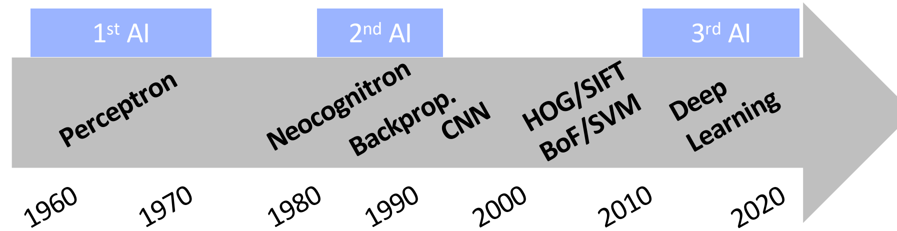
Pre-training without Natural Images

Hirokatsu Kataoka

AIST

What has the DNNs brought?

Problems in DNNs



– Annotation labor

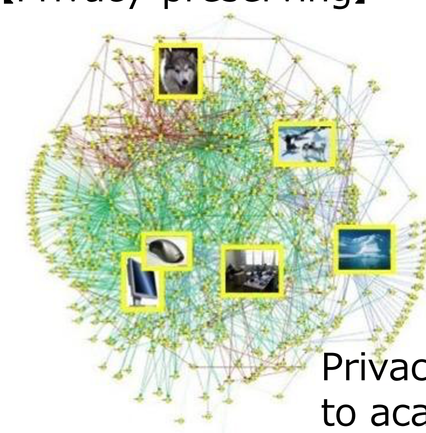
– Privacy-preserving on the Internet photos

【Large amount of annotation】

http://image-net.org/explore?wnid=n01503061

Takes 2 years, around 50k participants on AMT
100+M img DLs, 14M imgs, 21k categories

【Privacy-preserving】



IMAGENET
http://www.image-net.org/

Privacy is unclear, the usage is limited to academic/educational purpose only

The barriers of annotation labors / privacy-preserving are significant

More recent problems : AI-ethics

Dataset-related problems in CV

– Fairness / transparency in image datasets

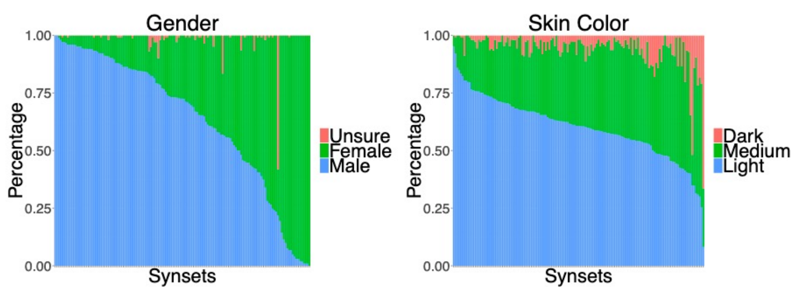
【Offensive labels】

- 80M Tiny Images contain offensive labels
- The dataset was suspended the public access

https://groups.csail.mit.edu/vision/TinyImages/

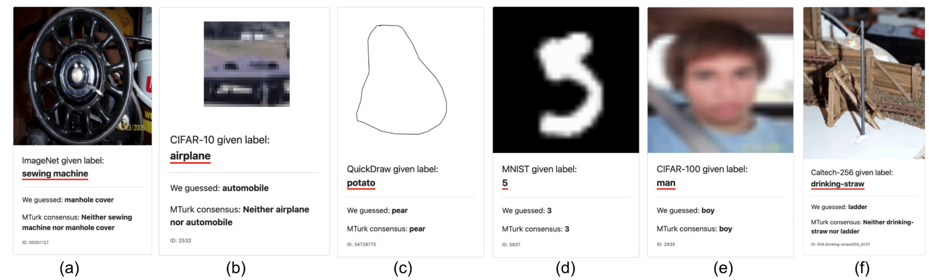
【Fairness】

Biased distributions in terms of gender/race depending on the category



https://arxiv.org/pdf/1912.07726.pdf

【Transparency】



Appr. 6% errors are included on ImageNet

C. G. Northcutt, et al. "Pervasive Label Errors in Test Sets Destabilize Machine Learning Benchmarks"
https://arxiv.org/pdf/2103.14749.pdf

Cannot ignore the ethical AI for their applications

Huge-scale datasets

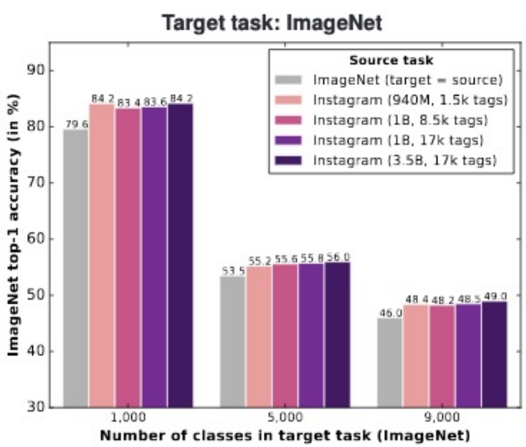
JFT-300M (Google, 2017/2021) / IG-3.5B (Meta, 2018)

300M images / 375M labels

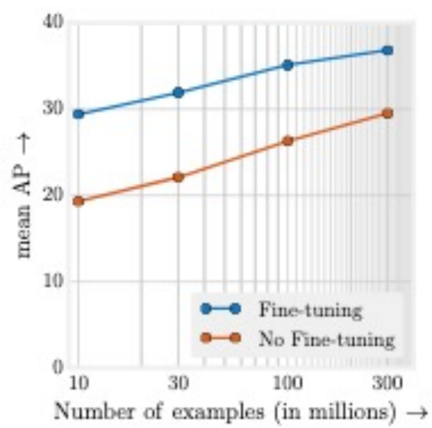
3.5B images / 3.5B weak labels

Does an image dataset x100 larger than ImageNet contributes to an enhancement of pre-trained visual model?

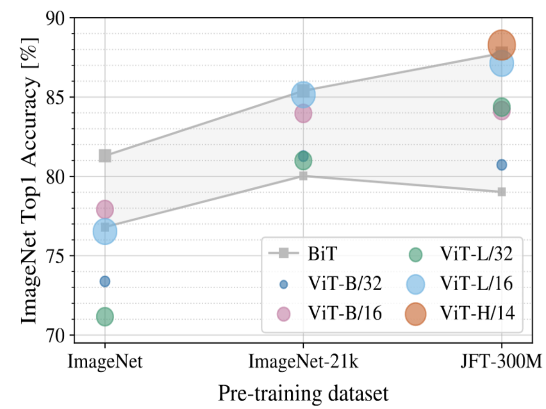
-> YES, larger-scale datasets enable to enhance a pre-trained visual model



Meta (IG-3.5B), ECCV 2018
https://arxiv.org/pdf/1805.00932.pdf



Google (JFT-300M), ICCV 2017
http://openaccess.thecvf.com/content_ICCV_2017/papers/Sun_Revisiting_Unreasonable_Effectiveness_ICCV_2017_paper.pdf



Google (JFT-300M / ViT), ICLR 2021
https://arxiv.org/pdf/2010.11929.pdf

"Larger-scale data is justified",
but private-dataset in a specific organization

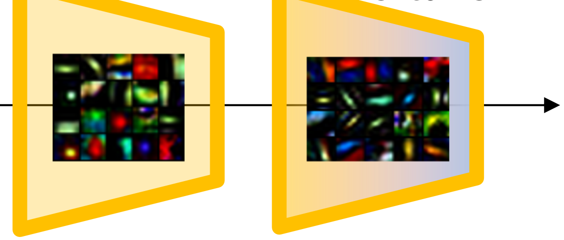
Recent vision-driven learning

Supervised Learning

e.g. ImageNet, Places, Open Images

gluon-cv.mxnet.io

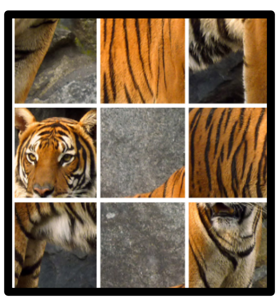
Pre-train Fine-tune



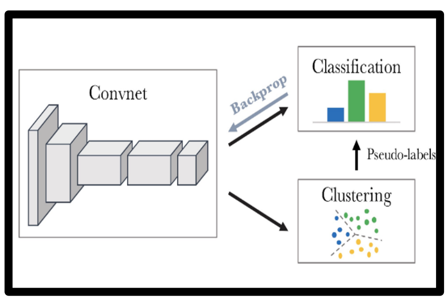
ImageNet + ResNet-50
76% @ImageNet val.

[He et al. CVPR16]

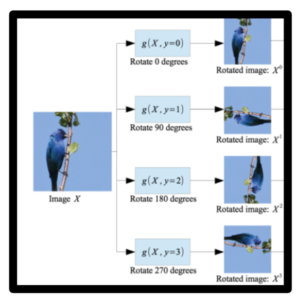
Self-supervised Learning (SSL)



Jigsaw Puzzle
[Noroozi al. ECCV16]



DeepCluster
[Caronet al. ECCV18]



Rotation Classify
[Gidaris et al. ICLR18]

SimCLR + ResNet-50
69%@ImageNet val.

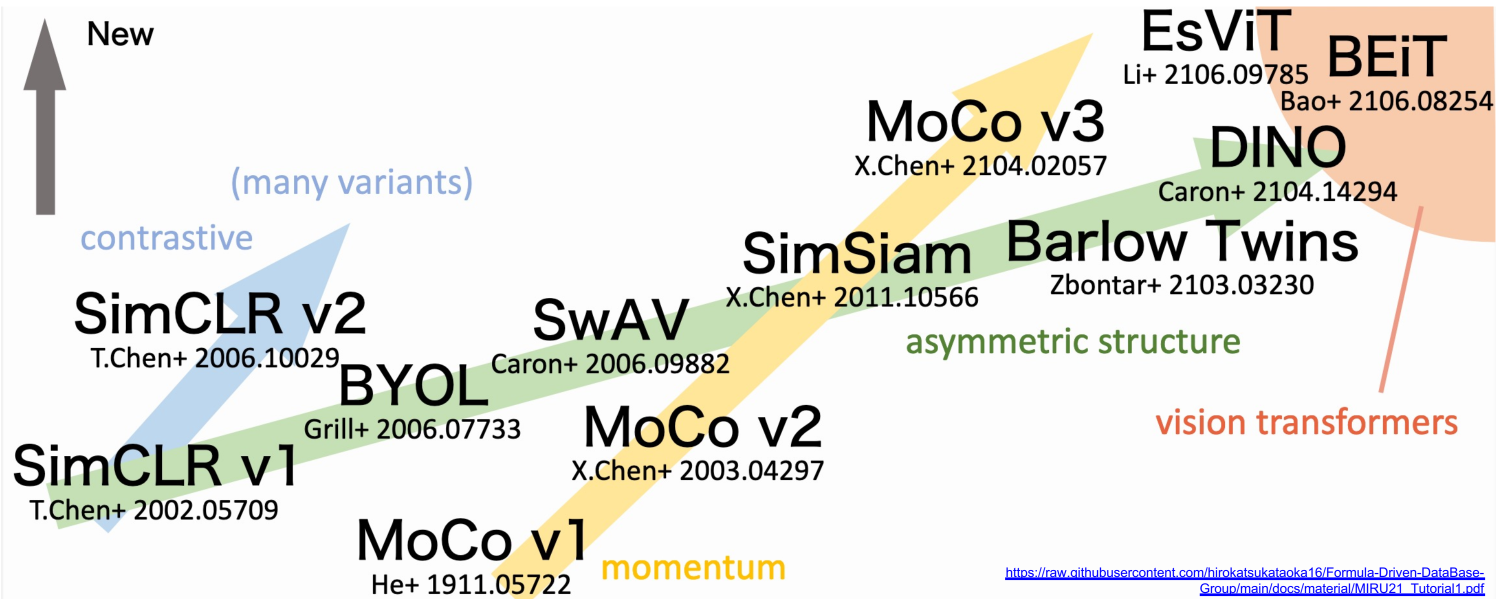
[Chen et al. ICML20]

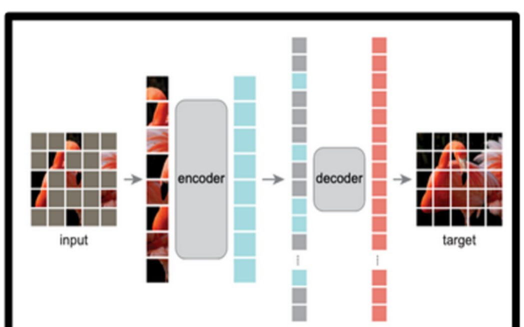
Existing the problems of image downloading and privacy-violations

Recent self-supervised learning

Brief review of SSL in 2021-2022

A recent method is closer to the performance of supervised learning





Masked AutoEncoder (MAE), CVPR 2022
Patches masking and reconstruction like BERT-MLM

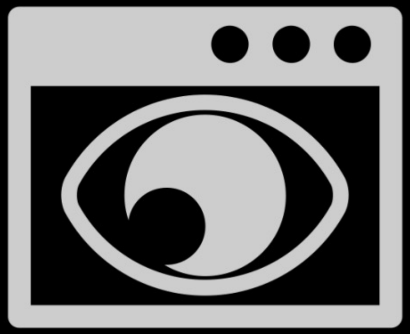
Ethical problems can occur as long as we use real images

To overcome the problems, it is better to automatically create datasets without any natural images

Annotation

FATE

Fairness, Accountability, Transparency and Ethics



Privacy

Can we pre-train DNN without any natural images?

Formula-driven Supervised Learning (FDSL)

– Automatically make image patterns and their labels
– With any mathematical formulas and/or functions

Fractal geometry from ImageNet dataset

DNN trains a natural principle from ImageNet dataset?

