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

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

NAVER AI Lab

A. ReLabel Algorithm

Algorithm A1 ReLabel Pseudo-code

```
for each training iteration do
    # Load image data and label maps (assume the minibatch size is 1 for simplicity)
    input, label_map = get_minibatch(dataset)
    # Random crop augmentation
    [$c_x$,$c_y$,$c_w$,$c_h$] = get_crop_region(size(input))
    input = random_crop(input, [$c_x$,$c_y$,$c_w$,$c_h$])
    input = resize(input, [224, 224])
    # LabelPooling process
    target = RoIAlign(label_map, coords=[$c_x$,$c_y$,$c_w$,$c_h$], output_size=(1, 1))
    target = softmax(target)
    # Update model
    output = model_forward(input)
    loss = cross_entropy_loss(output, target)
    model_update(loss)
end for
```

We present the pseudo-codes of ReLabel in Algorithm A1. We assume the minibatch size is 1 for simplicity. First, an input image and its saved label map are loaded from the dataset. Then the random crop augmentation is conducted on the input image. We then perform `RoIAlign` on the label map with the random crop coordinates $[c_x,c_y,c_w,c_h]$. Finally `softmax` function is conducted on the pooled label map to get a multi-label ground-truth in $[0, 1]^C$. The multi-label ground-truth is used for updating the model with the standard cross-entropy loss.

B. Re-labeling ImageNet: Detailed Procedure and Examples

We show the detailed process of obtaining a label map in Figure A1. The original classifier takes an input image, computes the feature map ($\mathbb{R}^{H\times W\times d}$), conducts *global average pooling* ($\mathbb{R}^{1\times 1\times d}$), and generates the predicted label $L_{\text{org}} \in \mathbb{R}^{1\times 1\times C}$ with the fully-connected layer ($\mathbf{W}_{\text{fc}} \in \mathbb{R}^{d\times C}$). On the other hand, the modified classifier do not have *global average pooling* layer, and outputs a *label map* $L_{\text{ours}} \in \mathbb{R}^{H\times W\times C}$ from the feature map ($\mathbb{R}^{H\times W\times d}$). Note that the fully-connected layer ($\mathbf{W}_{\text{fc}} \in \mathbb{R}^{d\times C}$) of the original classifier and 1×1 conv ($\mathbf{W}_{\text{1x1 conv}} \in \mathbb{R}^{1\times 1\times d\times C}$) of the modified classifier are identical.

We utilize EfficientNet-L2 [15] as our machine annotator classifier whose input size is 475×475. For all training images, we resize them into 475×475 without cropping and generate label maps by feed-forwarding them. The spatial size of label map (W, H) is $(15, 15)$, number of channel d is 5504, and the number of classes C is 1000.

We present several label map examples in Figure A2. From a label map $L \in \mathbb{R}^{H\times W\times C}$, we only show two heatmaps for the classifier's top-2 categories. The heatmap is $L[c_i, :, :] \in \mathbb{R}^{H\times W}$ where c_i is one of the top-2 categories. As shown in the examples, the top-1 and top-2 heatmaps are disjointly located at each object's position.

