Re-labeling ImageNet: from Single to Multi-Labels, from Global to Localized Labels

Sangdoo Yun Seong Joon Oh Byeongho Heo Dongyoon Han Junsuk Choe Sanghyuk Chun

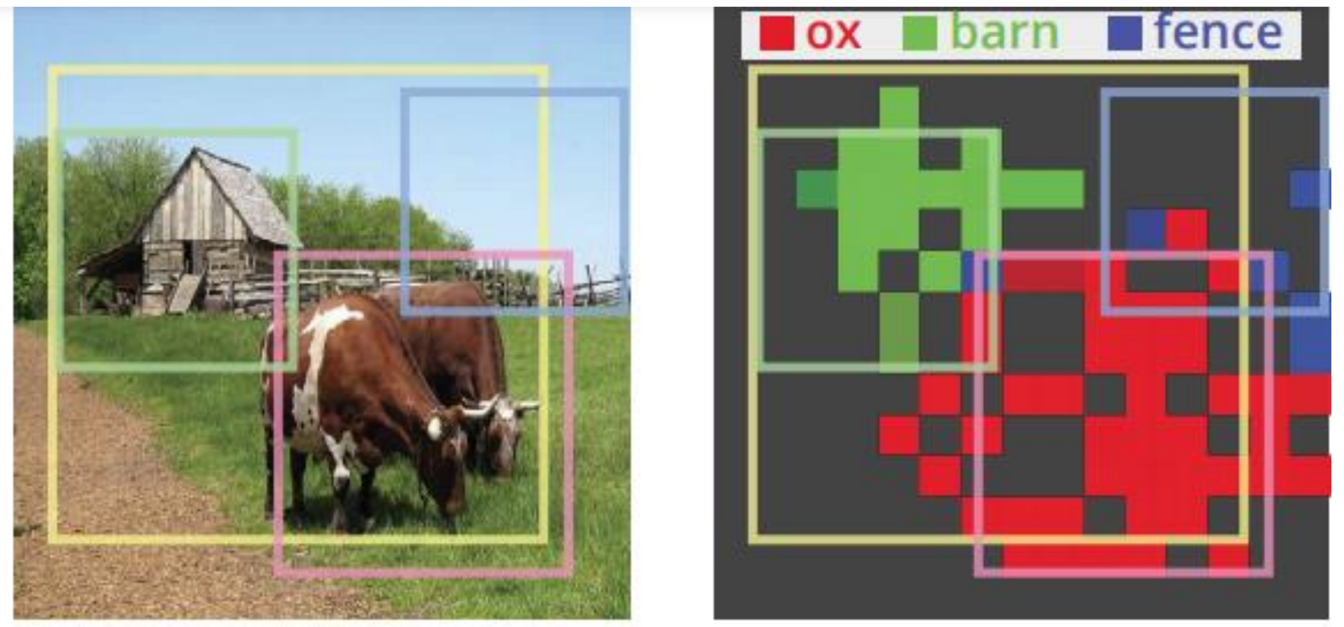
NAVER AI LAB

What's Wrong with ImageNet Labels?

- ImageNet-1K has ***single label*** annotation per image.
- However, ImageNet images often have ***multiple objects*** [a,b].
- Previous works: focus on **validation set** annotations.
- How about **training images**? – No work yet!
  - Problem: **RandomCrop** augmentation intensifies the label noise
  - Only 20% random crops have IoU > 0.5 with GT boxes.
  - → ***Localized labels*** are needed.
- **Goal**: original ***"Single"*** and ***"Global"*** ImageNet training labels → *"**Multiple**"* and *"**Localized**"* labels.

[a] Beyer et al., "Are we done with imagenet?", *arXiv*, 2020.
[b] Shankar et al., "Evaluating machine accuracy on imagenet", ICML, 2020.