Directly render and train Fractals

To replace a human-annotated dataset in context of pre-training without any real images and human labels

Pre-training without Natural Images

ACCV 2020 Best Paper Honorable Mention Award
International Journal of Computer Vision (IJCV), 2022

Hirokatsu Kataoka

AIST
http://www.hirokatsukataoka.net/

Proposed method: FractalDB

Formula-Driven Supervised Learning (FDSL)

1) to make a pre-trained CNN without any natural images

2) Based on the concept, we create Fractal DataBase

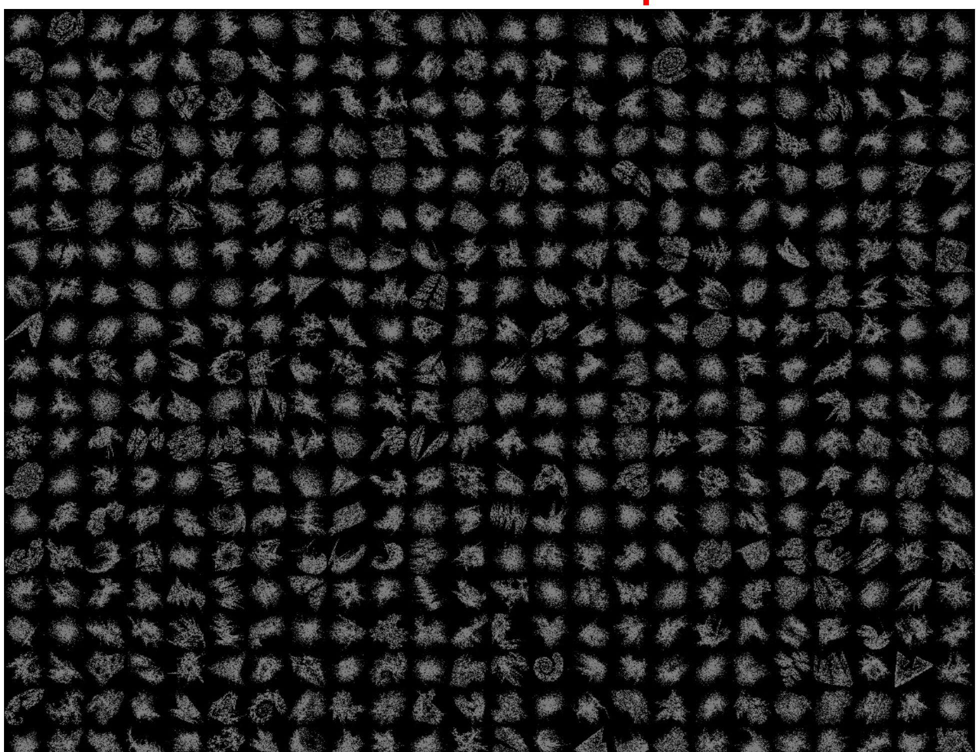
Proposed method: FractalDB

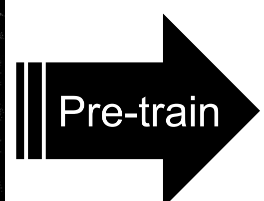
FractalDB

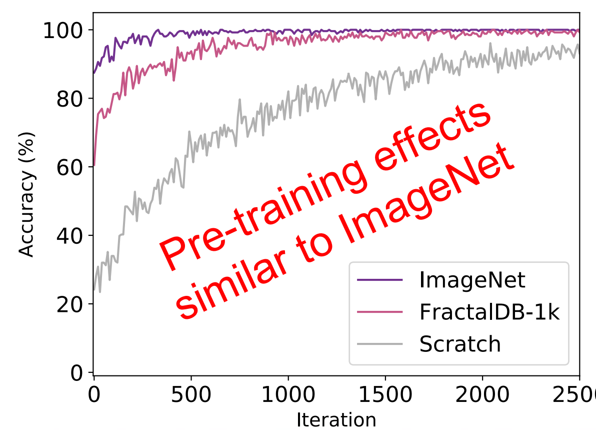
1) to make a pre-trained CNN without any natural images

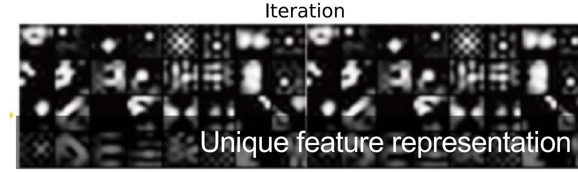
2) for a concept of Formula-driven Supervised Learning

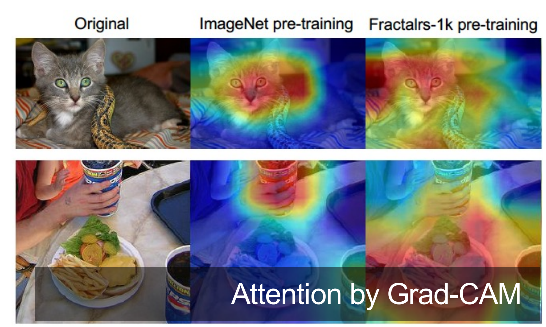
Surprising results which are similar with the effects of a supervised dataset



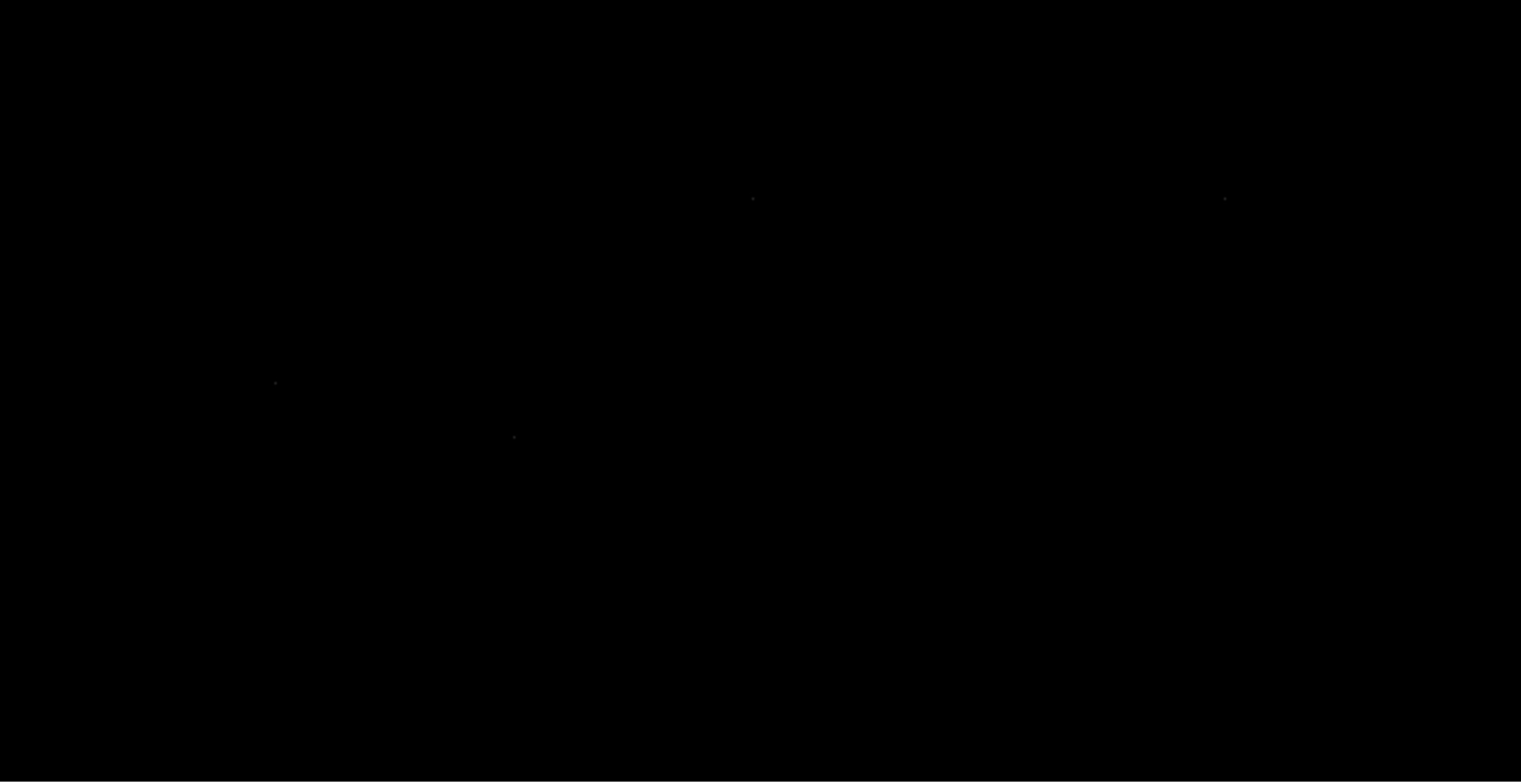








Fractal image rendering with Iterated Function System (IFS)



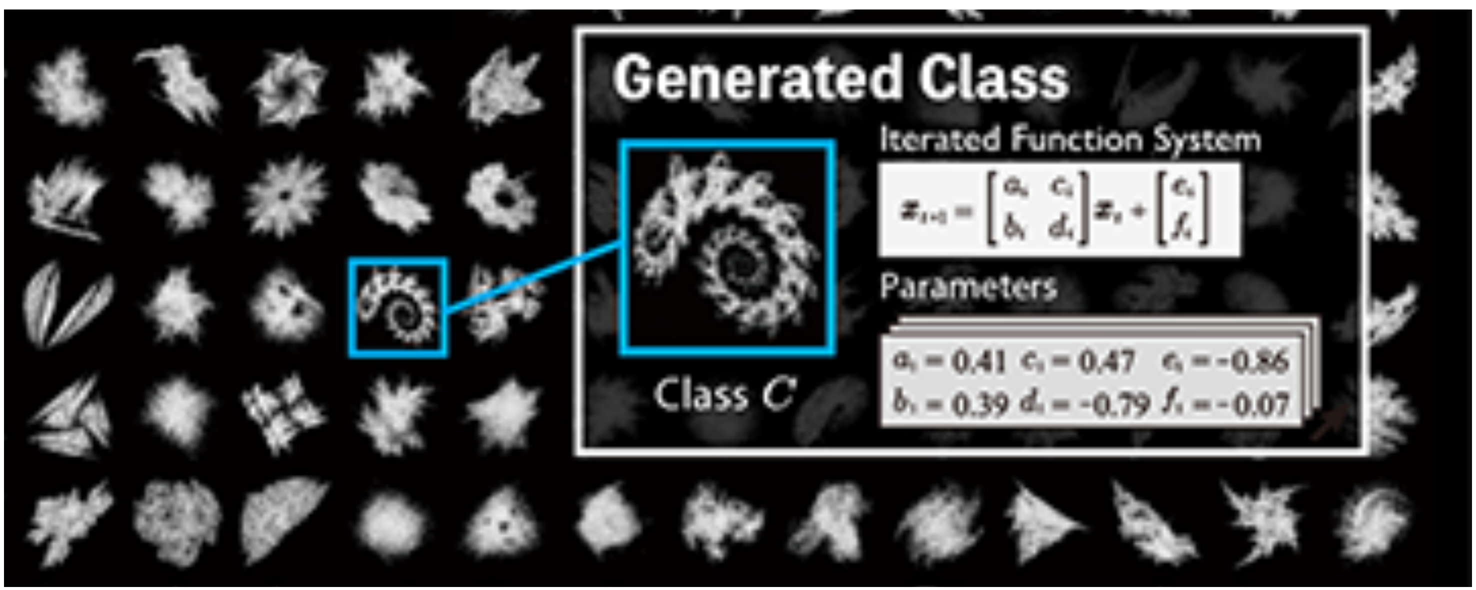
$$\mathrm{IFS} = \{\mathcal{X}; w_1, w_2, \cdots, w_N; p_1, p_2, \cdots, p_N\}$$ # Transformation probability

$$w_i(\boldsymbol{x}; \theta_i) = \begin{bmatrix} a_i & b_i \\ c_i & d_i \end{bmatrix} \boldsymbol{x} + \begin{bmatrix} e_i \\ f_i \end{bmatrix}$$ # Affine transformation

Definition of fractal category

Randomly searched image category

1. Image rendering with randomized $a \sim f$, w through IFS
2. Add category c if filling rate (> r) in the image
3. Iterate up to defined #category (C)
    - Parameter separation makes a different category

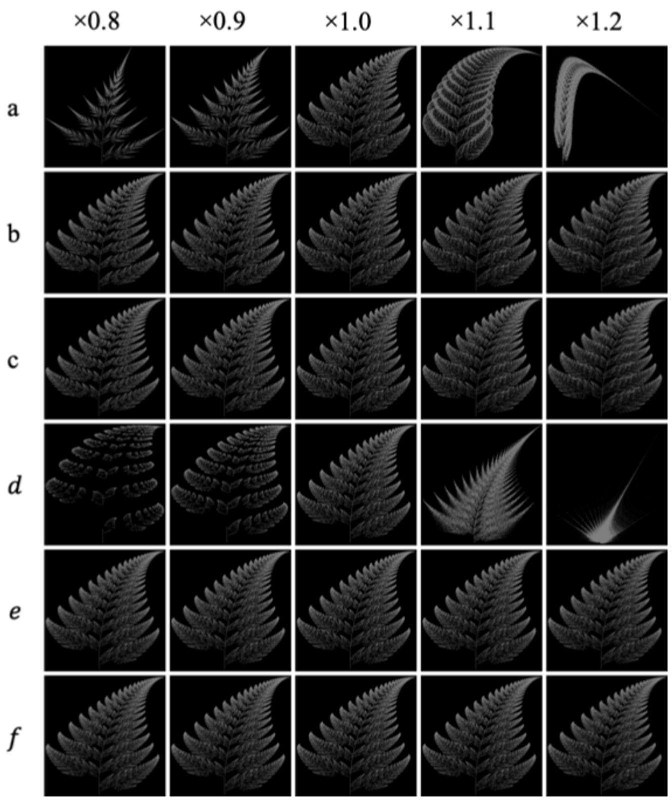


Fractal categories in FractalDB

Instance augmentation

Three different augmentation methods

1. Fluctuation of parameter set (x25)
2. Image rotation (x4)
3. Patch pattern (x10)



Parameter set (x25)