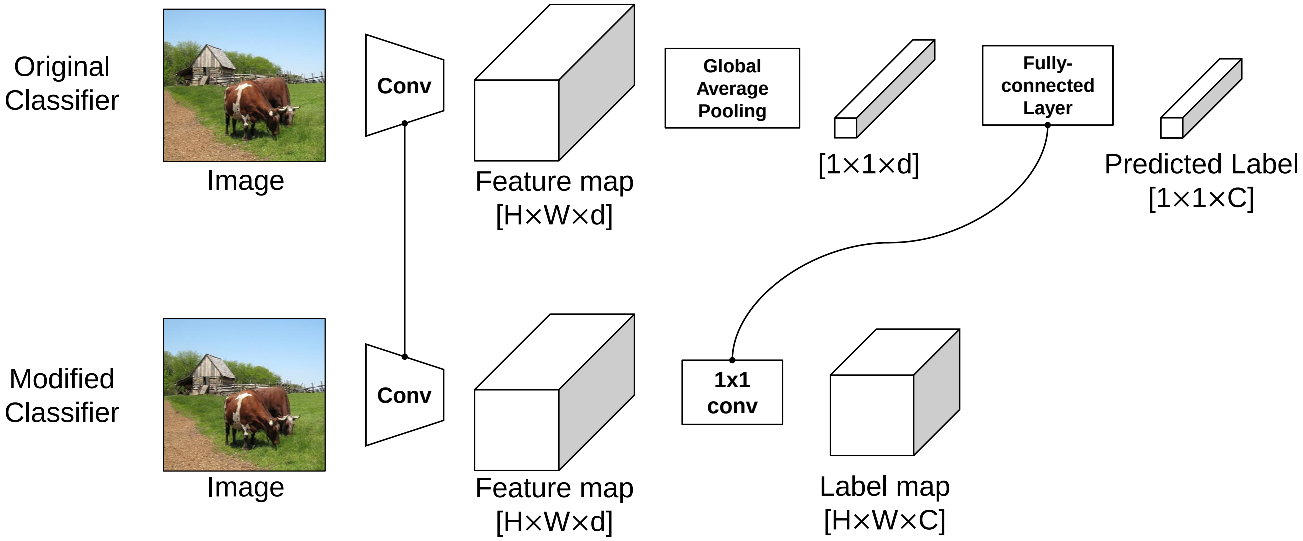

Figure A1. **Obtaining a label map.** The original classifier (upper) takes an input image and generates a predicted label $L_{\text{org}} \in \mathbb{R}^{1\times1\times C}$. On the other hand, the modified classifier (lower) outputs a *label map* $L_{\text{ours}} \in \mathbb{R}^{H\times W\times C}$ by removing the *global average pooling* layer. Note that the "Fully-connected Layer" ($\mathbf{W}_{\text{fc}} \in \mathbb{R}^{d\times C}$) of the original classifier and "1×1 conv" ($\mathbf{W}_{\text{1x1 conv}} \in \mathbb{R}^{1\times1\times d\times C}$) of the modified classifier are identical.

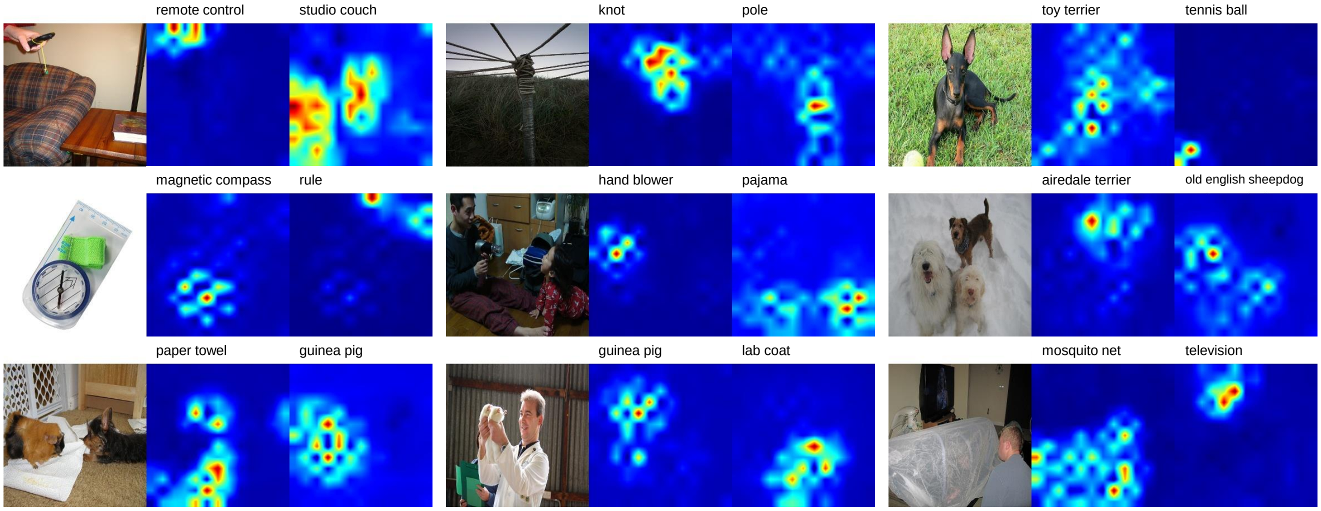

Figure A2. **Label map examples.** Each example presents the input image (left), label map of top-1 class (middle), label map of top-2 class (right).

