult to implement more than two hidden layers due to local optimization and gradient vanishing problems. However, starting with AlexNet [17] in 2012, many DNNs with hundreds of hidden layers have been proposed [12–14, 17, 26, 29, 32]. In particular, the Residual Network (ResNet) [12] has made it possible to efficiently deepen layers by residual blocks and achieve high performance for image recognition.

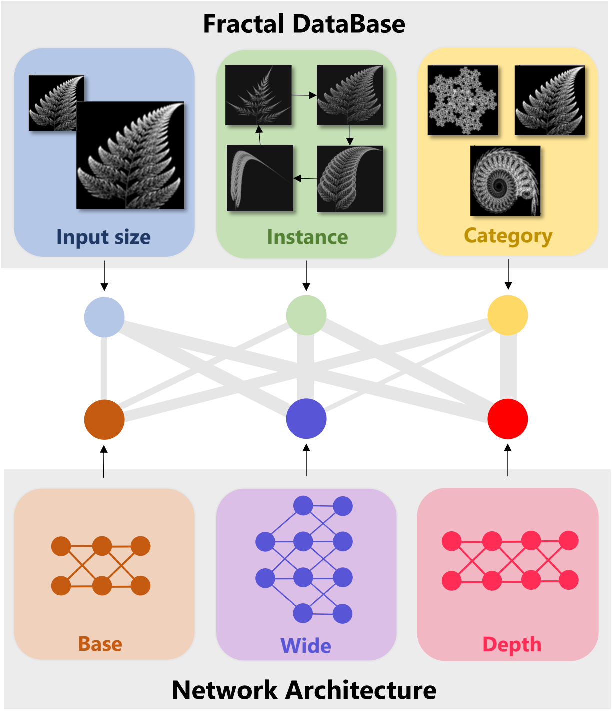

Figure 1. We proposed neural dataset search (NDS). In the pre-training phase, NDS can contribute to the higher pre-training effect, combined with neural architecture search.

In recent years, neural architecture search (NAS) [39], the automatic generation of optimal network architecture configurations for highly accurate recognition, has been the focus of much attention. The current NAS research focuses primarily on improving search algorithms, designing the search space, reducing search costs, and integrating direct indicators into the search process [19, 24, 30]. NAS allows for generating a neural architecture that maximizes the expected accuracy. This work goes one step further and proposes a new concept, neural dataset search (NDS), which not only achieves NAS from automatically generated data based on a formulation but also generates an optimal pre-

*indicates equal contribution.

training dataset by including even the generation rules as search spaces.

Pre-training is the standard technique to achieve better results with a limited dataset. Many SOTA models use pre-training models by huge datasets such as ImageNet [5] or JFT-300M [28]. However, such large-scale datasets have been reported for privacy and ethical issues [33, 34, 37]. For another option for natural images, synthetic datasets have been created [20, 25]. Although the standard way of creating a synthetic dataset is using a simulation environment, formula-driven supervised learning (FDSL) [15] is known for its operability and flexibility. FDSL simultaneously and automatically generates image patterns and paired labels using mathematical formulas. With FDSL, the dataset can be constructed without manual labor, allowing for free manipulation of dataset components, such as the number of instances per category. However, the best parameters for the generation rules and parameters are manually and empirically searched for in the current FDSL.

Therefore, in this paper, we discuss the potential of NDS, which investigates the pre-training image dataset configuration jointly with model architectures. We extend the FDSL and NAS framework, which automatically searches for optimal data rules and parameters for each downstream task and even network configuration (see Figure 1). If the pre-training dataset configuration contributes to better results depending on the model architecture, there is space for improvements of the pre-training effect by varying pre-training dataset parameters. Pre-training datasets for each model architecture have yet to be fully investigated. Therefore, we investigate whether the pre-training dataset also affects results depending on dataset architecture.