Re-labeling ImageNet Training Data



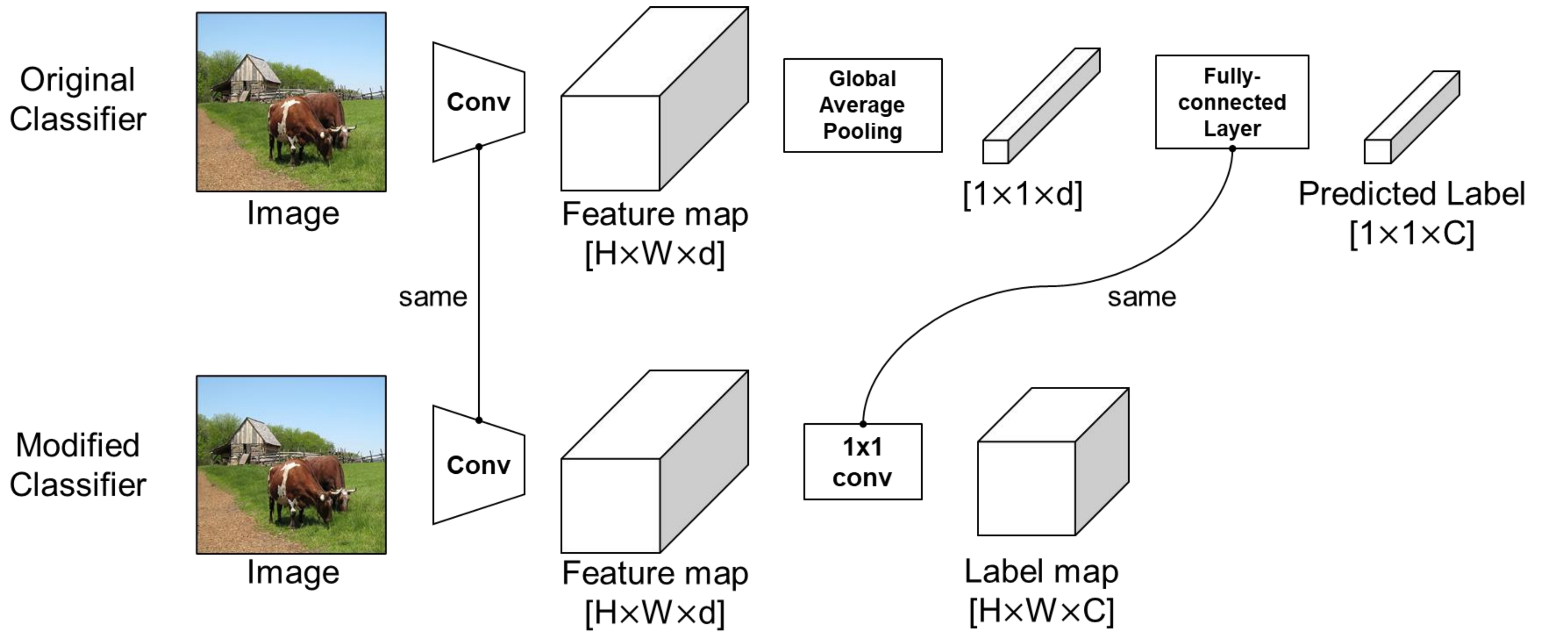
Original ImageNet label: ox 1.00 **ReLabel** annotation (label map)

ImageNet:	ox 1.00	ox 1.00	ox 1.00	ox 1.00
ReLabel:	**ox 1.00**	**barn 1.00**	**fence 0.33** **ox 0.14**	**barn 0.51** **ox 0.42**

