A Application: Training on CIFAR-10 with pre-trained ExRep

In this section, we present the results of additional experiments ran on CIFAR-10 using PyramidNet [42], which outperforms EfficientNet on CIFAR-10 in general.

Table 4. Classification performance on CIFAR-10. PyramidNet† indicates our implementation of the PyramidNet model without the use of extra training data and PyramidNet ExRep is the randomly initialized PyramidNet with the input distiller pre-trained on ImageNet. The best results are in bold.

Method	Dataset	Size	Top-1 Accuracy	Top-5 Accuracy
PyramidNet† [42]	CIFAR-10	32	96.24%	98.75%
PyramidNet ExRep (Ours)	CIFAR-10	32	**96.57%**	**99.00%**

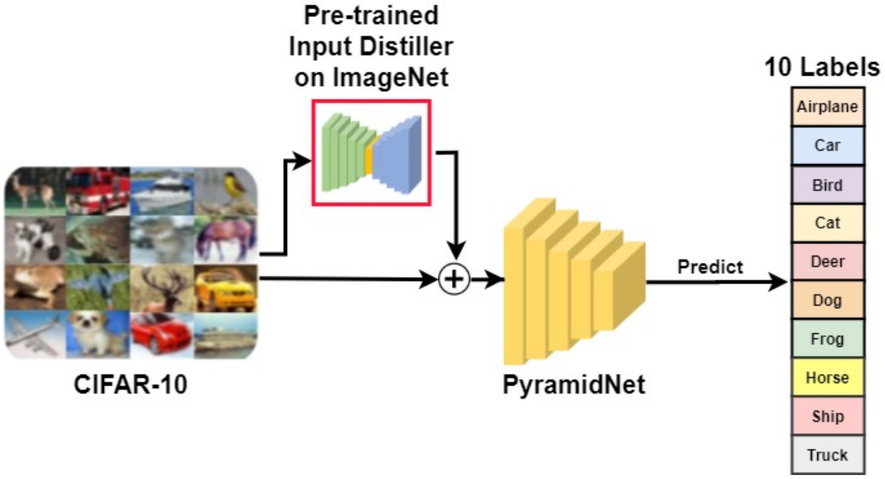

Fig. 4. Training of ExRep on CIFAR-10. The input distiller (red box) is pre-trained on ImageNet. The output of the input distiller is added to the CIFAR-10 images following stochastic sampling as explained in the *Training on CIFAR-10* subsection below.

Dataset description. We use PyTorch to run the experiments on CIFAR-10 [12]. CIFAR-10 contains 60,000 color images of 32×32 labeled as one of 10 categories. We use 50,000 images for training and the remaining 10,000 for evaluation.

Input distiller pre-trained on ImageNet. The classification performance of our pre-trained ExRep on CIFAR-10 is shown in Table 4. The input distiller is trained on ImageNet images of size 224×224. We apply bicubic interpolation to normalize the input image sizes.

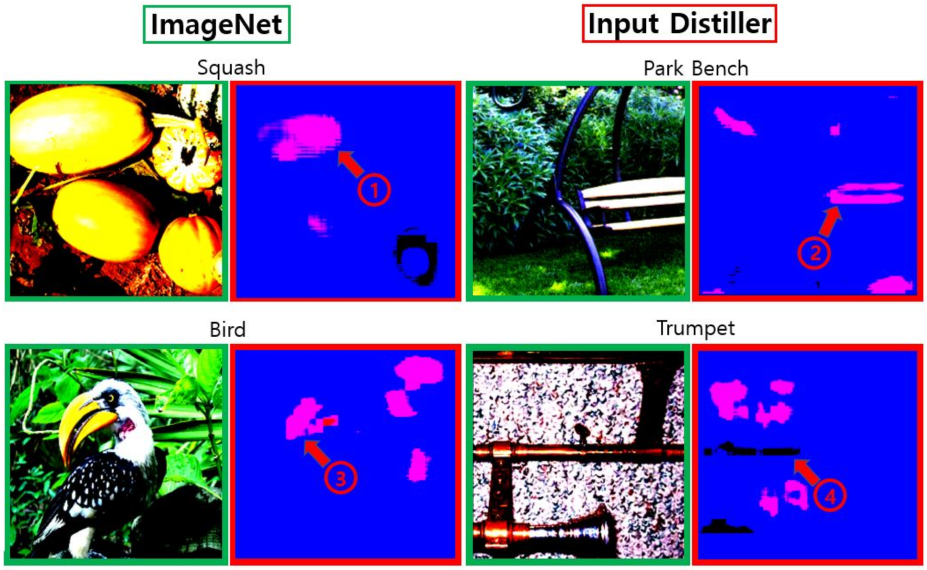

Fig. 5. Visualization of the input distiller outputs using ExRep with EfficientNet-B1 and 240×240 input image size. The images in the green border are original ImageNet images, while those in the red border are outputs generated by the input distiller. Indicated by red circles numbered 1 through 4, the respective salient regions of squashes, the park bench, the bird, and the trumpet are clearly highlighted in purple or black, directly relating the images to their corresponding class labels.