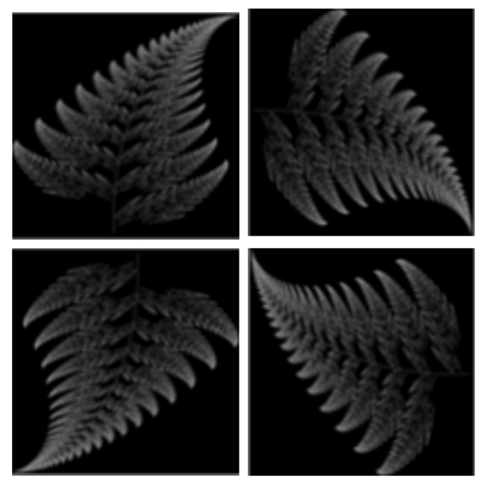
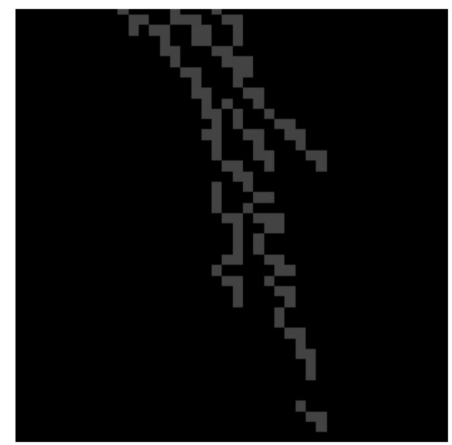


Image rotation (x4)



Patch pattern (x10)

Select ten randomly generated 3x3 patch patterns out of 256 (2^8)

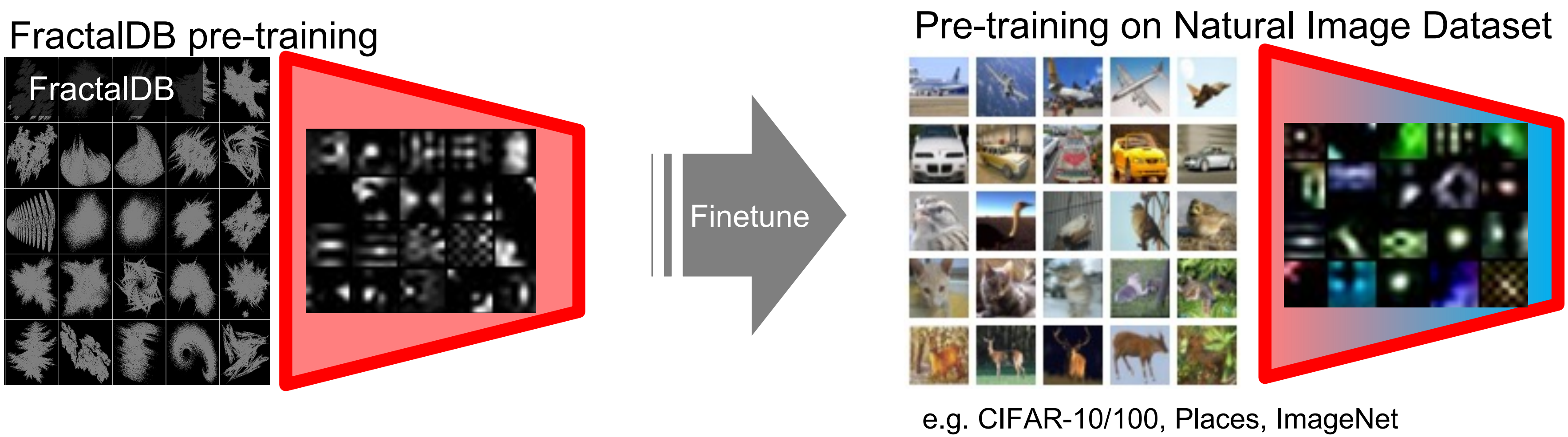
Up to x1000 instances per category

Experimental setting

Pre-training & Fine-tuning

– Pre-training without any natural images

– Fine-tuning in an ordinal way

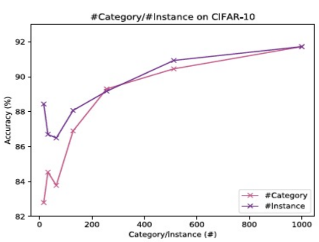


Parameters on FractalDB

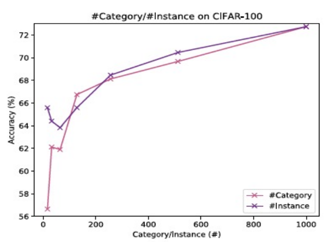
After the burden of exploration study,

– #Category, #instance, and patch-rendering are the most effective parameters in pre-training

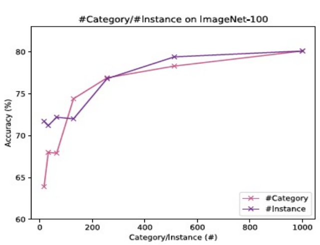
– A more difficult pre-train is slightly better in weights



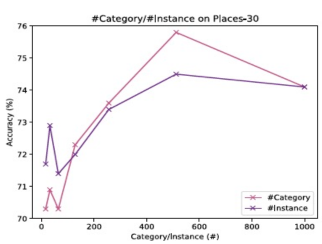
(a) CIFAR10



(b) CIFAR100



(c) ImageNet100



(d) Places30

Table 1. Patch vs. point.

	C10	C100	IN100	P30
Point	87.4	66.1	73.9	73.0
Patch (random)	**92.1**	**72.0**	**78.9**	**73.2**
Patch (fix)	**92.9**	**73.6**	**80.0**	**75.0**

Table 2. Filling rate.

	C10	C100	IN100	P30
.05	91.8	**72.4**	80.2	74.6
.10	**92.0**	72.3	**80.5**	**75.5**
.15	91.7	71.6	80.2	74.3
.20	91.3	70.8	78.8	74.7
.25	91.1	63.2	72.4	74.1

Table 3. Weights.

	C10	C100	IN100	P30
.1	92.1	72.0	78.9	73.2
.2	92.4	72.7	79.2	73.9
.3	92.4	72.6	79.2	74.3
.4	**92.7**	**73.1**	**79.6**	**74.9**
.5	91.8	72.1	78.9	73.5

Table 4. #Dot.

	C10	C100	IN100	P30
100k	**91.3**	70.8	78.8	74.7
200k	90.9	**71.0**	79.2	**74.8**
400k	90.4	70.3	**80.0**	74.5

Table 5. Image size.

	C10	C100	IN100	P30
256	**92.9**	**73.6**	80.0	75.0
362	92.2	73.2	**80.5**	**75.1**
512	90.9	71.0	79.2	73.0
724	90.8	71.0	79.2	73.0
1024	89.6	68.6	77.5	71.9

Please refer to our main paper.

Results (1/5)

Experimental comparisons on SL, SSL, and FDSL

Method	Pre-train Img	Type	C10	C100	IN1k	P365	VOC12	OG
Scratch	–	–	87.6	62.7	<u>**76.1**</u>	49.9	58.9	1.1
DC-10k	Natural	Self-supervision	89.9	66.9	66.2	<u>**51.5**</u>	67.5	15.2
Places-30	Natural	Supervision	90.1	67.8	69.1	–	69.5	6.4
Places-365	Natural	Supervision	**94.2**	76.9	71.4	–	**78.6**	10.5
ImageNet-100	Natural	Supervision	91.3	70.6	–	49.7	72.0	12.3
ImageNet-1k	Natural	Supervision	<u>**96.8**</u>	<u>**84.6**</u>	–	50.3	<u>**85.8**</u>	17.5
FractalDB-1k	Formula	Formula-supervision	93.4	75.7	70.3	49.5	58.9	**20.9**
FractalDB-10k	Formula	Formula-supervision	94.1	**77.3**	**71.5**	**50.8**	73.6	<u>**29.2**</u>

<u>**Underlined bold**</u>: best score, **Bold**: second best score

Results (1/5)

Comparison between training from scratch and proposed methods

Method	Pre-train Img	Type	C10	C100	IN1k	P365	VOC12	OG
Scratch	–	–	87.6	62.7	<u>**76.1**</u>	49.9	58.9	1.1
DC-10k	Natural	Self-supervision	89.9	66.9	66.2	<u>**51.5**</u>	67.5	15.2
Places-30	Natural	Supervision	90.1	67.8	69.1	–	69.5	6.4
Places-365	Natural	Supervision	**94.2**	76.9	71.4	–	**78.6**	10.5
ImageNet-100	Natural	Supervision	91.3	70.6	–	49.7	72.0	12.3
ImageNet-1k	Natural	Supervision	<u>**96.8**</u>	<u>**84.6**</u>	–	50.3	<u>**85.8**</u>	17.5
FractalDB-1k	Formula	Formula-supervision	93.4	75.7	70.3	49.5	58.9	**20.9**
FractalDB-10k	Formula	Formula-supervision	94.1	**77.3**	**71.5**	**50.8**	73.6	<u>**29.2**</u>

<u>**Underlined bold**</u>: best score, **Bold**: second best score

FractalDB pre-trained model achieved much higher rates than training from scratch

Results (1/5)

Comparison between SSL and proposed methods

Method	Pre-train Img	Type	C10	C100	IN1k	P365	VOC12	OG
Scratch	–	–	87.6	62.7	**<u>76.1</u>**	49.9	58.9	1.1
DC-10k	Natural	Self-supervision	89.9	66.9	66.2	**<u>51.5</u>**	67.5	15.2
Places-30	Natural	Supervision	90.1	67.8	69.1	–	69.5	6.4
Places-365	Natural	Supervision	**94.2**	76.9	71.4	–	**78.6**	10.5
ImageNet-100	Natural	Supervision	91.3	70.6	–	49.7	72.0	12.3
ImageNet-1k	Natural	Supervision	**<u>96.8</u>**	**<u>84.6</u>**	–	50.3	**<u>85.8</u>**	17.5
FractalDB-1k	Formula	Formula-supervision	93.4	75.7	70.3	49.5	58.9	**20.9**
FractalDB-10k	Formula	Formula-supervision	94.1	**77.3**	**71.5**	**50.8**	73.6	**<u>29.2</u>**

<u>Underlined bold</u>: best score, **Bold**: second best score

In the most cases, our method is better than the DeepCluster with 10k categories

Results (1/5)

Comparison between SL with 100k-order datasets and proposed methods

Method	Pre-train Img	Type	C10	C100	IN1k	P365	VOC12	OG
Scratch	–	–	87.6	62.7	**76.1**	49.9	58.9	1.1
DC-10k	Natural	Self-supervision	89.9	66.9	66.2	**51.5**	67.5	15.2
Places-30	Natural	Supervision	90.1	67.8	69.1	–	69.5	6.4
Places-365	Natural	Supervision	**94.2**	76.9	71.4	–	**78.6**	10.5
ImageNet-100	Natural	Supervision	91.3	70.6	–	49.7	72.0	12.3
ImageNet-1k	Natural	Supervision	**96.8**	**84.6**	–	50.3	**85.8**	17.5
FractalDB-1k	Formula	Formula-supervision	93.4	75.7	70.3	49.5	58.9	**20.9**
FractalDB-10k	Formula	Formula-supervision	94.1	**77.3**	**71.5**	**50.8**	73.6	**29.2**

Underlined bold: best score, **Bold**: second best score

The FractalDB pre-trained model is still better than 100k-order supervised datasets

Results (1/5)

Comparison between SL with 1M-order datasets and proposed methods

Method	Pre-train Img	Type	C10	C100	IN1k	P365	VOC12	OG
Scratch	–	–	87.6	62.7	**<u>76.1</u>**	49.9	58.9	1.1
DC-10k	Natural	Self-supervision	89.9	66.9	66.2	**<u>51.5</u>**	67.5	15.2
Places-30	Natural	Supervision	90.1	67.8	69.1	–	69.5	6.4
Places-365	Natural	Supervision	**94.2**	76.9	71.4	–	**78.6**	10.5
ImageNet-100	Natural	Supervision	91.3	70.6	–	49.7	72.0	12.3
ImageNet-1k	Natural	Supervision	**<u>96.8</u>**	**<u>84.6</u>**	–	50.3	**<u>85.8</u>**	17.5
FractalDB-1k	Formula	Formula-supervision	93.4	75.7	70.3	49.5	58.9	**20.9**
FractalDB-10k	Formula	Formula-supervision	94.1	**77.3**	**71.5**	**50.8**	73.6	**<u>29.2</u>**

<u>Underlined bold</u>: best score, **Bold**: second best score

Our method partially surpasses the ImageNet/Places pre-trained models

Results (2/5)

Mtd	PT Img	C10	C100	IN1k	P365	VOC12	OG
DC-10k	Natural	89.9	66.9	66.2	51.2	67.5	15.2
DC-10k	Formula	83.1	57.0	65.3	**53.4**	60.4	15.3
F1k	Formula	93.4	75.7	70.3	49.5	58.9	20.9
F10k	Formula	**94.1**	**77.3**	**71.5**	50.8	**73.6**	**29.2**

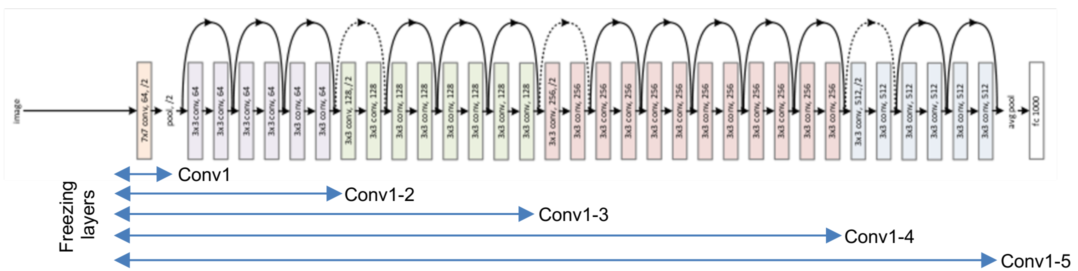
Bold: best score

DC-10k with fractal images cannot effectively pre-train to recognize natural images

This shows our method assigns an appropriate image pattern and the category

Results (3/5)

Freezing layer(s)	C10	C100	IN100	P30
Fine-tuning	93.4	75.7	82.7	75.9
Conv1	92.3	72.2	77.9	74.3
Conv1–2	92.0	72.0	77.5	72.9
Conv1–3	89.3	68.0	71.0	68.5
Conv1–4	82.7	56.2	55.0	58.3
Conv1–5	49.4	24.7	21.2	31.4