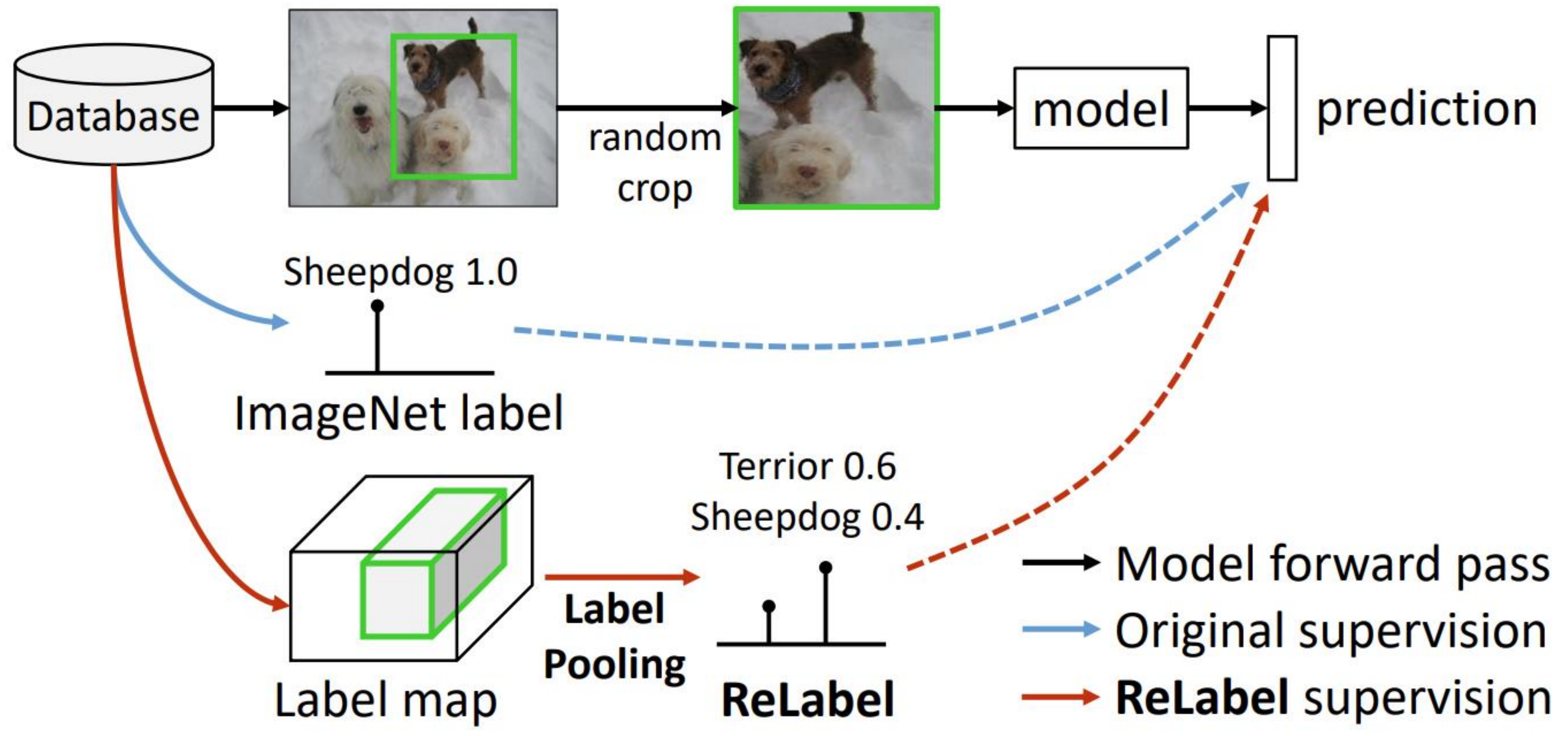
Q) How can we obtain the "dense pixel labeling" for 1.28M ImageNet training images?

Method

- We use a strong "**machine annotator**" (= a "teacher" model) trained on extra-large data to generate **localized multi-labels**.
- Keep the soft-labels of "machine annotator" → ***multi-labels***
- Without global average pooling, keep the label maps → ***localized labels***



- How to train with "***localized***" and "***multi-***" label?



- Our **ReLabel** is very **efficient!**
  - Total Label maps only need 10GB (save only top-5 logits).
  - Cheaper than KD that needs teacher's forward pass per iteration.
  - Re-usable labels (Published on GitHub).