To vary the pre-training dataset configuration, we use FDSL [15]. This research aim is to determine whether the pre-training dataset configuration and model architecture affect the results. If we can confirm there is further room for improvements by jointly considering the dataset configuration and model architecture, neural dataset search has the potential to achieve better pre-training. Specifically, we employ ResNet as the network architecture for this experiment and Fractal DataBase (FractalDB) as the pre-training dataset. Although it is desirable to ensure diversity on both the architecture and dataset sides in the verification process, we have limited ourselves to verifying ResNet and FractalDB due to computational cost. However, the experimental results are interesting enough to discuss the possibility of NDS.

Our main contributions are as follows; (i) We discuss the potentials of NDS, which explores the appropriate configuration in architecture and pre-training datasets. (ii) We confirmed that some model architectures achieved the best performance at the different dataset parameters. NDS has the potential to achieve better pre-training.

2. Related work

NAS. Choosing the best model architecture and training settings is a difficult problem. AutoML (Automated Machine Learning) is one way to achieve efficient training. AutoML is used to find better loss [18], augmentation process [3], and hyper parameters [4]. For the architecture, the neural architecture search (NAS) [19, 24, 30, 39] has been developed, which generates architectures that maximize the expected accuracy. NAS is usually applied to fine-tuning phase, and the number of pre-training phase research is limited.

Dataset Search. There is some prior research to gain a better representation of the dataset. Dataset distillation [27, 31] is one way to train with an efficient dataset size by compressing a dataset into small synthetic data. While dataset distillation creates another synthetic data, dataset pruning [35] removes redundant training datasets with a minor impact on the model's performance. These prior researches contribute to efficient training with a smaller dataset. The concept of our proposed natural dataset search differs from theirs in that we create an efficient dataset with synthetic data, not extracting from an existing dataset. In the context of the synthetic pre-training dataset, Task2Sim [21] is the model which searches for optimal parameters for the generation of the synthetic dataset. We further verify dataset parameters should be investigated jointly with architecture.

Image dataset and training framework. Undoubtedly, transfer learning with large-scale datasets has contributed to accelerating visual training [11]. Initially, the ImageNet [6] and Places [38] pre-trained models were widely used for diverse tasks. However, even in million-scale datasets, several concerns exist, such as AI ethics and copyright problems, e.g., fairness protection, privacy violations, and offensive labels. We must pay attention to the terms of use in large-scale image datasets and create pre-trained models accordingly.

On the one hand, to alleviate the image labeling labor required of human annotators, Self-Supervised Learning (SSL) progressed significantly in recent years [7, 10, 22, 23, 36]. The SSL methods are closer to supervised learning with human annotations regarding performance rates. In this context, formula-driven supervised learning (FDSL) [15] was proposed to overcome the problems of AI ethics and copyrights [33, 34, 37], and annotation labor. The framework is similar to self-supervised learning. However, FDSL methods do not require any natural images taken by a camera or simulation environment to create synthetic images. The framework simultaneously and automatically generates image patterns and paired labels for pre-training image representations using simple mathematical formulas. We use FDSL to vary dataset configurations since FDSL allows for free manipulation of dataset components, such as the number of instances per category.

3. Methodology

We present the method of fractal image generation, which is the source of FractalDB, a pre-training dataset for NDS. First, we explain the process of generating fractal images using a recursive function called the Iterated Function System (IFS). We also explain how FractalDB is constructed (see Section 3.1). After that, we present the pre-training strategy of FractalDB (see Section 3.2).

3.1. FractalDB

In this study, we use FractalDB, which is a representative dataset in FDSL. This is because FractalDB is simpler than other FDSL methods and has fewer search parameters. This makes it ideal for the initial study of NDS, where computational resources are an issue. FractalDB is constructed by the iterated function system IFS, which represents fractal geometry. An IFS generate A point set of fractal X = $\{x_1$,x_2,...,$x_K\}$ constituting where K is the number of points. Note that the fractal is obtained when $K \to \infty$, but we assume K is a finite number for computational efficiency. An IFS is defined by a set of transformations $w_i : \mathcal{X} \to \mathcal{X}$ and their corresponding probabilities p_i in a complete metric space $\mathcal{X}$. An IFS Θ is denoted by

$$\Theta = \{\mathcal{X}; w_1, w_2, \cdots, w_N; p_1, p_2, \cdots, p_N\}, \tag{1}$$

here, N is the number of pairs (w, p) to be considered. The transformation w_i is defined by an affine transformation. The transformation w during fractal image generation on a two-dimensional Euclidean plane is given by

$$w_i(x) = \begin{bmatrix} a_i & b_i \\ c_i & d_i \end{bmatrix} x + \begin{bmatrix} e_i \\ f_i \end{bmatrix}. \tag{2}$$

