Pre-training without Natural Images

Hirokatsu Kataoka[1], Kazushige Okayasu[1,2], Asato Matsumoto[1,3],
Eisuke Yamagata[4], Ryosuke Yamada[1,2],
Nakamasa Inoue[4], Akio Nakamura[2], Yutaka Satoh[1,3]

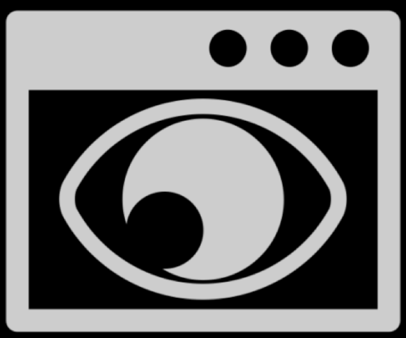
1: AIST 2: TDU 3: Univ. of Tsukuba 4: TITech

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

Annotation

FATE

Fairness, Accountability, Transparency and Ethics



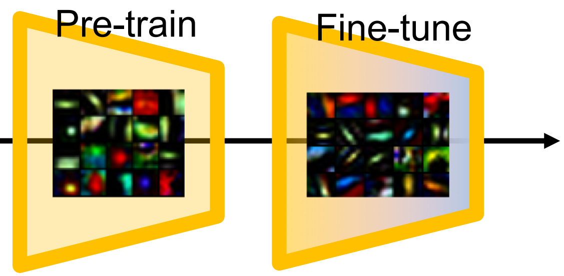
Privacy

Recent vision-driven learning

Supervised Learning

e.g. ImageNet, Places, Open Images

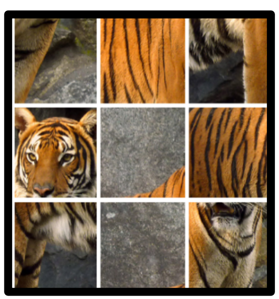
gluon-cv.mxnet.io



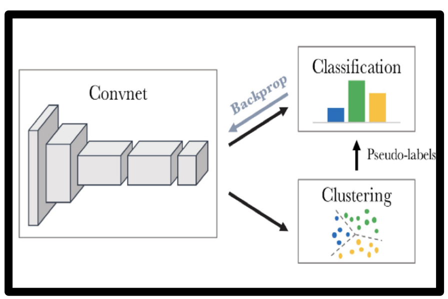
ImageNet + ResNet-50
76% @ImageNet val.

[He et al. CVPR16]

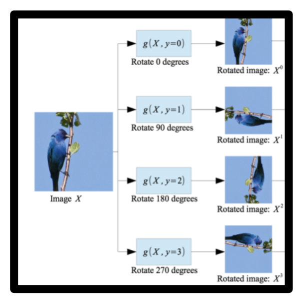
Self-supervised Learning (SSL)



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.

Can we pre-train CNN without any natural images?

Formula-driven Supervised Learning (FDSL)

- Automatically make image patterns and their labels
- With any mathematical formulas and functions

Fractal geometry from ImageNet dataset

CNN 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 natural images and human labels

Proposed method: FractalDB

FractalDB

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

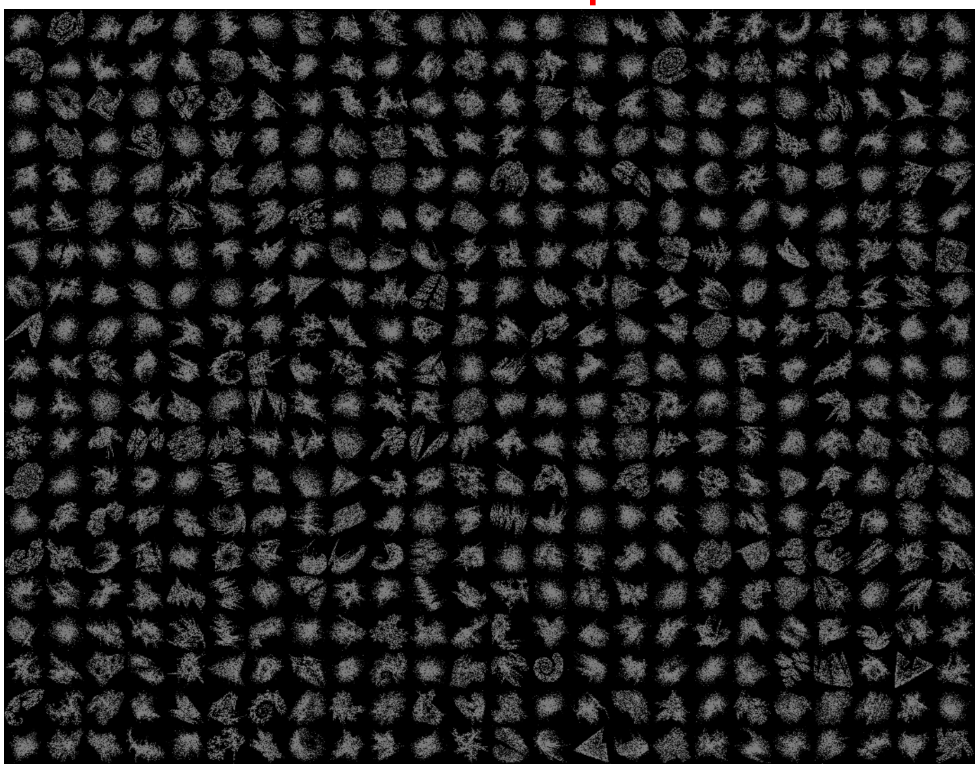
2) for a concept of Formula-driven Supervised Learning