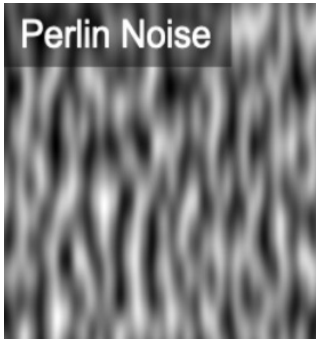


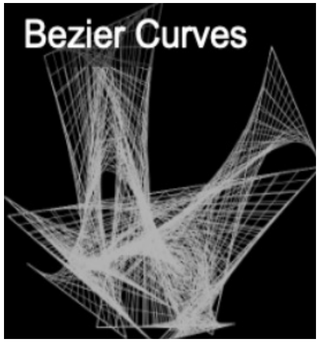
Full fine-tuning is the best

Moreover, earlier layers tend to be good feature representations

Results (4/5)

Pre-training	C10	C100	IN100	P30
Scratch	87.6	60.6	75.3	70.3
Bezier-144	87.6	62.5	72.7	73.5
Bezier-1024	89.7	68.1	73.0	73.6
Perlin-100	90.9	70.2	73.0	73.3
Perlin-1296	90.4	71.1	79.7	74.2
FractalDB-1k	**93.4**	**75.7**	**82.7**	**75.9**



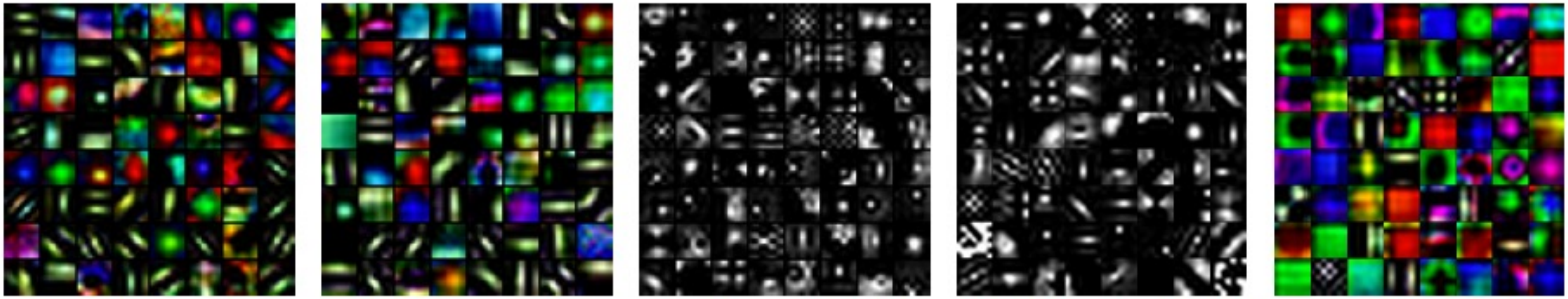


We compare Formula-driven Supervised Learning with other principles

The FractalDB pre-trained model outperforms other methods

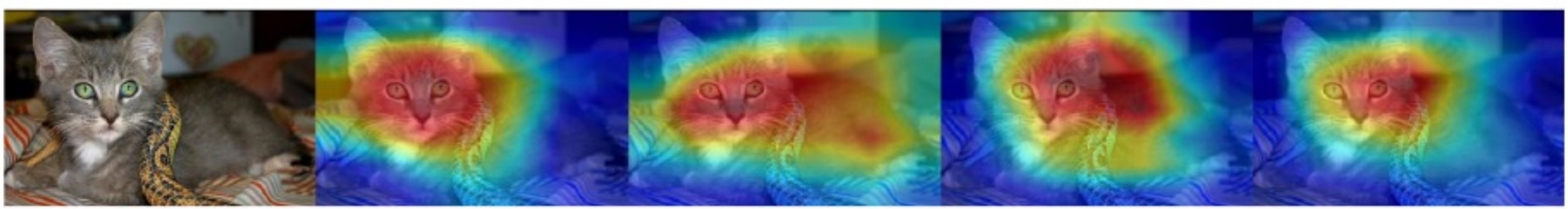
Results (5/5)

Visualization of Conv1



(a) ImageNet (b) Places365 (c) Fractal-1K (d) Fractal-10K (e) DC-10k

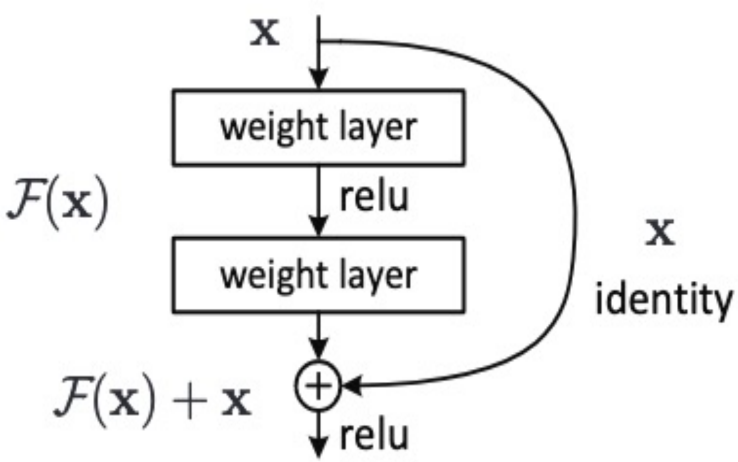
Original | ImageNet-1k →CIFAR-10 | Places365 →CIFAR-10 | FractalDB-1k →CIFAR-10 | FractalDB-10k →CIFAR-10



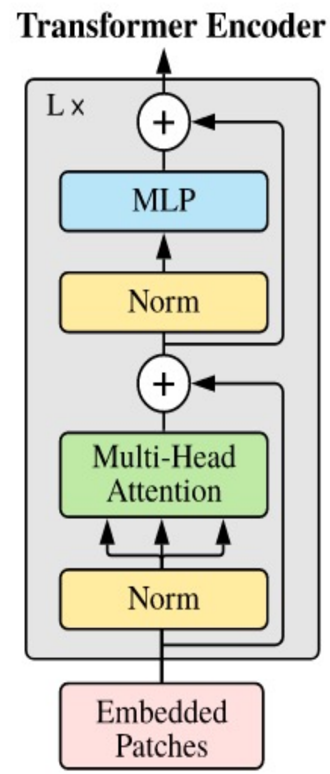
FractalDB pre-trained model acquires different representations yet look at a similar area

Paradigm Shift in Computer Vision

‘Convolution’ to ‘Self-attention’



[He al. CVPR16]



[Vaswani al. NIPS17]
Figure from [Dosovitskiy al. ICLR21]

Can Vision Transformers Learn without Natural Images?

AAAI 2022

Hirokatsu Kataoka

AIST

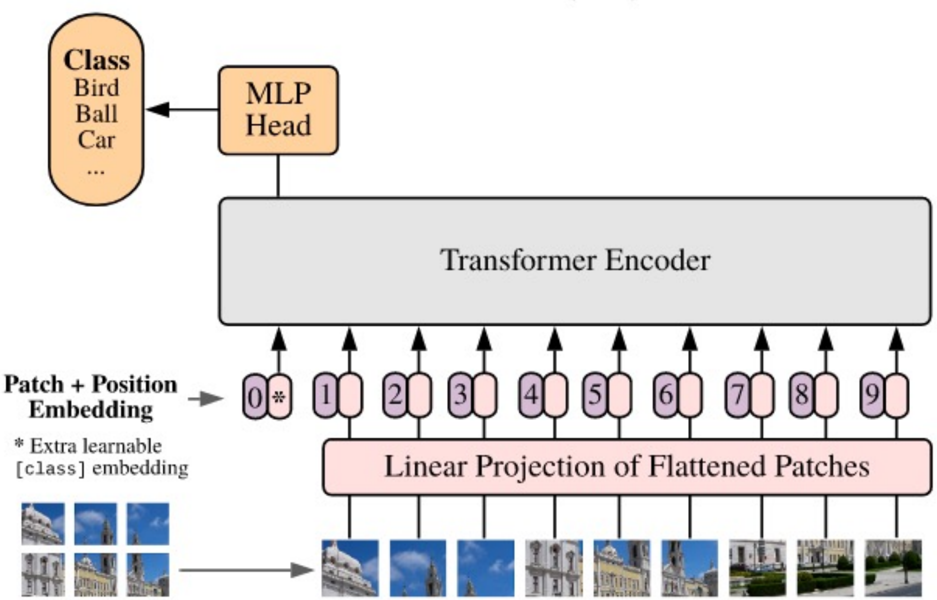
http://www.hirokatsukataoka.net/

Vision Transformer (ViT), so far

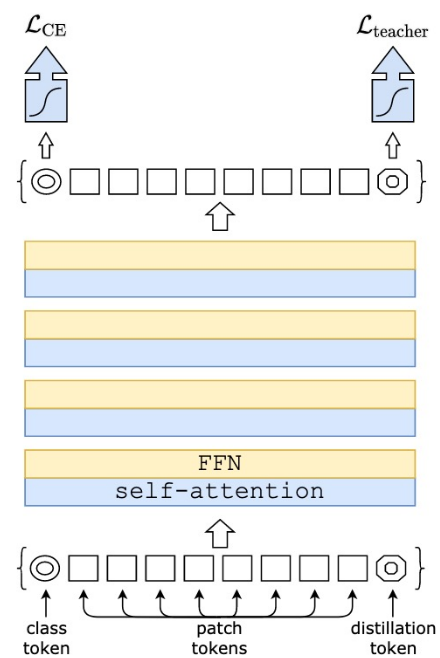
One more shift in Transformer

- ViT to DeiT (Data-efficient image Transformer)
- JFT-300M to ImageNet-1k in pre-training

Can ViT learn without real images?



[Dosovitskiy al. ICLR21]



[Touvron al. arXiv20]

Settings of Architecture and Dataset

Architecture

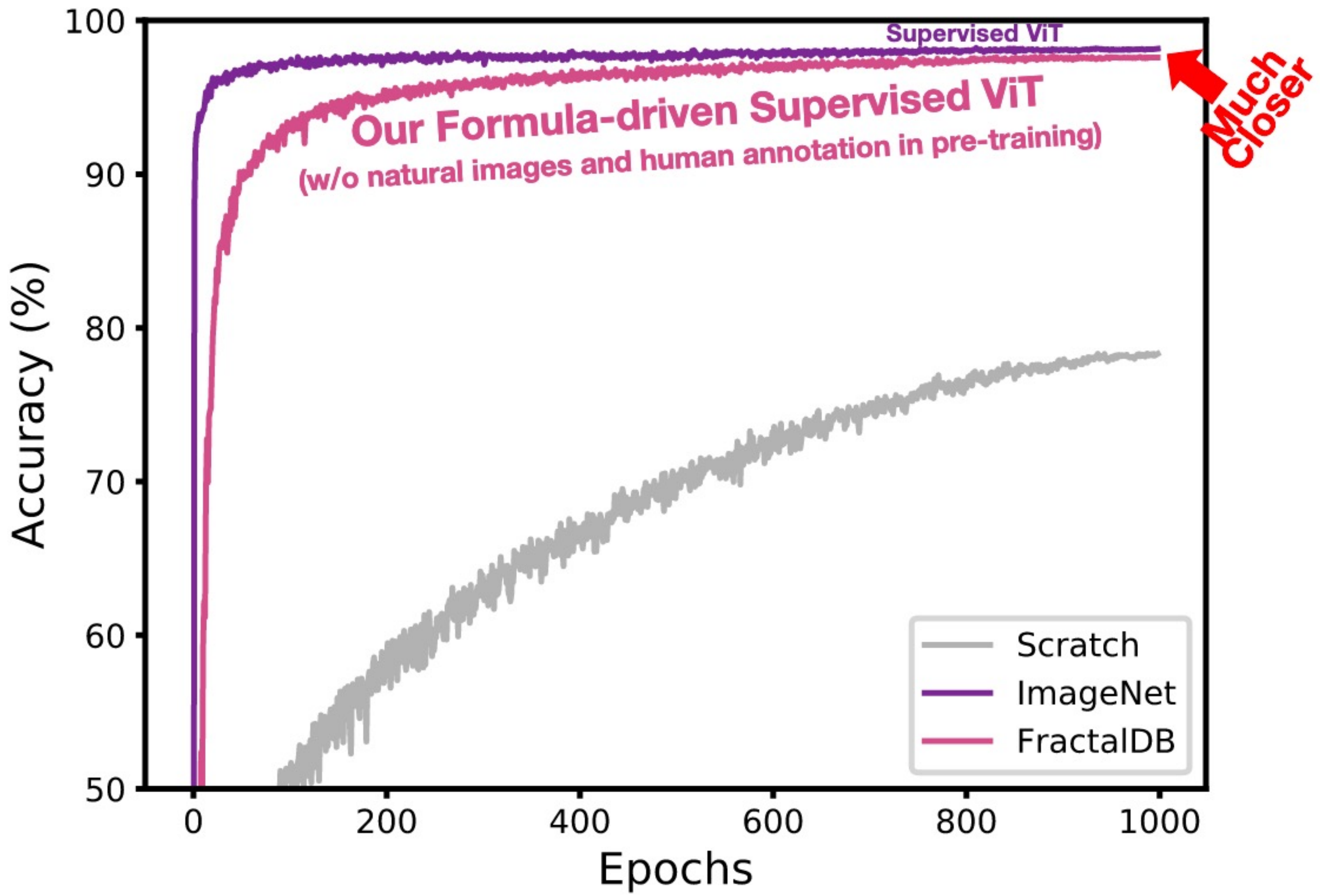
- ViT
  - No big difference from ViT on real image datasets
  - We assign richer data augmentation proposed in DeiT

Dataset

- FractalDB
  - Grayscale is better than colored FractalDB
    - ResNet: colored FractalDB is slightly better
    - DeiT: grayscale FractalDB is better
  - Longer pre-training is better
    - 300 epochs in ViT

FractalDB pre-trained DeiT

– We succeeded a DeiT training without natural images



Results (1/2)

vs. Supervised Learning

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	–	<u>**89.4**</u>
ImageNet-100	Natural	Supervision	94.7	77.8	67.4	97.2	78.8	78.1	–
ImageNet-1k	Natural	Supervision	<u>**98.0**</u>	<u>**85.5**</u>	<u>**89.9**</u>	<u>**99.4**</u>	<u>**88.7**</u>	<u>**80.0**</u>	–
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**