The probability p_i is set to

$$p_i = \frac{|\det A_i|}{\sum_{i=1}^{N} |\det A_i|}, \tag{3}$$

where $detA_i = a_i d_i - b_i c_i$. Fractal images are generated by recursively calculating the corresponding transformation w according to the N types of probabilities p_i using w_i.

i. Parameters of affine transform $\{a_i, b_i, c_i, d_i, e_i f_i\}$ are randomly sampled from the uniform distribution over $[-1.0, 1.0]$. The probabilities p_i are determined by equation 3.

ii. The number of pairs $(w_i$,$p_i)$ N is randomly determined from a discrete uniform distribution over $[2, 3..., 8]$. Using method (i), N sets of parameters are prepared and the IFS is determined.

iii. The affine transformation (Equation 2) is applied to coordinate x_{t-1} according to probability p_i, and the new coordinate x_t is obtained.

iv. By repeating (iii) K times, the point set X=$\{x_1$, x_2,...,$x_K\}$ is generated.

Category definition. FractalDB defines categories based on rendered fractal regions. This time, fractal images whose fractal region occupies 20% or more of the whole image are defined as a category, according to the original FractalDB. For example, repeat steps (i) (iv) until the number of categories reaches 1,000 to construct a FractalDB-1k with 1,000 categories.

Instance augmentation. FractalDB provides instance augmentation within a category by IFS perturbing, rotating, and randomly drawing for fractal images in each category. In particular, the perturbation is larger, the rendered image representation differs significantly from the original fractal image. For each original fractal image, FractalDB-1k combines 25 patterns of IFS perturbation, 4 patterns of rotation, and 10 patterns of random patch drawing. Thereby, 1,000 instances of fractal images are generated in each category.

IFS perturbation refers to the process of applying perturbations to the IFS parameters determined as a category, then generating fractal images using the perturbed parameters. Rotation is an operation that rotates the point cloud and is equivalent to rotating the fractal image. When generating intra-class images, a 90-degree rotation is applied four times. Random patch rendering involves rendering the obtained point cloud on the image, not as points, but as patches with random values. In FractalDB-1k, 10 random patch patterns were used to generate images with 10 different random patch patterns.

3.2. FractalDB Pre-training

This section describes the FractalDB pre-training method. FractalDB even generates supervised labels corresponding to fractal images, as explained in section 3.1. Therefore, FractalDB can achieve supervised pre-training by supervised labeled dataset $D = \{(x_i, y_i)\}_{i=1}^{N}$. For FractalDB, the cross-entropy loss is used, which is given by

$$\mathcal{L}_{\text{ce}}(\theta; D) = -\frac{1}{N} \sum_{i=1}^{N} \sum_{c=1}^{C} t_{i,c} \log y_{i,c}, \tag{4}$$

where $y_i = f_\theta(x_i) \in \mathbb{R}^C$ is the output vector of a learnable network f_θ, such as a ResNet, θ is a set of parameters, and C is the number of categories. The details of the pre-training conditions in this experiment are presented in Section 4. Typically, the number of images N should be equal to or more than one million in order to achieve good pre-training performance.

Table 1. Pre-training effects of ImageNet100 (IN100) and ImageNet1k (IN1k) on ResNet-{18, 50, 152} Note that this is not a comparison. Please see the tendency between network and accuracy.