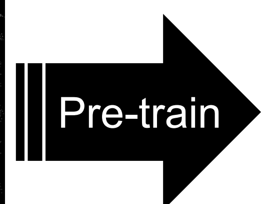
Proposed method: FractalDB

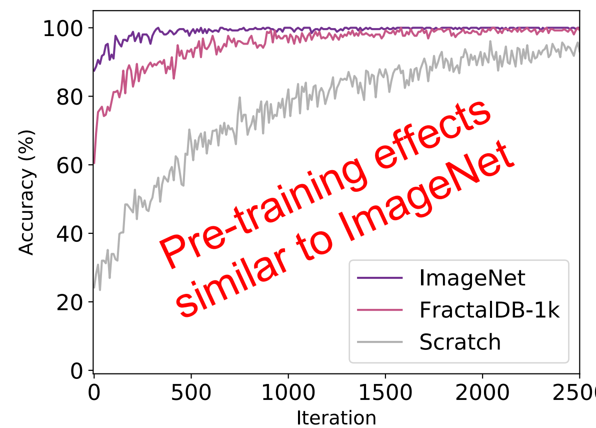
FractalDB

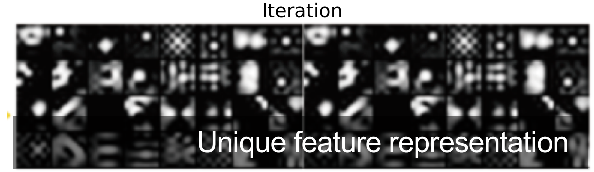
1) to make a pre-trained CNN without any natural images
2) for a concept of Formula-driven Supervised Learning

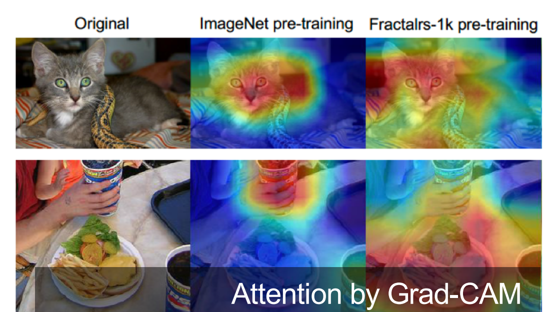
Surprising results which are similar to the effects of a supervised dataset











Fractal image rendering with Iterated Function System (IFS)

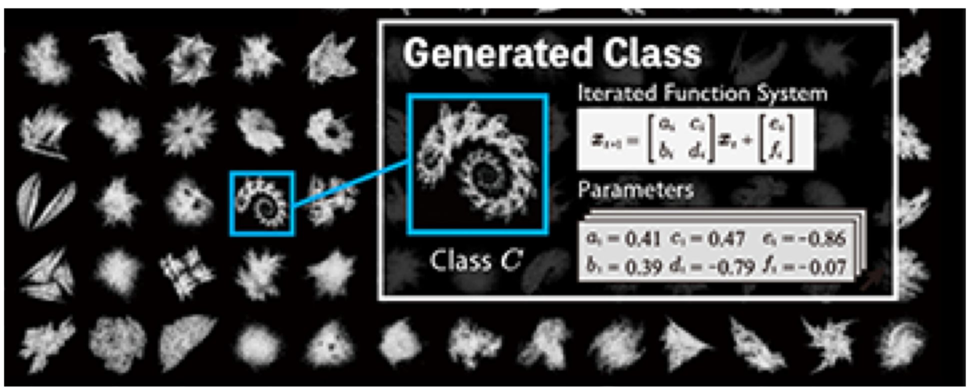
$$\mathrm{IFS} = \{\mathcal{X}; w_1, w_2, \cdots, w_N; p_1, p_2, \cdots, p_N\}$$ # Transformation probability