<u>**Underlined bold**</u>: best score, **Bold**: second best score

FractalDB pre-trained model achieved much higher rates than training from scratch

Results (1/2)

vs. Supervised Learning

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	–	<u>**89.4**</u>
ImageNet-100	Natural	Supervision	94.7	77.8	67.4	97.2	78.8	78.1	–
ImageNet-1k	Natural	Supervision	<u>**98.0**</u>	<u>**85.5**</u>	<u>**89.9**</u>	<u>**99.4**</u>	<u>**88.7**</u>	<u>**80.0**</u>	–
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**

<u>**Underlined bold**</u>: best score, **Bold**: second best score

Though our method cannot beat the ImageNet pre-trained model,

the FractalDB pre-trained model partially surpasses the Places pre-trained models

Results (2/2)

vs. Self-supervised Learning

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	<u>**80.8**</u>	87.1
SimCLRv2	YES	**97.4**	<u>**84.1**</u>	**84.9**	<u>**98.9**</u>	**86.2**	80.0	**88.5**
FractalDB-10k	NO	<u>**97.6**</u>	**83.5**	<u>**87.7**</u>	**98.8**	<u>**86.9**</u>	78.5	<u>**88.8**</u>

<u>**Underlined bold**</u>: best score, **Bold**: second best score

The proposed method recorded higher accuracies than SSL methods

with MoCoV2, Rotation, and Jigsaw

Results (2/2)

vs. Self-supervised Learning

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	<u>**80.8**</u>	87.1
SimCLRv2	YES	**97.4**	<u>**84.1**</u>	**84.9**	<u>**98.9**</u>	**86.2**	80.0	**88.5**
FractalDB-10k	NO	<u>**97.6**</u>	**83.5**	<u>**87.7**</u>	**98.8**	<u>**86.9**</u>	78.5	<u>**88.8**</u>

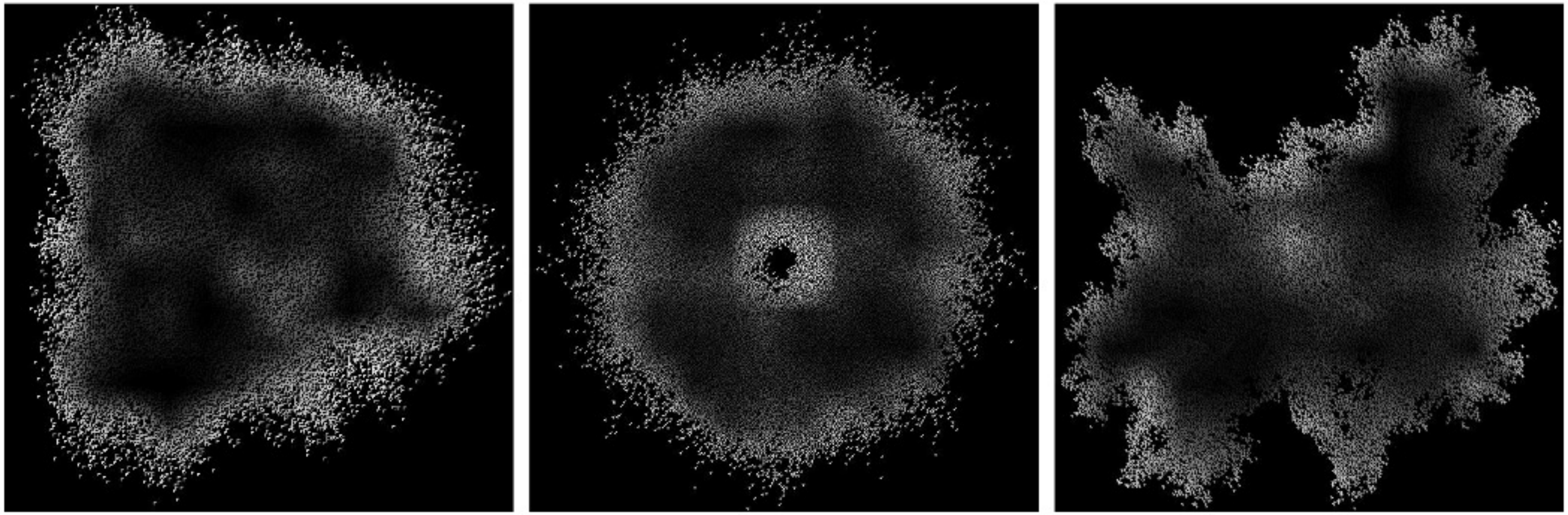
<u>**Underlined bold**</u>: best score, **Bold**: second best score

The FractalDB-10k pre-trained DeiT performs slightly higher in average accuracy on representative datasets (88.8 vs. 88.5)

Visualization of attention maps

FractalDB pre-trained model focused on contours

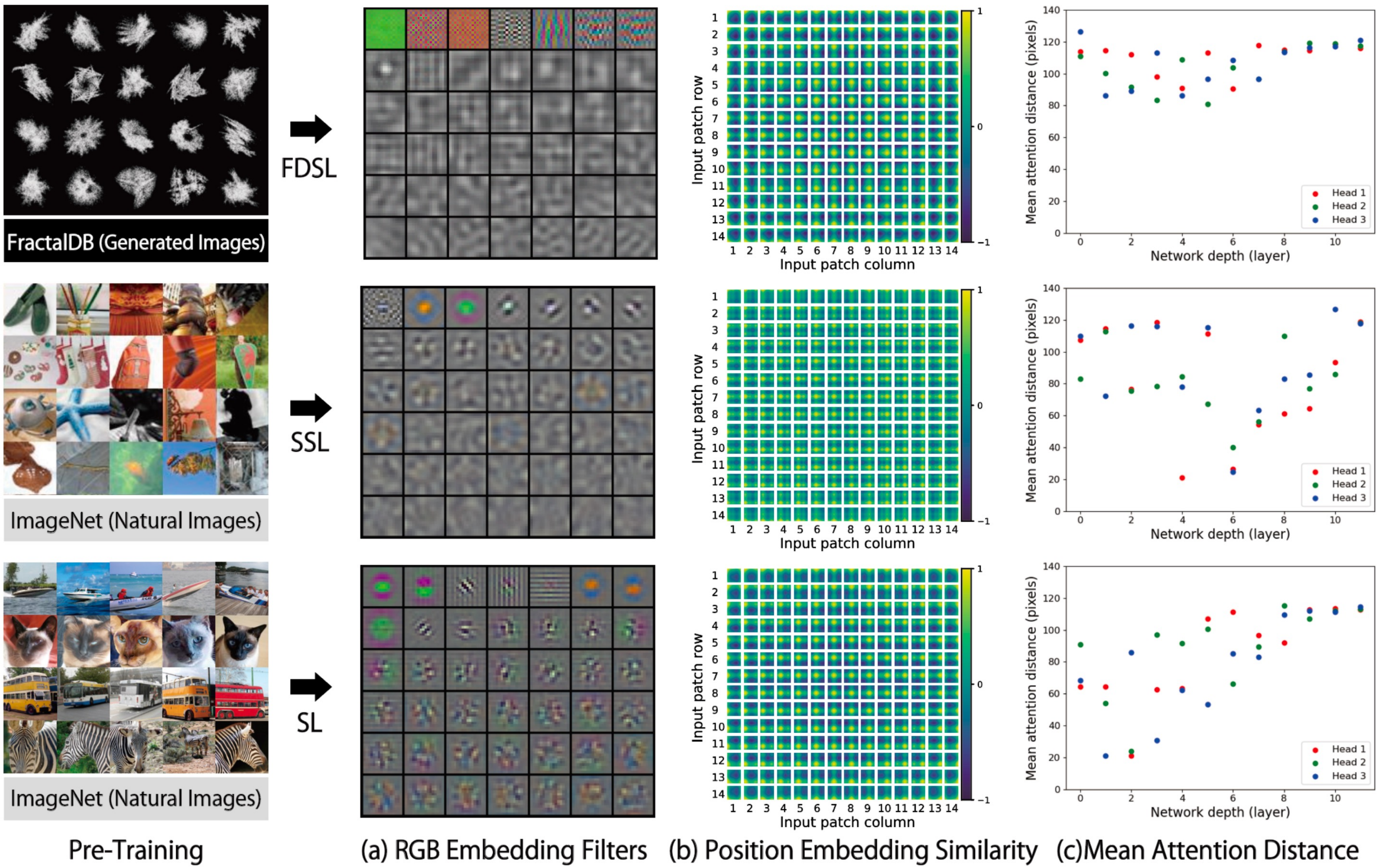
– The figures show attention on fractal images



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

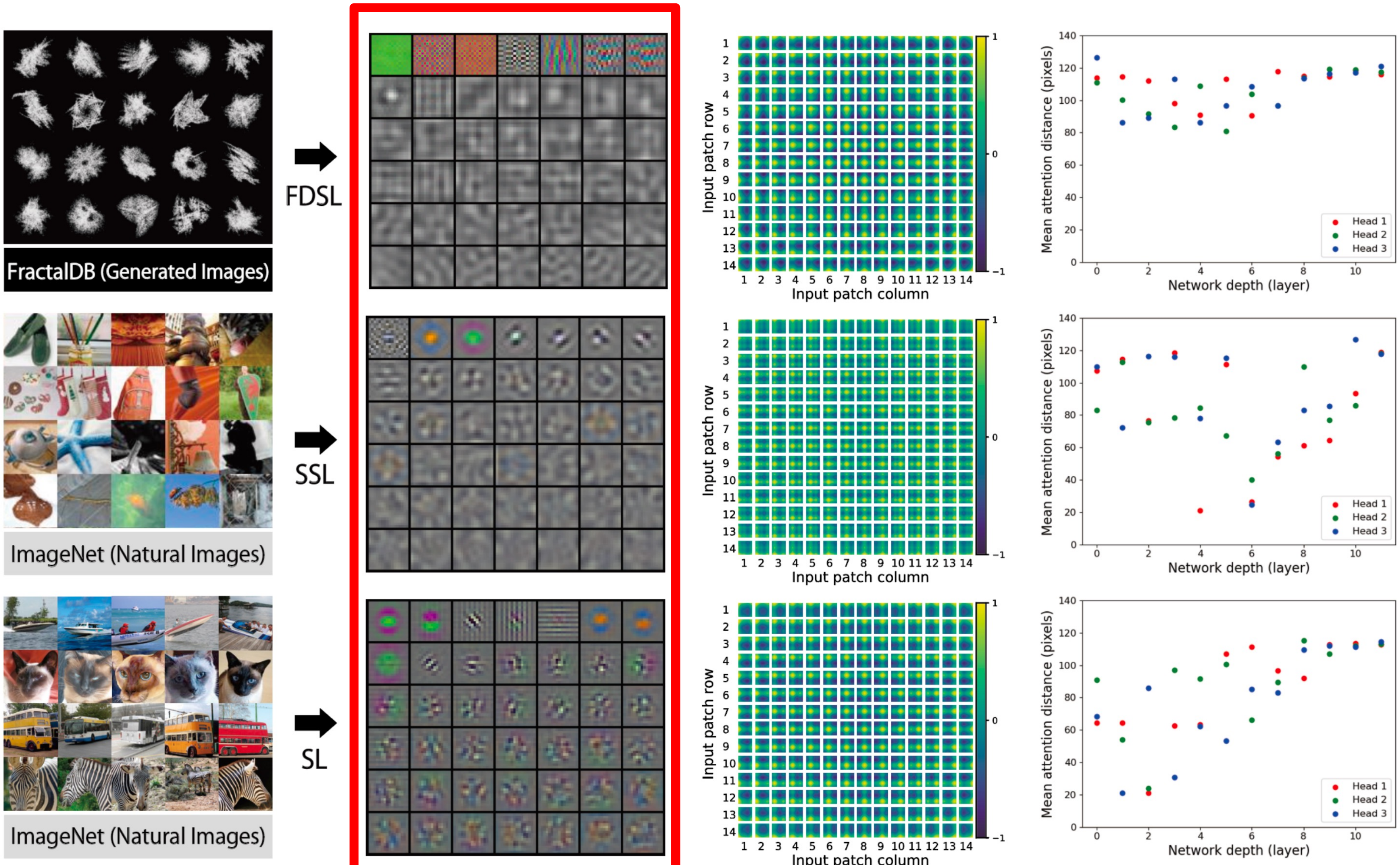
Visualization

Characteristics of FDSL, SSL, and SL



Visualization of embedding filters

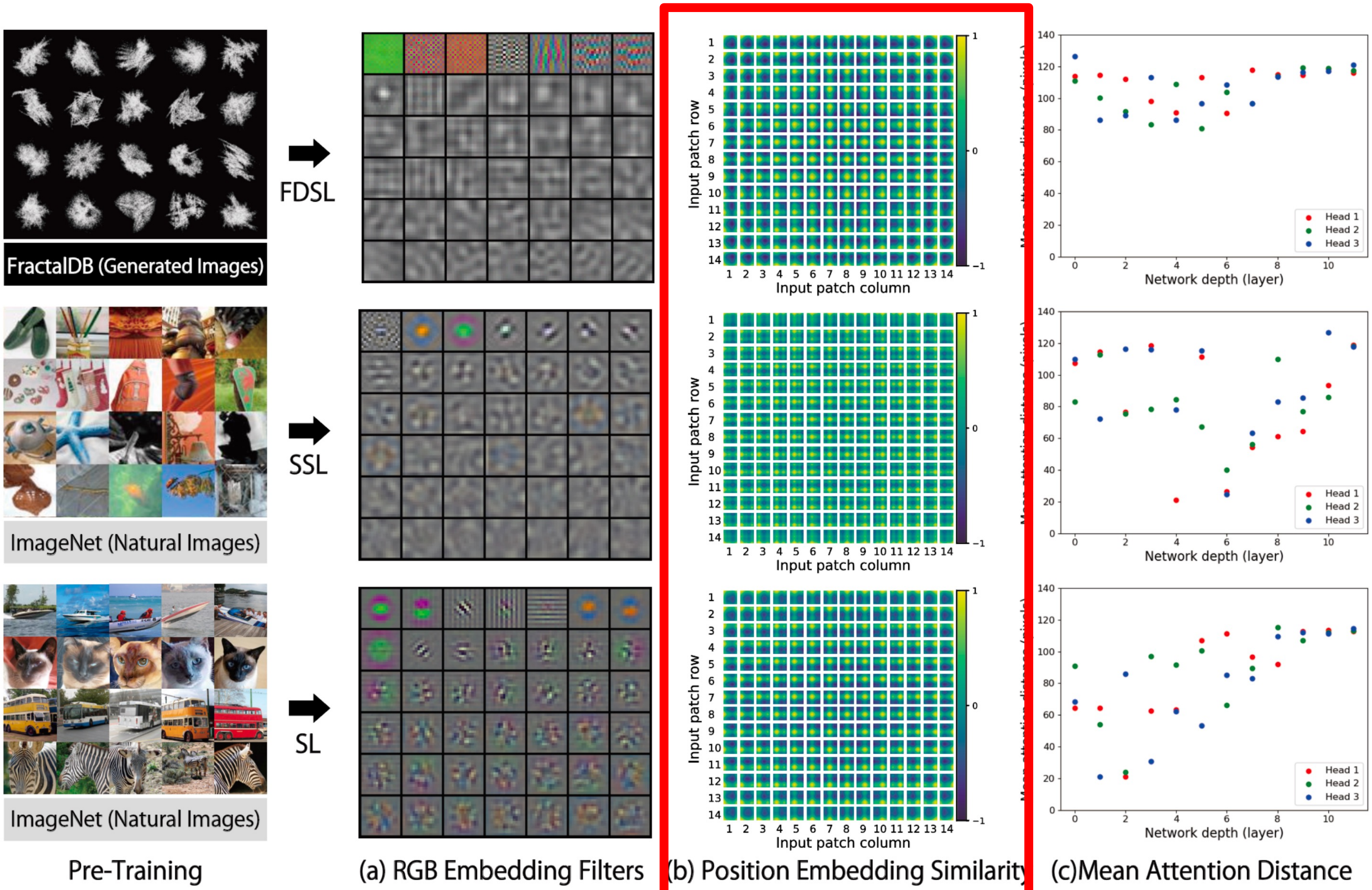
Visual representation in the initial filter