Architecture	Pre-train	Caltech	A40	F101	VOC07
ResNet-18	–	41.32	30.01	70.23	62.30
	IN100	74.99	52.55	70.86	74.01
	IN1k	**91.87**	**76.23**	**80.07**	**86.14**
ResNet-50	–	30.76	21.95	66.31	56.45
	IN100	71.81	51.23	70.69	73.22
	IN1k	**94.39**	**82.61**	**85.33**	**88.43**
ResNet-152	–	24.83	15.6	58.79	52.72
	IN100	72.21	45.91	67.04	72.67
	IN1k	**95.89**	**84.58**	**86.44**	**90.82**

4. Experimental setting

Throughout the experiments, we would like to verify the joint search for network architecture and image datasets in transfer learning which consists of pre-training and fine-tuning. We introduce these experiments in both network architecture and pre-training and fine-tuning details.

In this experiment, we use ResNet-{18, 50, 152}. ResNet is a widely used CNN in image recognition. Recently, the Vision Transformer has also been validated. However, in this study, we focus on ResNet only from a computational point of view, since the experiments are exploratory on a large pre-training dataset.

For the pre-training, we generate pre-trained models of ImageNet and FractalDB under the following conditions, in accordance with [15]. An optimization method is Stochastic Gradient Descent (SGD), where the weight decay is set to 0.0004, the inertia term is set to 0.9, and the initial learning rate is set to 0.01. The learning rate is multiplied by 0.1 when the number of epochs reaches 30 or 60. The input image is resized to 224 × 224 during learning. Learning in the pre-training is terminated when the number of epochs reaches 90 epochs.

The optimization method and hyperparameters of ResNet-50 during fine-tuning are the same as those used for pre-training. For training, the input image is resized to 256 × 256 and then randomly cropped to 224 × 224. For testing, the input image is resized to 256 × 256 and then center-cropped to 224 × 224. As with the pre-training, fine-tuning is terminated when the number of epochs reaches 90 epochs. We investigated Fine-tuning with Caltech101 [9], Stanford 40 Action Dataset (Action40) [1], Food101 [2], PscalVOC2007 / 2012 [8], and CIFAR10 / 100 [16] in order to evaluate it in various image classification tasks.

Table 2. Pre-training effects of instance augmentation on ResNet.

IFS weights	Caltech	A40	F101	VOC07
small	63.72	30.31	74.57	70.2
base	64.07	26.68	74.5	69.44
large	62.26	28.0	74.25	68.25

Table 3. Pre-training effects of fractal image size on ResNet.

Image size	Caltech	A40	F101	VOC07
256^2	**66.81**	**33.38**	**74.5**	**70.22**
362^2	64.07	26.68	**74.5**	69.44
512^2	63.1	27.78	72.67	67.86

5. Results and Analysis

In this section, we aim to investigate the performance changes in downstream tasks depending on the architecture and the configuration of the pre-training dataset. First, we rethink the correlation between the ImageNet pre-trained model and the architecture size (See Section 5.1). Next, we investigate the correlation between the parameters in terms of the FractalDB generation and the architecture size (See Section 5.2). Last, we verify the correlation between the FractalDB configurations (category and instance) and the architecture size (See Section 5.3).

5.1. ImageNet pre-training on ResNet (Table 1)

In this experiment, we investigate the effect of pre-training on dataset size and architecture size on real images by comparing the ImageNet100 pre-trained model with the ImageNet-1k pre-trained model on ResNet-{18, 50, 152} for image classification. Although pre-training on real images has already been verified in various papers on dataset size and architecture size, this experiment again confirms our baseline and discusses the limitation of pre-training on real images from a different perspective.

Table 1 shows the comparison between scratch training on ImageNet-100 and -1k pre-training with ResNet-18, 50, 152. Unsurprisingly, ImageNet-1k performed the best on all fine-tuning data sets and ResNet-{18, 50, 152}. This is consistent with previous studies showing that classification accuracy on ImageNet and ResNet improves in proportion to the size of the pre-training dataset.