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

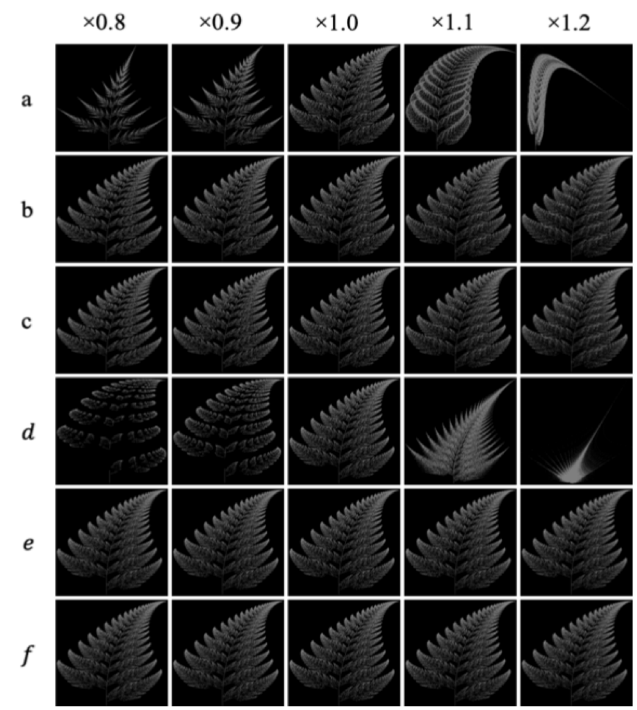
Definition of fractal category

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



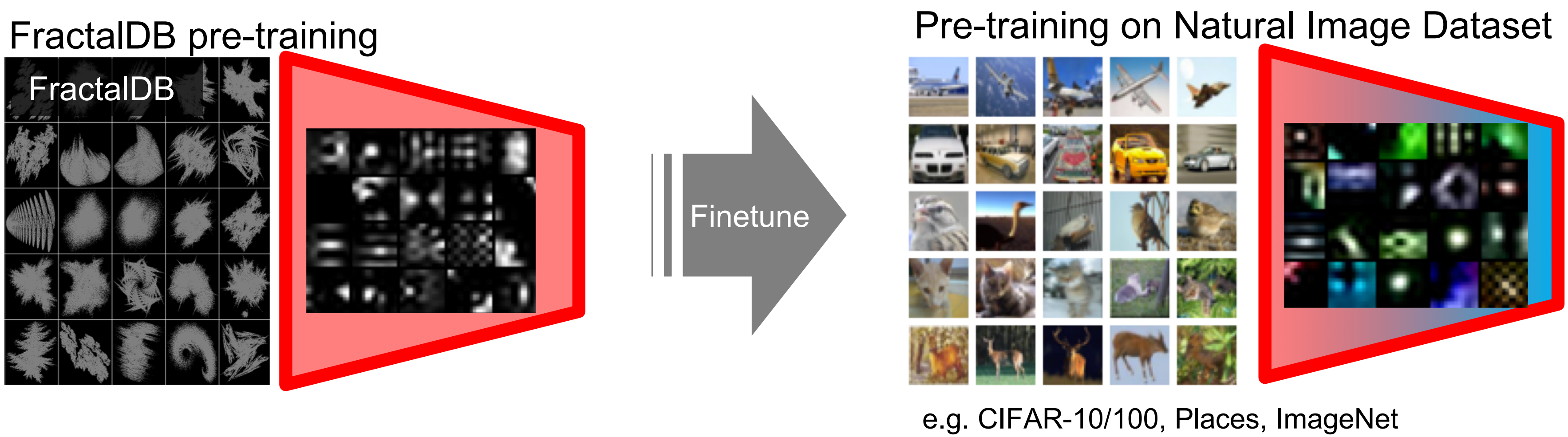
Instance augmentation in category

Experimental setting

Pre-training & Fine-tuning

– Pre-training without any natural images

– Fine-tuning in an ordinal way

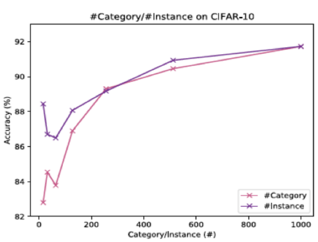


Parameters on FractalDB

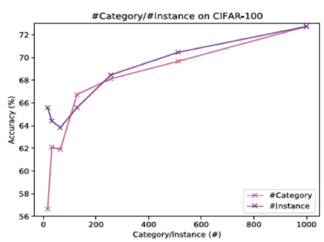
After the burden of exploration study,

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

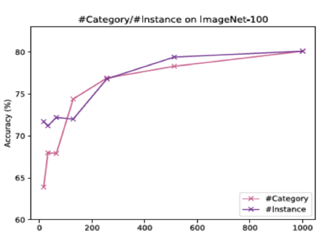
– A more difficult pre-train is slightly better in weights



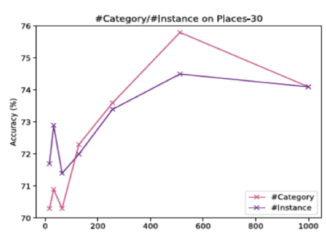
(a) CIFAR10



(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)

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

Results (1/5)

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