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

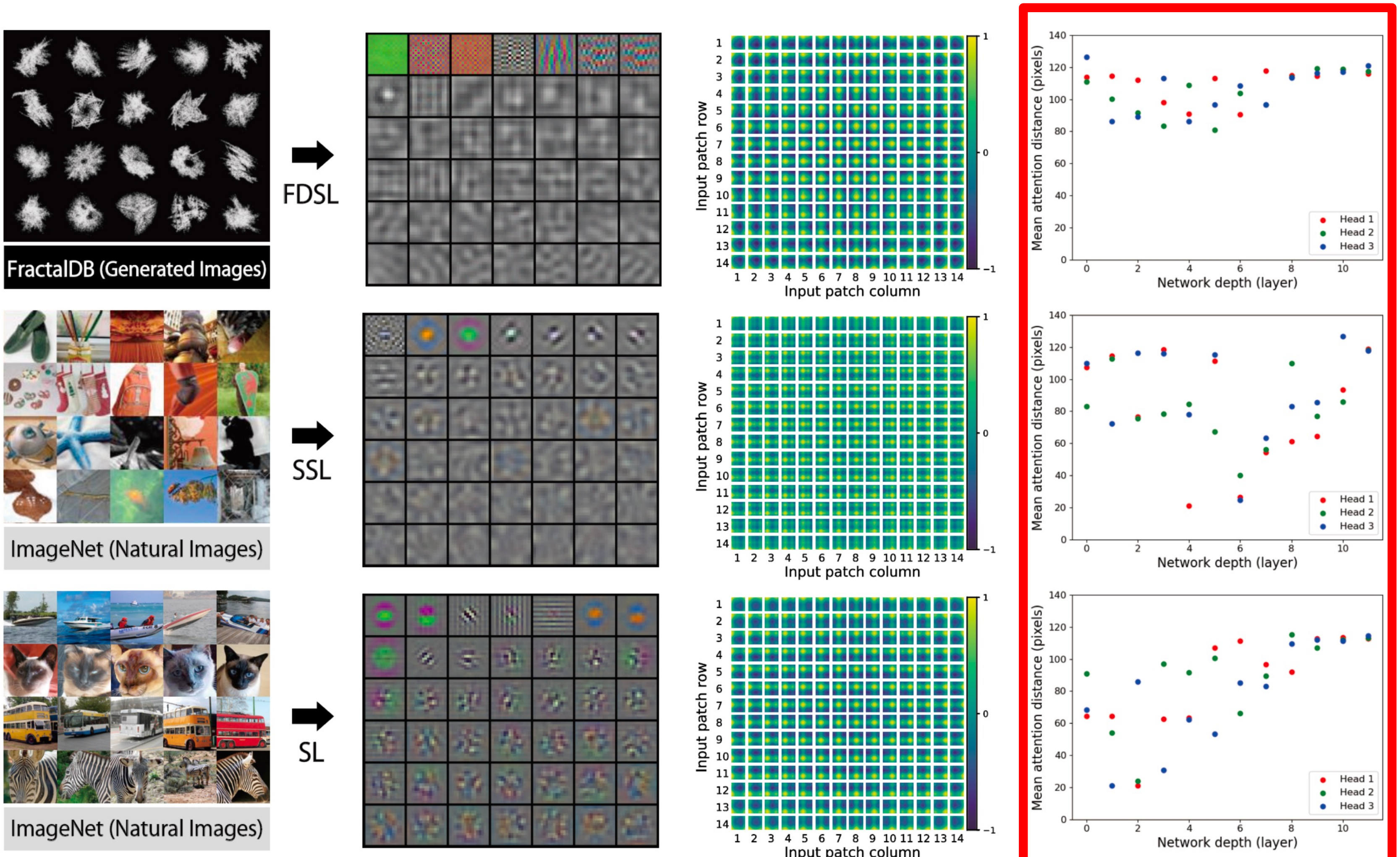
Visualization of position embedding similarity

Cosine similarity of positional embedding



Visualization of mean attention distance

FDSL tends to look at wide-spread areas



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

Can vision transformers learn without natural images?

→ Answer is “Yes”. The FractalDB pre-training achieved to nearly perform the ImageNet-1k pre-training.

CVPR JUNE 19-24 2022
NEW ORLEANS • LOUISIANA

Replacing Labeled Real-image Datasets with Auto-generated Contours

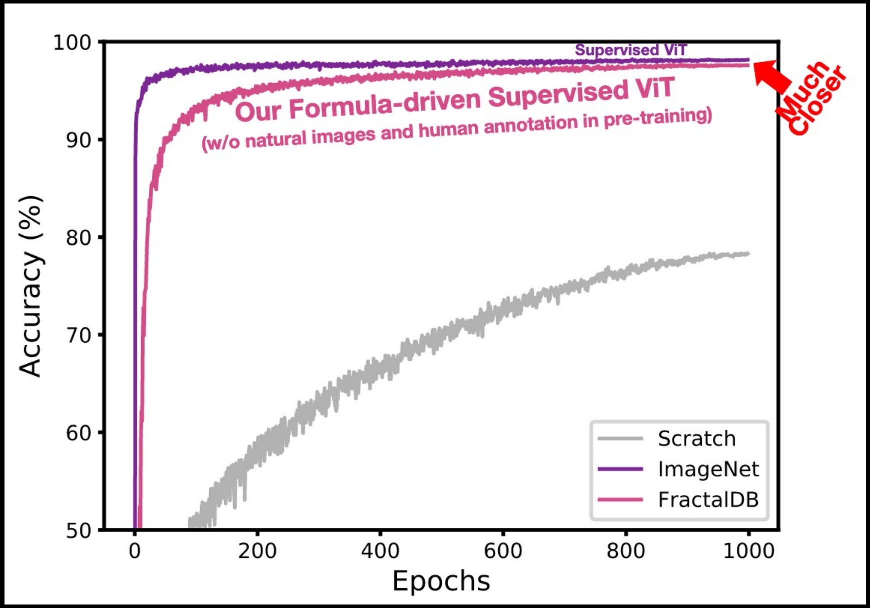
CVPR 2022

Hirokatsu Kataoka[*], Ryo Hayamizu[*], Ryosuke Yamada[*], Kodai Nakashima[*], Sora Takashima[*,], Xinyu Zhang[*,**], Edgar Josafat MARTINEZ-NORIEGA[*,**], Nakamasa Inoue[*,**], Rio Yokota[*,**]**

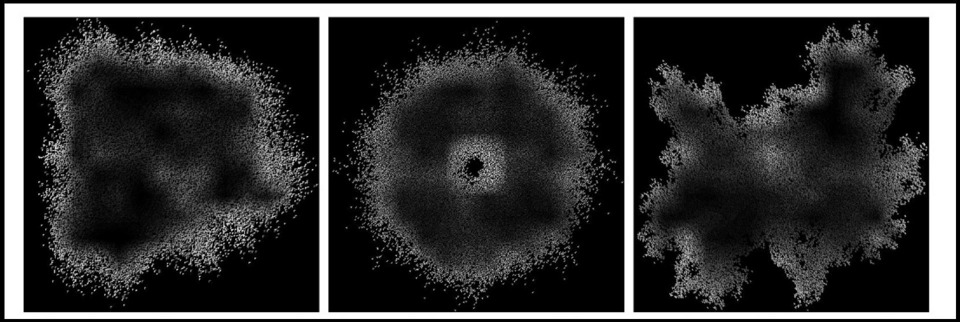
*** National Institute of Advanced Industrial Science and Technology (AIST)**
****Tokyo Institute of Technology**

Successfully constructed a FractalDB pre-trained ViT

Conventional 14M to 0 in terms of real images



Visualizing self-attention in ViT



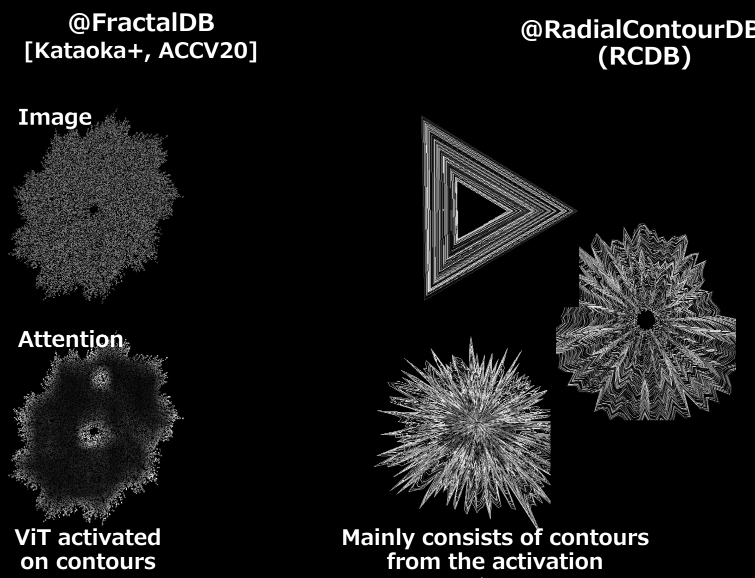
→The fact describes that it is not important the usage of Fractal, it just focuses on object contours?

CVPR 2022 accepted paper

Replacing labeled real-image datasets with auto-generated contours

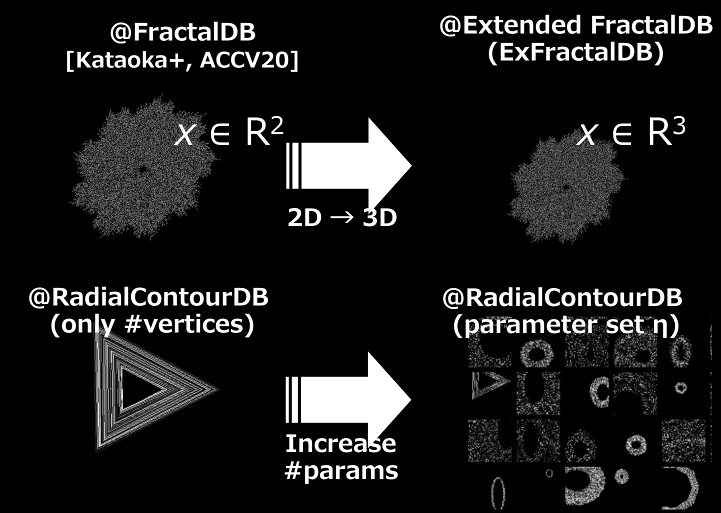
We have verified two different hypotheses

Hypothesis 1:
Object contours are what matter in FDSL datasets



As the extreme case of contour classification, we implemented RCDB mainly consists of radial contours

Hypothesis 2:
Task difficulty matters in FDSL pre-training



“#parameters” relate to task difficulty

Validation on classification, detection, and segmentation

ImageNet-1k / MS COCO dataset

Image Classification / Object Detection, Instance Segmentation

Real images: ImageNet-21k

Accuracy on ImageNet-1k

81.8%

3D fractal images: ExFractalDB-21k

82.7%

Contour images: RCDB-21k

82.4%

Pre-training	COCO Det AP_{50} / AP / AP_{75}	COCO Inst Seg AP_{50} / AP / AP_{75}
Scratch	63.7 / 42.2 / 46.1	60.7 / 38.5 / 41.3
ImageNet-1k	69.2 / 48.2 / 53.0	66.6 / 43.1 / 46.5
ImageNet-21k	**70.7 / 48.8 / 53.2**	**67.7 / 43.6 / 47.0**
ExFractalDB-1k	69.1 / **48.0 / 52.8**	66.3 / **42.8** / 45.9
ExFractalDB-21k	**69.2 / 48.0** / 52.6	**66.4 / 42.8 / 46.1**
RCDB-1k	68.3 / 47.4 / 51.9	65.7 / 42.2 / 45.5
RCDB-21k	67.7 / 46.6 / 51.2	64.8 / 41.6 / 44.7