We also focus on the performance of ResNet-{18, 50, 152} under the same pre-training conditions, which is of particular importance in this study. There is a tendency for identification performance to deteriorate with increasing architecture size for the scratch learning and ImageNet100 pre-trained models. This result suggests that unless the size of the pre-training dataset is above a certain level, the pre-training effect cannot be expected as the architecture size increases.

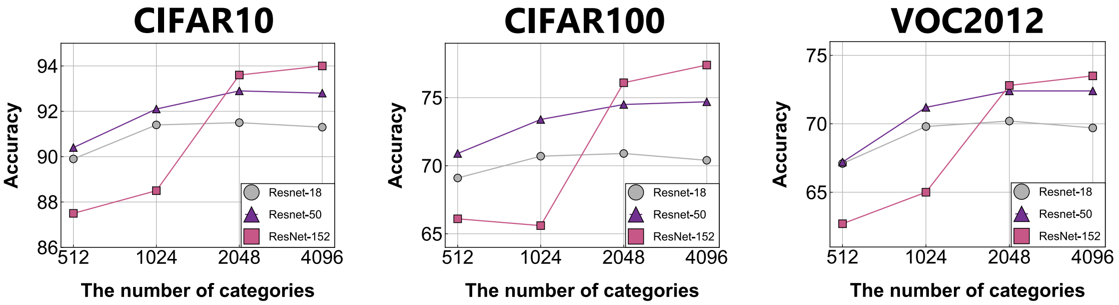

Figure 2. Relationship between ResNet layers and FractalDB **categories**

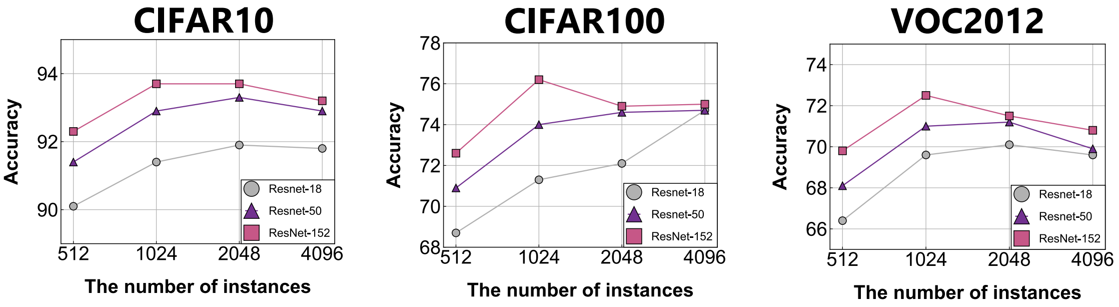

Figure 3. Relationship between ResNet layers and FractalDB **instances**

5.2. Fractal image representation on ResNet (Table 3 and Table 2)

In this experiment, we focus on FractalDB pre-training in ResNet-50 for parameters related to image representation. The parameters can be explored when generating FractalDB. In this paper, we explore two parameters related to image representation: instance dilation and image size, due to computational cost. As described in Section 3, instance expansion is achieved by varying each parameter of IFS, the generation rule. This allows for fine-tuning of the rendered fractal shape.The effect of fractal image diversity within a category on the pre-training effect is investigated. At the same time, the image size of the fractal image rendering can be adjusted. Obviously, the larger the image sizes, the finer the shapes. Therefore, we evaluate the effect of image size on the pre-training effect when rendering.

First, Table 2 shows the experimental results for each IFS parameter variation rate{20%:small, 40%:base, 60%:large} in the instance expansion. Depending on the fine-tuning dataset, Table 2 shows that there is an optimal parameter variation rate. In particular, a performance difference of 3.32% was observed between the IFS weights (base) and the IFS weights (small) for the Food101 dataset.

Next, Table 3 shows the experimental results for the image sizes $\{224^2, 362^2, 512^2\}$. In Table 3, we confirmed that 224^2 fractal images are more effective in pre-training than 362^2 and 512^2 fractal images. In particular, Food101's performance gap between 224^2 and 362^2 was 5.6%.