C. Results on ImageNetV2

We present full ImageNetV2 [11] results in Table A1. Three metrics "Top-Images", "Matched Frequency", and "Threshold 0.7" are reported with two baselines Label smoothing [12] and Label cleaning [2]. ReLabel obtained 80.5, 67.3, and 76.0 accuracies on ImageNetV2 "Top-Images", "Matched Frequency", and "Threshold 0.7", where the gains are +1.5, +2.1, and +1.7 pp against the vanilla ResNet-50, respectively.

D. Implementation details

We present the implementation details in this section.

Network	Supervision	ImageNet	ImageNetV2 (Top-Images)	ImageNetV2 (Matched Frequency)	ImageNetV2 (Threshold 0.7)
ResNet-50	Original	77.5	79.0	65.2	74.3
ResNet-50	Label smoothing (ϵ=0.1) [12]	78.0	79.5	66.0	74.6
ResNet-50	Label cleaning [2]	78.1	79.1	64.9	73.9
ResNet-50	**ReLabel**	**78.9**	**80.5**	**67.3**	**76.0**

Table A1. **ImageNetV2 results.** We report performances on ImageNetV2 [11] metrics.

D.1. Training Hyper-parameters

In most experiments, we have trained the models with SGD optimizer with learning rate 0.1 and weight decay 0.0001. For further improved performance, we utilized AdamP [6] optimizer with learning rate 0.002, and weight decay 0.01 when applying additional tricks such as CutMix regularizer or extra training data (ImageNet-21K).

D.2. EfficientNet on ImageNet

We utilize an open-source pytorch codebase [8] to train EfficientNet variants on ImageNet. We utilize AdamP [6] optimizer and set training epochs 400, minibatch size 512, learning rate 0.002, and weight decay 0.01 with four NVIDIA V100 GPUs. Dropout and drop path [9] regularizers are used with dropout rate 0.2 and drop path rate 0.2, respectively. We also utilize Random erasing [18], RandAugment [4], and Mixup [17] augmentations as suggested in [8]. All training settings are samely used for both vanilla training and ReLabel training of EfficientNet variants.

D.3. Knowledge Distillation

Training with knowledge distillation is also conducted on the pytorch codebase [8]. For teacher network, we use official EfficientNet (B1-B7) [13] weights trained with noisy student [15] techniques. We utilize outputs of networks after soft-max layer and the cross-entropy loss between teacher and students is only used for the distillation loss [7]. The temperature and cross-entropy with ground truth were not used. Since the EfficientNet teachers are trained with large-size images ($240 \times 240 - 600 \times 600$), we put the large-size image for teacher network and resize it to 224×224 for inputs of student network. We adopt SGD with Nesterov momentum for the optimizer and the standard setting [5] with long epochs: learning rate 0.1, weight decay 10^{-4}, batch size 256, training epochs 300 and cosine learning rate schedule with four NVIDIA V100 GPUs.

D.4. COCO Multi-label Classification

As in recent multi-label classification works [3, 14, 16, 1], the classifier model is initialized with ImageNet-pretrained model and fine-tuned on COCO multi-label dataset [10]. We utilize the official pytorch ImageNet-pretrained model using torchvision toolbox[1]. The weight of the final fully-connected layer is modified from $\mathbb{R}^{d\times1000}$ to $\mathbb{R}^{d\times80}$ to fit the number of classes for COCO dataset and the weight matrix is randomly initialized. For fine-tuning, we utilize AdamP [6] optimizer and cosine learning rate schedule with initial learning rate 0.0002 and weight decay 0.01. We set the minibatch size to 128. The input resolution is 224×224 for ResNet-50 and 448×448 for for ResNet-101. To obtain machine-generated label maps, we utilize a pre-trained TResNet-XL model [1] whose input size is 640×640 and mAP is 88.4%.