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

Results (1/5)

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)

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

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

Results (1/5)

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)

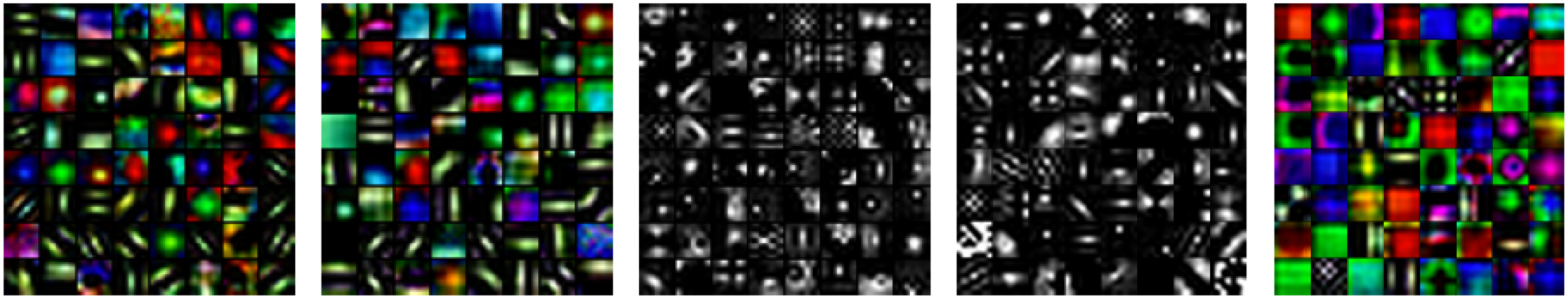
Pre-training	C10	C100	IN100	P30
Scratch	87.6	60.6	75.3	70.3
Bezier-144	87.6	62.5	72.7	73.5
Bezier-1024	89.7	68.1	73.0	73.6
Perlin-100	90.9	70.2	73.0	73.3
Perlin-1296	90.4	71.1	79.7	74.2
FractalDB-1k	**93.4**	**75.7**	**82.7**	**75.9**

We compare Formula-driven Supervised Learning with other principles

The FractalDB pre-trained model outperforms other methods

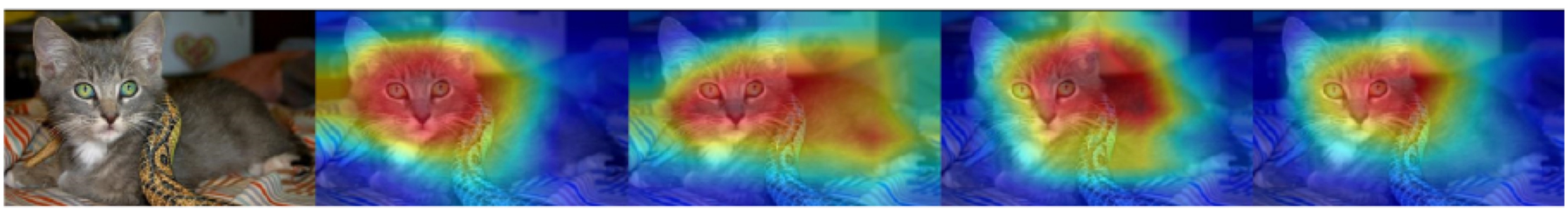
Results (5/5)

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

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 for watching.