These results suggest that it is important to search for the optimal image representation parameters in FractalDB since the data representation parameters have a certain impact on the pre-training effect. Therefore, we used two representative parameters as the first study.

5.3. FractalDB configuration and architecture size (Figure 2 and Figure 3)

The relationship between the dataset configuration of FractalDB and the architecture size is investigated in this experiment. For the dataset configuration, we focus on the number of categories and the number of instances. These are considered to have a large impact on the pre-training effect. For both the number of categories and instances, we experiment with four patterns $\{512, 1024, 2048, 4096\}$.

The number of categories in FractalDB and the experimental results on ResNet-$\{18, 50, 152\}$ are shown in Figure 2. For all three fine-tuning datasets, the classification accuracy tends to increase as the number of categories increases. It is interesting to note that the performance of ResNet-152 is lower than that of ResNet-18 and ResNet-52 up to 512 and 1024 categories, but higher than that of ResNet-18 and ResNet-152 at 2048 and 4096 categories.

The number of instances of FractalDB and the experimental results for ResNet-$\{18, 50, 152\}$ are shown in Figure 3. Figure 3 confirms that there is an optimal number of instances for each ResNet. Unlike the number of categories, the classification accuracy does not increase monotonically with the number of instances.

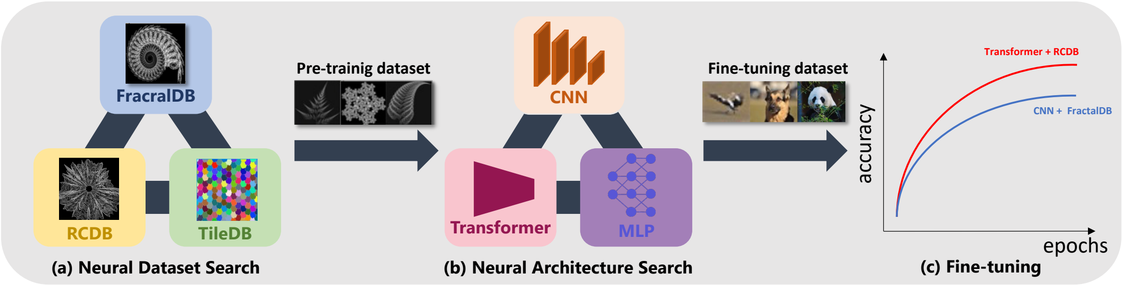

Figure 4. Overview of NDS framework. By performing both NDS and NAS simultaneously, it is possible to construct a pre-trained model that acquires better visual features. Note that the NDS must be a dataset that has some parameters related to image representation. For example, in NDS, the search space is a dataset generated from different rules. In NAS, as in existing studies, the search space is different network architectures. Then, by placing each of them in the search space and finding the optimal combination, the optimal combination of pre-trained datasets and network architectures for various downstream tasks can be efficiently generated.

The above suggests that the expansion of the number of categories is an effective way to expand the pre-training dataset. In addition, pre-training cannot be effective unless a certain amount of data is secured for architectures with a relatively large number of parameters, such as ResNet-152. This is similar to the experiment on the pre-training of ImageNet in Section 5.1.

6. Discussion and Future work

We summarize the main observations from our experiments as follows:

1. In pre-training, regardless of the type of real and synthetic images, the size of the trainable architecture increases as the number of data increases.
2. In ResNet-50, depending on the generation parameters (instance generation method, image size), the pre-training effectiveness of FractalDB varies.
3. As the architecture size increases, the number of categories in FractalDB must be more than a certain number. The improvement is non-linear, especially as the architecture size increases.
4. Depending on the architecture, there is an optimal configuration for the number of categories and instances of FractalDB.

Based on these observations, we provide our answers to a few important questions that may encourage people to rethink the NDS direction.