E. ReLabel Examples on ImageNet

We present ReLabel exsamples on ImageNet training data in Figure A3. We show the full training images (left) and the random cropped patches (right). The random crop coordinates are denoted by blue bounding boxes. We also present the original ImageNet label and the new multi-labels by ReLabel. As shown in the examples, ReLabel can generate location-specific multi-labels with more precise supervision than the original ImageNet label.

References

[1] Emanuel Ben-Baruch, Tal Ridnik, Nadav Zamir, Asaf Noy, Itamar Friedman, Matan Protter, and Lihi Zelnik-Manor. Asymmetric loss for multi-label classification. *arXiv preprint arXiv:2009.14119*, 2020. 3

[1] https://github.com/pytorch/vision

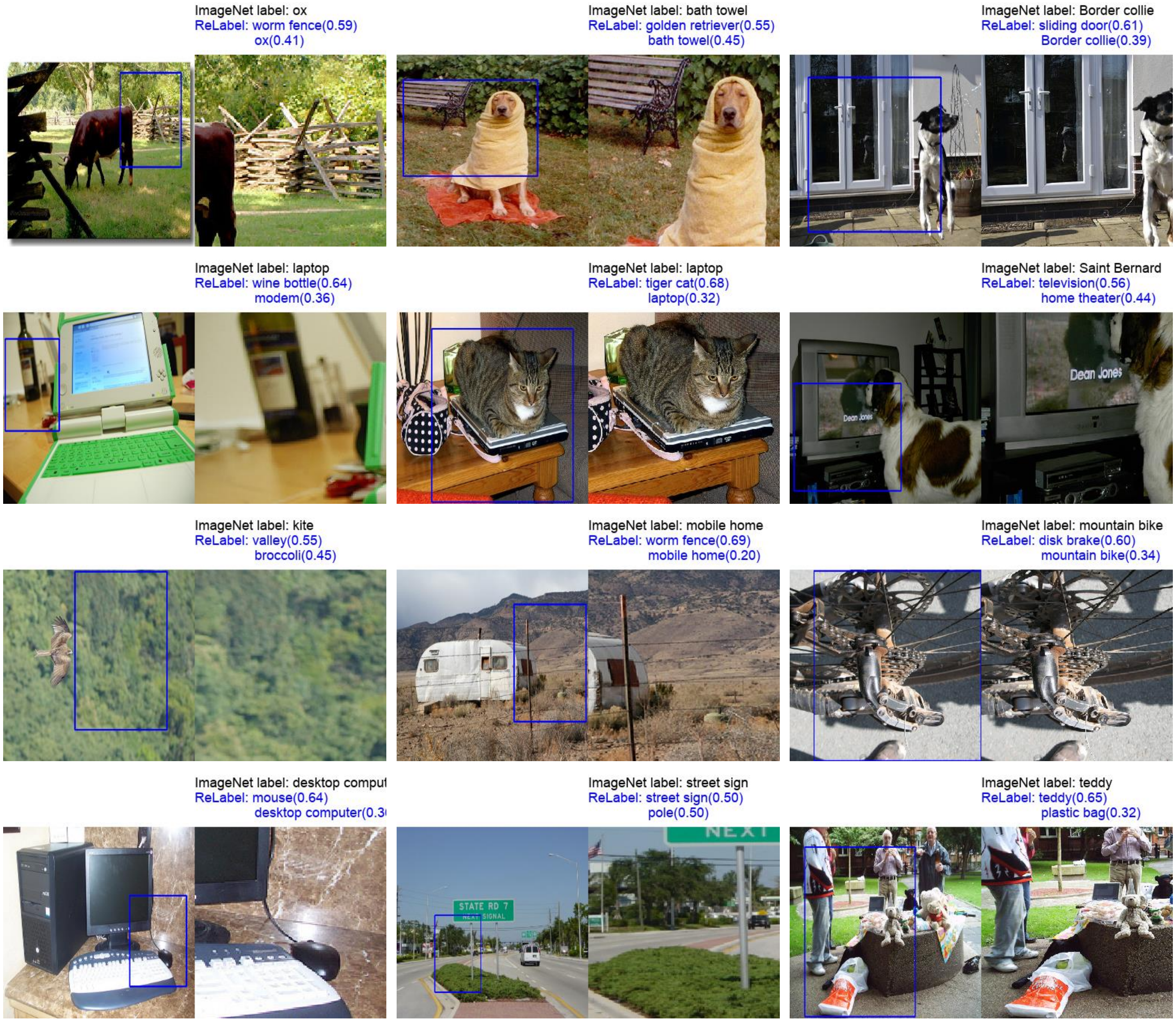
ImageNet label: ox
ReLabel: worm fence(0.59)
ox(0.41)
ImageNet label: bath towel
ReLabel: golden retriever(0.55)
bath towel(0.45)
ImageNet label: Border collie
ReLabel: sliding door(0.61)
Border collie(0.39)
ImageNet label: laptop
ReLabel: wine bottle(0.64)
modem(0.36)
ImageNet label: laptop
ReLabel: tiger cat(0.68)
laptop(0.32)
ImageNet label: Saint Bernard
ReLabel: television(0.56)
home theater(0.44)
Dean Jones
ImageNet label: kite
ReLabel: valley(0.55)
broccoli(0.45)
ImageNet label: mobile home
ReLabel: worm fence(0.69)
mobile home(0.20)
ImageNet label: mountain bike
ReLabel: disk brake(0.60)
mountain bike(0.34)
ImageNet label: desktop comput
ReLabel: mouse(0.64)
desktop computer(0.3
ImageNet label: street sign
ReLabel: street sign(0.50)
pole(0.50)
STATE RD 7
NEXT
ImageNet label: teddy
ReLabel: teddy(0.65)
plastic bag(0.32)