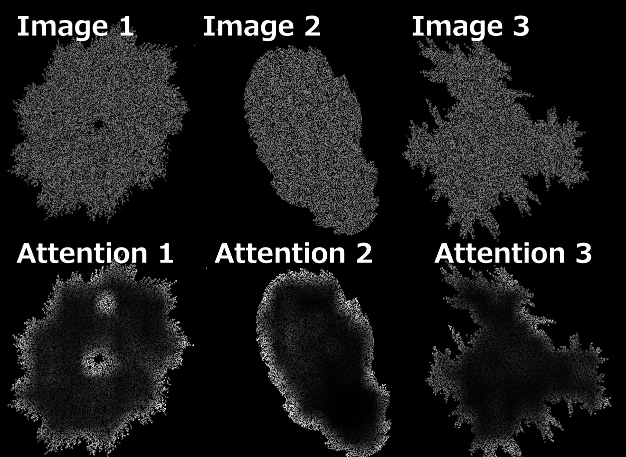
Exceeded ImageNet-21k pre-training
Radial contours also surpassed the accuracy with ImageNet pre-training in addition to Fractal pre-training

Our pre-trained models perform good fine-tuning rates on COCO dataset even though the usage of contour classification only

Hypothesis 1

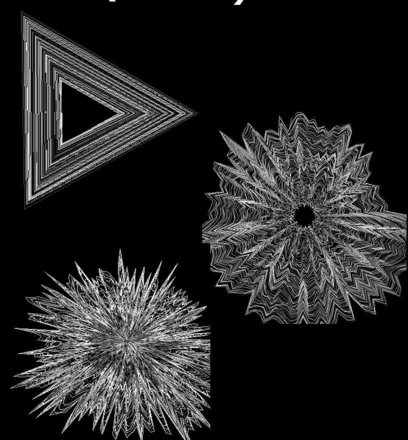
Object contours are what matter in FDSL datasets

@FractalDB [Kataoka+, ACCV20]



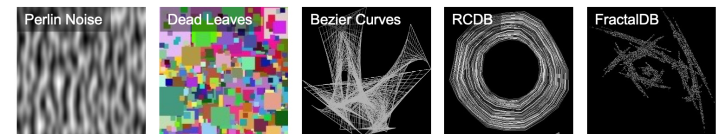
ViT activated on contours of fractal images

@RadialContourDB (RCDB)



RCDB mainly consists of contours

Pre-training	C10	C100	Cars	Flowers
Scratch	78.3	57.7	11.6	77.1
Perlin Noise [21]	95.0	78.4	70.6	96.1
Dead Leaves [3]	95.9	79.6	72.8	96.9
Bezier Curves [21]	96.7	80.3	82.8	98.5
RCDB	**96.8**	**81.6**	84.2	**98.7**
FractalDB [27]	**96.8**	**81.6**	**86.0**	98.3



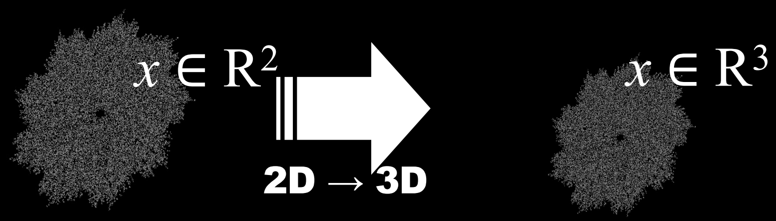
Radial contour pre-training similarly reached FractalDB without heavy tuning

Hypothesis 2

Task difficulty matters in FDSL pre-training

@FractalDB
[Kataoka+, ACCV20]

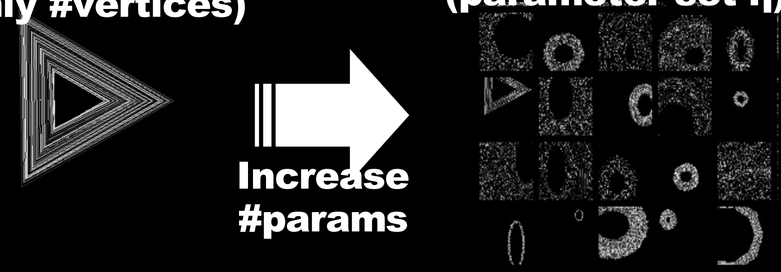
@Extended FractalDB (ExFractalDB)



- 3D Fractal rendering
- Projecting onto 2D image plane from a random viewpoint

@RadialContourDB (only #vertices)

@RadialContourDB (parameter set n)



- We mainly adjust #vertices
- Additional parameters, e.g., #polygons, smoothness for category generation

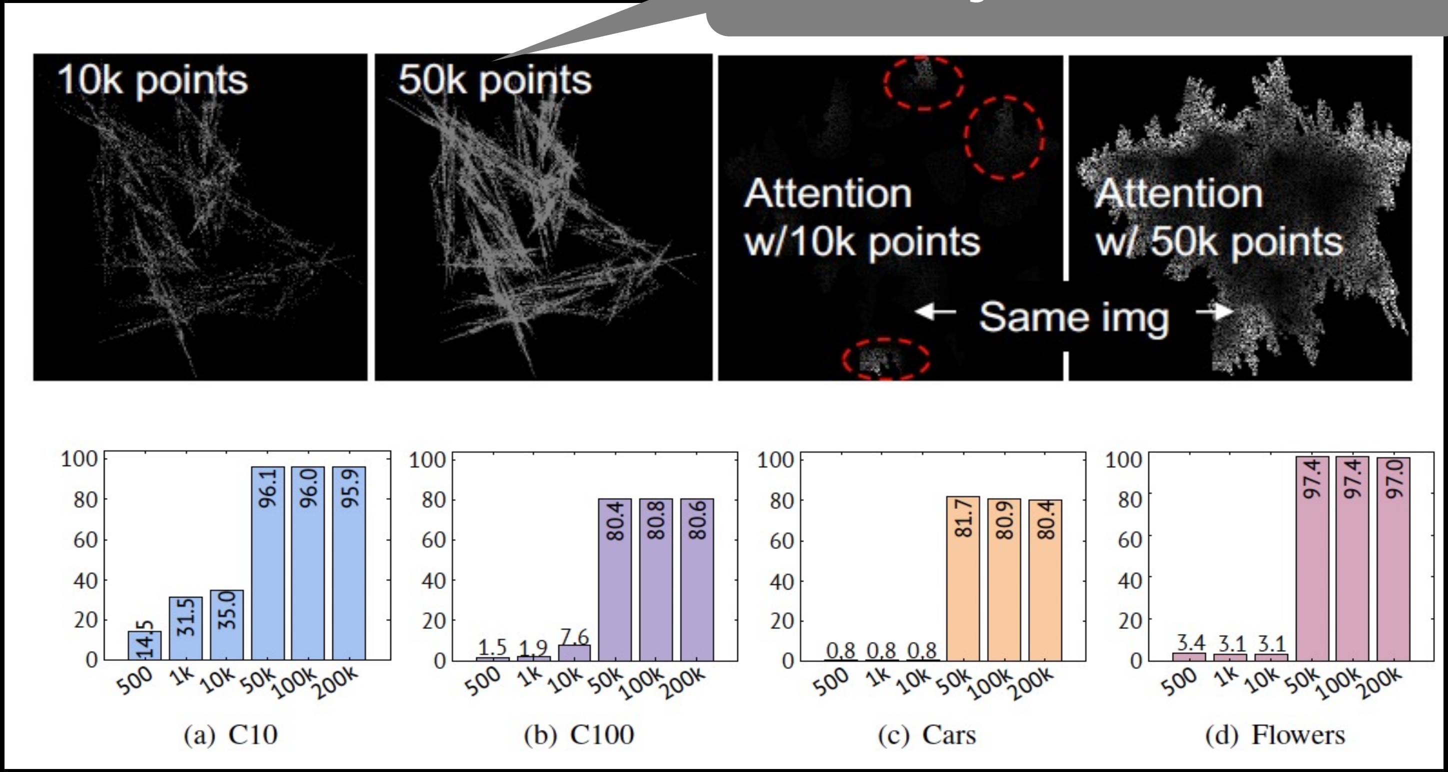
Pre-training	C10	C100	Cars	Flowers
BC	96.9 (0.2)	81.4 (1.1)	85.9 (3.1)	97.9 (-0.6)
RCDB	97.0 (0.2)	**82.2** (0.6)	86.5 (2.4)	**98.9** (0.2)
ExFractalDB	**97.2** (0.4)	81.8 (0.2)	**87.0** (1.0)	**98.9** (0.6)

In relation to #formula-parameters, the image variation contributes to the pre-training effect

Failure modes (ExFractalDB)

Investigate when and how FDSL can fail

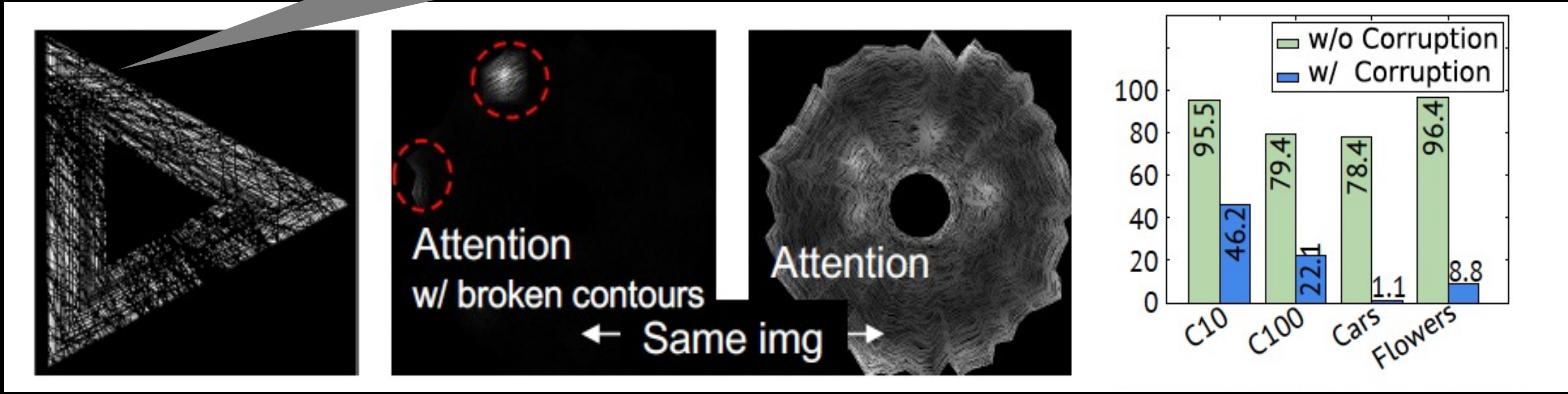
Fractal images start to form a contour in 50k or higher



In point-rendered FractalDB, although the fractal images with 50k points trained the visual representations, the fractal images with 10k points failed

Failure modes (RCDB)

We deliberately draw lines with the same color as the background



At the same time, the RCDB with broken contours failed to acquire a visual representation. The attention and accuracy were also broken from the visualization and result.

Point Cloud Pre-training with Natural 3D Structures

CVPR 2022

Ryosuke Yamada*, Hirokatsu Kataoka*, Naoya Chiba, Yukiyasu Domae*, Testuya Ogata*, ****

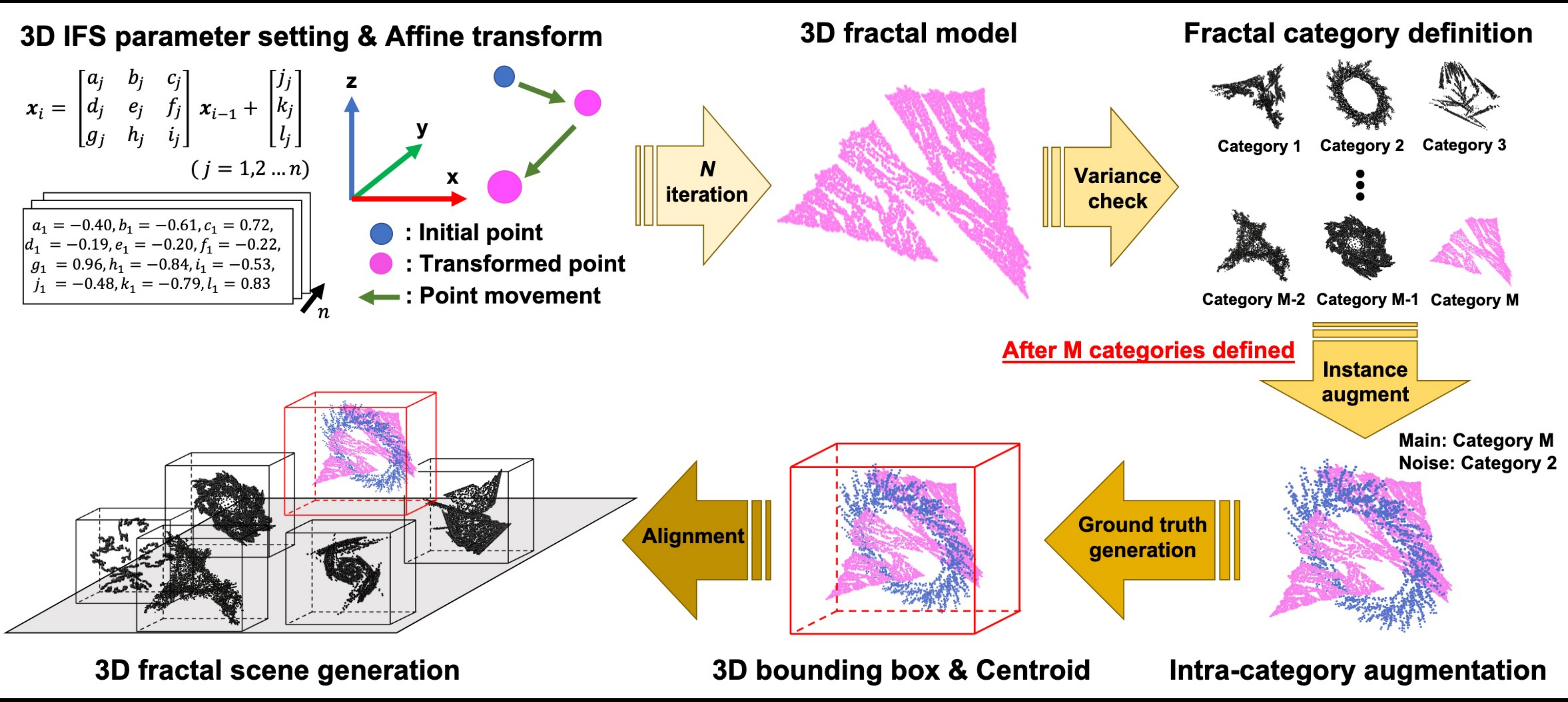
*** National Institute of Advanced Industrial Science and Technology (AIST)**
****Waseda University**

Absence of a definitive pre-training dataset in 3D domain
Higher construction costs by comparing to 2D datasets

Can we acquire a general 3D representation from a principle in our real world?