The limitations of pre-training models with real images are highlighted by the experimental results and the recent trend toward large-scale pre-training models. Recently, the number of architectural parameters has reached tens of billions. This is because it is easier to obtain more generalized feature representations with increasing network size. However, as this experiment shows, increasing the network size requires increasing the amount of training data. Datasets such as JFT-300M, which is larger than ImageNet, are essential for building more effective pre-training models.However, pre-training on real image datasets has its limitations. This is because the cost of curating, privacy, rights of use, and ethical issues associated with real datasets are major obstacles. In fact, JFT-300M is a private dataset. It is not accessible to all researchers. Based on the above considerations, as a possible solution to the above problems with real images, we believe that pre-training using synthetic image datasets is important. In particular, since FDSL can automatically generate datasets from mathematical expressions, researchers can automatically construct pre-training datasets on the fly. There is no need to download data.

Therefore, as a future direction of NAS, we believe that NDS based on synthetic data, such as formula-driven supervised learning, will play an important role in computer vision. This study focused on FractalDB and ResNet as initial studies, but the datasets and network architecture need to be validated in various combinations, as shown in Figure 4. In particular, in NDS, the use of FDSL, which generates a data set based on a set of rules, makes it possible to search for optimal rules for various downstream tasks. The problem is the optimal dataset search in pre-training, which requires dataset search with feedback from fine-tuning results. Although we have not been able to present a concrete methodology for NDS in this study from the viewpoint of computational cost, we expect that the first step in the future will be to automate NDS by utilizing reinforcement learning and existing NAS algorithms. We hope this paper will spark many researchers to explore the possibility of NDS.

Acknowledgement. This paper is based on results obtained from a project, JPNP20006, commissioned by the New Energy and Industrial Technology Development Organization (NEDO). Computational resource of AI Bridging Cloud Infrastructure (ABCI) provided by the National Institute of Advanced Industrial Science and Technology (AIST) was used. We want to thank Kensho Hara and Yue Qiu for their helpful research discussions.

References

[1] A. Khosla A.L. Lin L.J. Guibas B. Yao, X. Jiang and L. Fei-Fei. Human action recognition by learning bases of action attributes and parts. In *The IEEE International Conference on Computer Vision (ICCV)*, page 1331–1338, 2011. 4

[2] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – mining discriminative components with random forests. In *European Conference on Computer Vision (ECCV)*, page 446–461, 2014. 4

[3] Ekin D. Cubuk, Barret Zoph, Dandelion Mané, Vijay Vasudevan, and Quoc V. Le. Autoaugment: Learning augmentation strategies from data. In *The IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 113–123, 2019. 2

[4] Xiaoliang Dai, Alvin Wan, Peizhao Zhang, Bichen Wu, Zijian He, Zhen Wei, Kan Chen, Yuandong Tian, Matthew Yu, Peter Vajda, and Joseph E. Gonzalez. Fbnetv3: Joint architecture-recipe search using predictor pretraining. In *The IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16276–16285, June 2021. 2

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *The IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, 2009. 2

[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)*, pages 248–255, 2009. 2

[7] C. Doersch, A. Gupta, and A. Efros. Unsupervised Visual Representation Learning by Context Prediction. In *The IEEE International Conference on Computer Vision (ICCV)*, pages 1422–1430, 2015. 2

[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*, 111(1):98–136, Jan. 2015. 4

[9] Li Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *The IEEE International Conference on Computer Vision and Pattern Recognition (CVPR) Workshop*, pages 178–178, 2004. 4

[10] S. Gidaris, P. Singh, and N. Komodakis. Unsupervised Representation Learning by Predicting Image Rotations. In *International Conference on Learning Representation (ICLR)*, 2018. 2

[11] K. He, R. Girshick, and P. Dollár. Rethinking ImageNet Pre-training. In *The IEEE International Conference on Computer Vision (ICCV)*, 2019. 2

[12] 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)*, pages 770–778, 2016. 1

[13] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu. Squeeze-and-Excitation Networks . *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 42:2011–2023, 2020. 1

[14] 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)*, pages 4700–4708, 2017. 1

[15] 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. 2, 4

[16] Alex Krizhevsky and Geoffrey Hinton. Learning Multiple Layers of Features from Tiny Images. *University of Toronto*. 4