Figure A3. **ReLabel examples during ImageNet training.** We present selected examples generated by ReLabel during ImageNet training. For each example, the left image is the full training image and the right image is the random cropped patch. The random crop coordinates are denoted by blue bounding boxes. The original ImageNet label and ReLabel are also presented.

[2] Lucas Beyer, Olivier J Hénaff, Alexander Kolesnikov, Xiaohua Zhai, and Aäron van den Oord. Are we done with imagenet? *arXiv preprint arXiv:2006.07159*, 2020. 2, 3

[3] Zhao-Min Chen, Xiu-Shen Wei, Peng Wang, and Yanwen Guo. Multi-label image recognition with graph convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5177–5186, 2019. 3

[4] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 702–703, 2020. 3

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 3

[6] Byeongho Heo, Sanghyuk Chun, Seong Joon Oh, Dongyoon Han, Sangdoo Yun, Youngjung Uh, and Jung-Woo Ha. Slowing down the weight norm increase in momentum-based optimizers. *arXiv preprint arXiv:2006.08217*, 2020. 3

[7] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 3

[8] https://github.com/rwightman/pytorch-image models. *Pytorch Image Models*. 3

[9] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Weinberger. Deep networks with stochastic depth. In *ECCV*, 2016. 3

[10] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 3

[11] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *International Conference on Machine Learning*, pages 5389–5400, 2019. 2, 3

[12] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016. 2, 3

[13] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114, 2019. 3

[14] Zhouxia Wang, Tianshui Chen, Guanbin Li, Ruijia Xu, and Liang Lin. Multi-label image recognition by recurrently discovering attentional regions. In *Proceedings of the IEEE international conference on computer vision*, pages 464–472, 2017. 3

[15] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10687–10698, 2020. 1, 3

[16] Renchun You, Zhiyao Guo, Lei Cui, Xiang Long, Yingze Bao, and Shilei Wen. Cross-modality attention with semantic graph embedding for multi-label classification. In *AAAI*, pages 12709–12716, 2020. 3

[17] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 3

[18] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. *arXiv preprint arXiv:1708.04896*, 2017. 3