Formula-driven 3D Point Cloud Pre-training

Overview



3D IFS parameter setting & Affine transform
$x_i = \begin{bmatrix} a_j & b_j & c_j \\ d_j & e_j & f_j \\ g_j & h_j & i_j \end{bmatrix} x_{i-1} + \begin{bmatrix} j_j \\ k_j \\ l_j \end{bmatrix}$
$(j = 1,2 \ldots n)$
$a_1 = -0.40, b_1 = -0.61, c_1 = 0.72,$
$d_1 = -0.19, e_1 = -0.20, f_1 = -0.22,$
$g_1 = 0.96, h_1 = -0.84, i_1 = -0.53,$
$j_1 = -0.48, k_1 = -0.79, l_1 = 0.83$
n
z
y
x
: Initial point
: Transformed point
: Point movement
N iteration
3D fractal model
Variance check
Fractal category definition
Category 1
Category 2
Category 3
Category M-2
Category M-1
Category M
After M categories defined
Instance augment
Main: Category M
Noise: Category 2
Intra-category augmentation
Ground truth generation
3D bounding box & Centroid
Alignment
3D fractal scene generation

Point Cloud Fractal Database: 3D Fractal generation

How could we render 3D Fractal model
→ Extend the transformation matrix from 2D to 3D

3D IFS $= \{(\boldsymbol{w_j}, p_j)\}_{j=1}^{N}$ $\boldsymbol{w_j}$: Affine Transformation p_j: Selection probability

1. 3D-IFS parameters setting

$$\mathcal{W}_1 = \begin{bmatrix} 0.57 & -0.68 & 0.40 \\ -0.55 & -0.61 & -0.16 \\ -0.59 & 0.63 & 0.08 \end{bmatrix} + \begin{bmatrix} 0.18 \\ -0.22 \\ 0.50 \end{bmatrix}$$

N

3. Variance check & category definition

$$min(Var[x], Var[y], Var[z]) = \mathbf{0.17} \ldots > 0.15$$

2. Affine transformation

$$\boldsymbol{x}_i = w_j\ \boldsymbol{x}_{i-1}$$

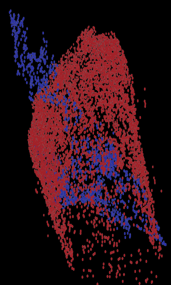
$(i = 1, 2, 3, .., n)$

$\boldsymbol{x} = [x, y, z]^T$

3D fractal model: $P = \{\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$

Instance augmentation / 3D scene generation

Mixed instance from 2 models



FractalNoiseMix

Main Category (80%)

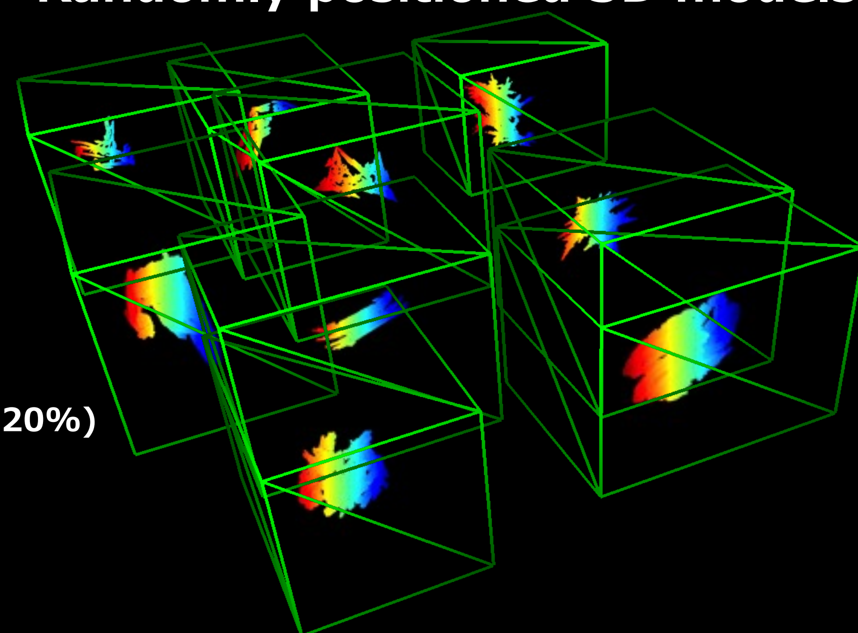
#Points: 3,200

#Points: 4,000

Noise Category(20%)

#Points: 800

Randomly positioned 3D models



Experimental results: 3D object detection in point clouds

Comparisons on ScanNetV2 / SUN RGB-D

Pre-training	Backbone	Parameter	Input	ScanNetV2 mAP@0.25	ScanNetV2 mAP@0.50	SUN RGB-D mAP@0.25	SUN RGB-D mAP@0.50
Scratch	PointNet++	0.95M	Geo + Height	57.9	32.1	57.4	32.8
Scratch	SR-UNet	38.2M	Geo	57.0	35.8	56.1	34.2
RandomRooms [51]	PointNet++	0.95M	Geo + Height	61.3	36.2	59.2	35.4
PointContrast [67]	SR-UNet	38.2M	Geo	59.2	38.0	57.5	34.8
CSC [26]	SR-UNet	38.2M	Geo	-	**39.3**	-	**36.4**
PC-FractalDB	PointNet++	0.95M	Geo + Height	**61.9**	38.3	**59.4**	33.9
PC-FractalDB	PointNet++ ×2	38.2M	Geo + Height	**63.4**	**39.9**	**60.2**	35.2
PC-FractalDB	SR-UNet	38.2M	Geo	59.4	37.0	57.1	**35.9**

Underlined bold: best score **Baseline** **Ours**

PC-FractalDB 61.9 vs 59.2 (PointContrast; ECCV 2020)
vs 61.3 (RandomRoom; ICCV 2021)

ScanNetV2 / mAP @ 0.25

Experimental results: classification PT vs. detection PT

Pre-training comparison between classification and detection
- We only add detection head in VoteNet, with PointNet++ backbone

	ScanNetV2 mAP@0.25	SUN RGB-D mAP@0.25
PointNet++	48.8	49.8
VoteNet	**61.1**	**57.6**

Detection pre-training performs much higher scores

Experimental results: self-supervision vs. formulad-supervision

Self-supervised label and formula-supervised label on PC-FractalDB
- Self-supervised label: PointContrast (ECCV 2020)
- Formula-supervised label: Fractal category (ours)

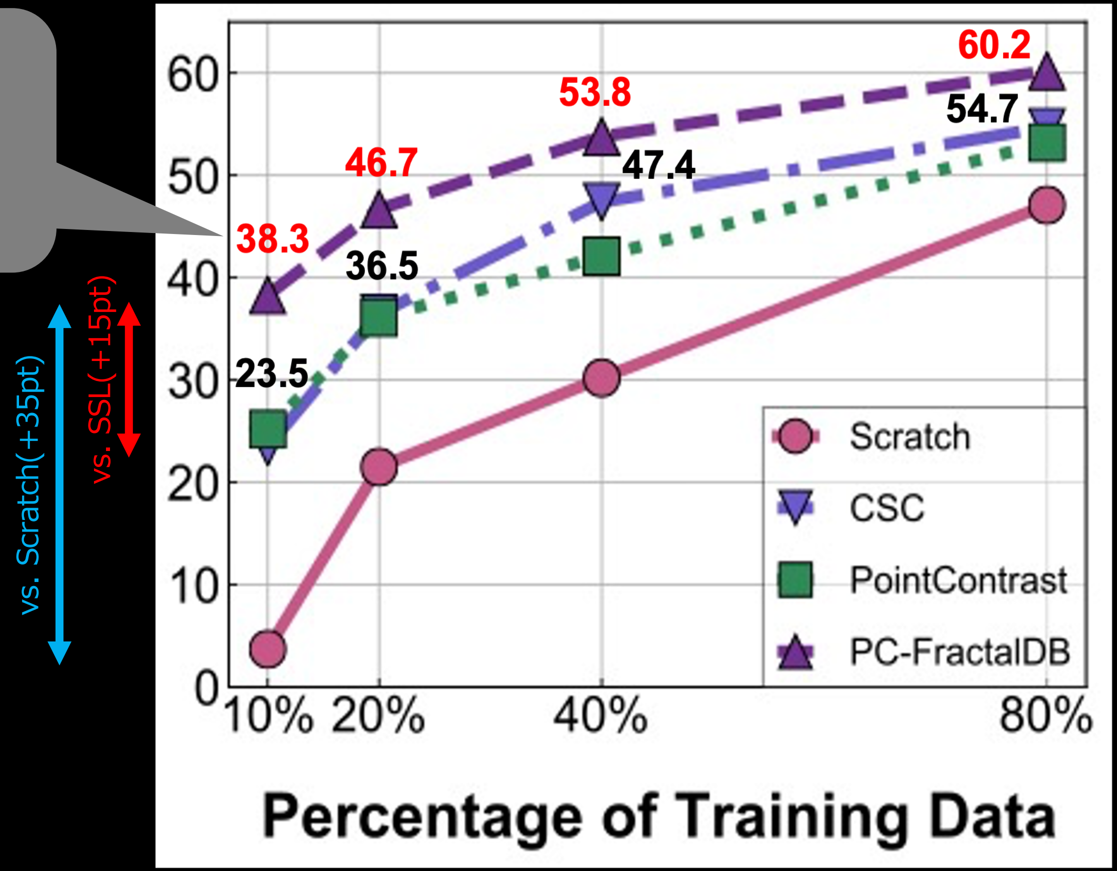
Supervisor label	ScanNetV2 mAP@0.25	SUN RGB-D mAP@0.25
PointContrast (SSL)	57.6	54.3
3D IFS (FDSL)	**59.4**	**57.1**

It is better to assign data and label from a single equation

Experimental results: Limited data/annotation

Higher accuracy on a dataset with limited data

10% amount :
+15% vs. SSL
+35% vs. from scratch



mAP@0.25

Future direction (1/3)

Towards a better pre-trained dataset

– FractalDB pre-trained model partially outperformed ImageNet-1k/Places-365 pre-trained models

– 80M Tiny Images/ ImageNet (human-related categories) withdrew public access

– We got a good feature representation without natural images

Future direction (2/3)

Different image representation from human annotated datasets

- FractalDB pre-trained model acquire a unique feature
- Steerable pre-training may be available
- Flexible dataset construction: Object detection, semantic segmentation…

Future direction (3/3)

Are fractals a good rendering formula?

– We are looking for better image patterns and their categories

– There is scope to improve the image representation and use a better rendering engine

– Any mathematical formulas, natural laws, and rendering functions can be employed to create image patterns and their image labels in the automatically created dataset

For the research community

@MIT A. Torralba Lab

Learning to See by Looking at Noise

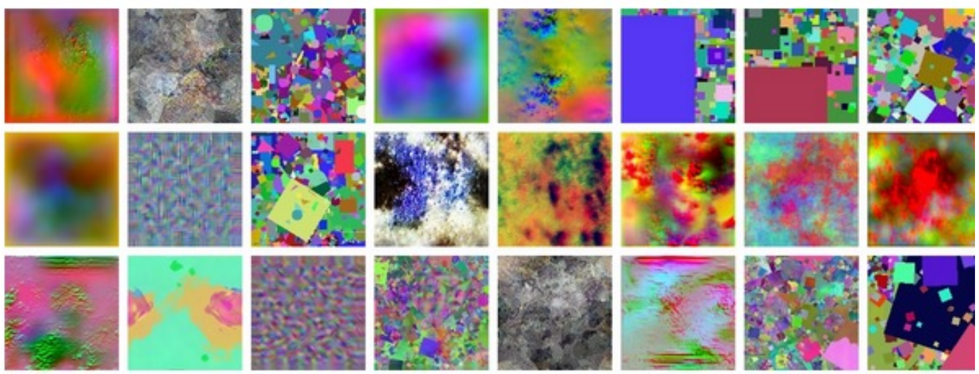
Manel Baradad*
MIT CSAIL

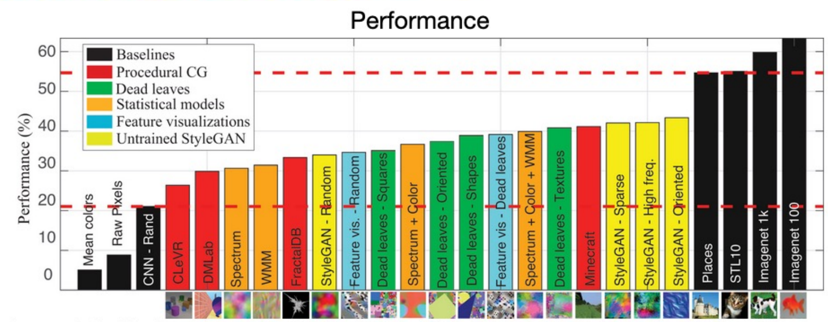
Jonas Wulff*
MIT CSAIL

Tongzhou Wang
MIT CSAIL

Phillip Isola
MIT CSAIL

Antonio Torralba
MIT CSAIL





Top-1 accuracy for the different models proposed and baselines for Imagenet-100. The horizontal axis shows generative models sorted by performance. The two dashed lines represent approximated upper and lower bounds in performance that one can expect from a system trained from samples of a generic generative image model.

[Paper] [Code] [Datasets]

https://mbaradad.github.io/learning_with_noise/

For classification on ImageNet itself, the current state-of-the-art in self-supervised learning is, of course, much higher (81.0% [68]) than our results. Yet, only a few years ago self-supervised methods reported a similar accuracy to what we report here. We therefore believe it is an open and worthwhile challenge to improve learning from noise over the next 4 years as much as self-supervised learning improved over the last 4 years.

If we could improve the FDSL, ImageNet pre-trained model may be replaced so as to protect fairness, preserve privacy, and decrease annotation labor.

Thank you.