[17] 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. 2012. 1

[18] Hao Li, Tianwen Fu, Jifeng Dai, Hongsheng Li, Gao Huang, and Xizhou Zhu. Autoloss-zero: Searching loss functions from scratch for generic tasks. In *The IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 999–1008, 2022. 2

[19] Chenxi Liu, Barret Zoph, Maxim Neumann, Jonathon Shlens, Wei Hua, Li-Jia Li, Li Fei-Fei, Alan Yuille, Jonathan Huang, and Kevin Murphy. Progressive neural architecture search. In *European Conference on Computer Vision (ECCV)*, 2018. 1, 2

[20] John McCormac, Ankur Handa, Stefan Leutenegger, and Andrew J. Davison. Scenenet rgb-d: Can 5m synthetic images beat generic imagenet pre-training on indoor segmentation? In *The IEEE International Conference on Computer Vision (ICCV)*, pages 2697–2706, 2017. 2

[21] Samarth Mishra, Rameswar Panda, Cheng Perng Phoo, Chun-Fu Richard Chen, Leonid Karlinsky, Kate Saenko, Venkatesh Saligrama, and Rogerio S. Feris. Task2sim: Towards effective pre-training and transfer from synthetic data. In *The IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9184–9194, 2022. 2

[22] M. Noroozi and P. Favaro. Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles. In *European Conference on Computer Vision (ECCV)*, 2016. 2

[23] 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. 2

[24] Jian Ren, Zhe Li, Jianchao Yang, Ning Xu, Tianbao Yang, and David J. Foran. Eigen: Ecologically-inspired genetic approach for neural network structure searching from scratch. In *The IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9051–9060, 2019. 1, 2

[25] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M. Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *The IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3234–3243, 2016. 2

[26] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *Inter-*

national Conference on Learning Representations (ICLR), 2015. 1

[27] Felipe Petroski Such, Aditya Rawal, Joel Lehman, Kenneth O. Stanley, and Jeff Clune. Generative teaching networks: Accelerating neural architecture search by learning to generate synthetic training data. *ArXiv*, 2019. 2

[28] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 843–852, 2017. 2

[29] 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)*, pages 1–9, 2015. 1

[30] Mingxing Tan, Bo Chen, Ruoming Pang, Vijay Vasudevan, Mark Sandler, Andrew Howard, and Quoc V. Le. Mnasnet: Platform-aware neural architecture search for mobile. In *The IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2815–2823, 2019. 1, 2

[31] Tongzhou Wang, Jun-Yan Zhu, Antonio Torralba, and Alexei A. Efros. Dataset distillation. *arXiv*, 2020. 2

[32] 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)*, pages 1492–1500, 2017. 1

[33] Kaiyu Yang, Klint Qinami, Li Fei-Fei, Jia Deng, and Olga Russakovsky. Towards fairer datasets. In *Conference on Fairness, Accountability, and Transparency*, pages 547–558, jan 2020. 2

[34] Kaiyu Yang, Jacqueline Yau, Li Fei-Fei, Jia Deng, and Olga Russakovsky. A study of face obfuscation in imagenet. In *International Conference on Machine Learning (ICML)*, 2022. 2

[35] Shuo Yang, Zeke Xie, Hanyu Peng, Min Xu, Mingming Sun, and Ping Li. Dataset pruning: Reducing training data by examining generalization influence. *ArXiv*, 2023. 2

[36] Tong Zhang, Congpei Qiu, Wei Ke, Sabine Süsstrunk, and Mathieu Salzmann. Leverage your local and global representations: A new self-supervised learning strategy. In *The IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16580–16589, June 2022. 2

[37] Dora Zhao, Angelina Wang, and Olga Russakovsky. Understanding and evaluating racial biases in image captioning. In *The IEEE International Conference on Computer Vision (ICCV)*, 2021. 2

[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)*, 40, 2017. 2

[39] B. Zoph and Q. V. Le. Neural Architecture Search with Reinforcement Learning. In *International Conference on Learning Representation (ICLR)*, 2017. 1, 2