Experiments

<ImageNet classification results>

Network	Supervision	ImageNet single-label	ImageNetV2 [36] multi-label	ReaL [2] multi-label	Shankar *et al.* [39] multi-label
ResNet-50	Original	77.5	79.0	83.6	85.3
ResNet-50	Label smoothing (ϵ=0.1) [45]	78.0	79.5	84.0	84.7
ResNet-50	Label cleaning [2]	78.1	79.1	83.6	85.2
ResNet-50	**ReLabel**	**78.9**	**80.5**	**85.0**	**86.1**

<Results on multiple architectures>

Architecture	Resources: Params	Resources: Flops	Supervision: Vanilla	Supervision: ReLabel
ResNet-18	11.7M	1.8B	71.7	72.5 (+0.8)
ResNet-50	25.6M	3.8B	77.5	78.9 (+1.4)
ResNet-101	44.7M	7.6B	78.1	80.7 (+2.6)
EfficientNet-B0	5.3M	0.4B	77.4	78.0 (+0.6)
EfficientNet-B1	7.8M	0.7B	79.2	80.3 (+1.1)
EfficientNet-B2	9.2M	1.0B	80.3	81.0 (+0.7)
EfficientNet-B3	12.2M	1.8B	81.7	82.5 (+0.8)
ReXNet ($\times$1.0)	4.8M	0.4B	77.9	78.4 (+0.5)

<Robustness benchmark>

Models	FGSM	ImageNet-A	ImageNet-C	BCG
ResNet-50	25.7	4.9	27.9	25.9
+ ReLabel	31.3 (+5.6)	7.1 (+2.2)	28.1 (+0.2)	34.6 (+8.7)
+ CutMix	42.4 (+16.7)	11.4 (+6.5)	47.5 (+19.6)	34.1 (+8.2)
+ Extra data	45.0 (+19.3)	24.8 (+19.9)	54.2 (+26.3)	36.0 (+10.1)

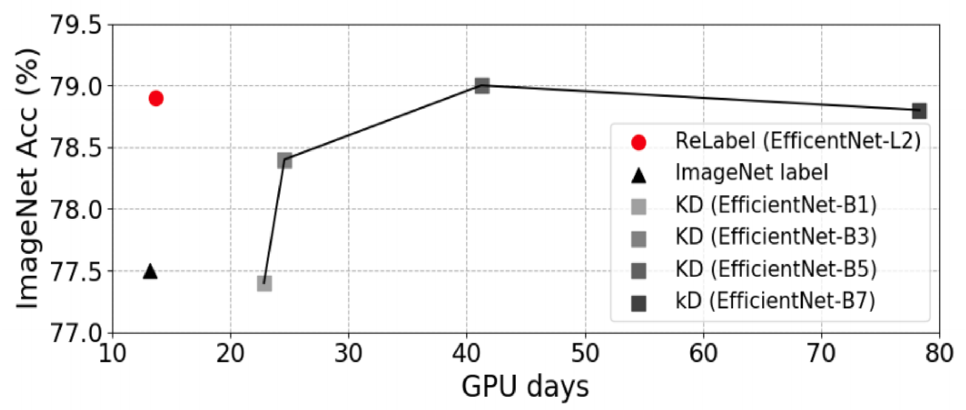
<Additional tricks towards SOTA>

Model	ImageNet top1 (%)
ResNet-50	77.5
+ **ReLabel**	78.9 (+1.4)
+ **ReLabel** + CutMix	80.2 (+2.7)
+ **ReLabel** + CutMix + Extra data	81.2 (+3.7)
ResNet-101	78.1
+ **ReLabel**	80.7 (+2.6)
+ **ReLabel** + CutMix	81.6 (+3.5)

<COCO multi-class classification>

	COCO (mAP)
ResNet-50	69.0
ResNet-50 + **ReLabel** (machine)	72.7
ResNet-50 + **ReLabel** (oracle)	73.2
ResNet-101	76.6
ResNet-101 + **ReLabel** (machine)	79.0
ResNet-101 + **ReLabel** (oracle)	80.9

<Comparison with Knowledge distillation>



<ReLabel Examples>

Conclusion

- We propose a re-labeling strategy, **ReLabel** for ImageNet training data.
- **ReLabel** improves the model performance across tasks with 3% extra computation.
- Our re-labeled ImageNet and codes: https://github.com/naver-ai/relabel_imagenet.