Training on CIFAR-10. PyramidNet [42] represents a randomly initialized model trained on CIFAR-10 without using our pre-trained ExRep. We set α to 200, the depth to 272, and use pyramidal bottleneck layers in PyramidNet. Figure 4 depicts the pipeline of our proposed model trained on CIFAR-10. We apply stochastic sampling to transform CIFAR-10 images into a form compatible to that of the output of the input distiller for addition. We select random numbers by sampling from the normal distribution and exclude those that are greater than γ, set to 0.01, to alleviate the computational cost. To feed the CIFAR-10 images to the input distiller in our model, we resize them to match the size of ImageNet images through bicubic interpolation. After the image processing by the input distiller, we reapply bicubic interpolation to obtain the original CIFAR-10 image size. Details of the input distiller transformation is described in Section 3.1.

Evaluation on CIFAR-10. The results on CIFAR-10 are presented in Table 4. Note that the models in our experiments are trained without using any augmentation or regularization methods, such as CutMix [39], AugMix [40], or ShakeDrop [41]. Our PyramidNet ExRep achieves a 0.33% increase in top-1 accuracy from 96.24% to 96.57%. Accordingly, we demonstrate that our ExRep can be extended to other datasets and achieve a better performance on them as well.

B Qualitative Analysis

Visualization of the input distiller samples. We analyze the outputs generated by the input distiller from the ImageNet input images. Note that these outputs are added to the original ImageNet images (Eq. 1). Some ImageNet image samples and their corresponding outputs generated by the input distiller are shown in Fig. 5. We can observe that the input distiller can focus on salient regions (highlighted in purple and black) of the image, such as the peel of the squash, the wood forming the back of the park bench, the beak of the bird, and the metal structure of the trumpet, giving extra representation to their corresponding class labels. Combined with the original input images, the outputs of the input distiller can thus provide additional representational information, producing a similar effect as using extra training data.

C Structure of Input Distiller

Table 5. Input distiller network. Conv, IN, ResBlock denote the convolution layer, Instance Normalization, and the residual block. Resblock consists of two series of Conv3×3, IN, and LeakyReLU. We set the input size to 240×240 and the dimension of maximum Conv to 256.

Block i	Operator $\mathcal{F}_i$	Resolution $H_i \times W_i$	# Channels C_i	# Layers L_i
1	Conv3×3	240×240	68	1
2	Encoder-IN-ResBlock	120×120	136	5
3	Encoder-IN-ResBlock	60×60	256	5
4	Encoder-IN-ResBlock	30×30	256	5
5	Decoder-IN-ResBlock	60×60	136	5
6	Decoder-IN-ResBlock	120×120	136	5
7	Decoder-IN-ResBlock	240×240	68	5
8	IN & LeakyReLU & Conv1×1	240×240	3	2

D Training Details

We train our ExRep under the following setting: SGD optimizer with momentum 0.9, weight decay 0.0001, Batch Normalization with momentum 0.99, and initial learning rate 0.00256, which decays by 0.97 per 2.4 epochs. We use the default batch size of 256 and reduce the batch size, when we can no longer fit the model into our GPU memory. We also use dropout [37] with a dropout probability 0.2 and stochastic depth [38] with a drop connect probability 0.2. We perform data augmentation under the following setting: color jittering with parameter 0.4, random erase regions as the constant 0 with count 1 and probability 0.2. We also apply label smoothing on softmax function as in [3].

For the input distiller, the dimension of maximum Conv is set to 256, and the feature dimension of both the teacher and student models is set to 1280. The same setting is used for both EfficientNet-B0 and B1. For the CRD loss, we set the linear transform dimension to 128, N (the number of negative samples) to 4096, T (temperature) to 0.07, and M (cardinality) to 0.5; we set the balancing factors λ_{CE}, λ_{KL}, λ_{CRD}, and λ_{Critic} to 1.0, 1.0, 0.8, and 0.1, respectively.

E L^2-Starting Point Loss

Table 6. Performance results on L^2-SP loss. This experiment is conducted on ImageNet, and the base model is EfficientNet-B0.

Method	Base Model	Size	Top-1 ACC	Top-5 ACC
ExRep (CE+KD+CRD+Adv)	EfficientNet-B0	224	77.9%	93.6%
ExRep (CE+KD+L^2-SP+Adv)	EfficientNet-B0	224	78.3%	94.1%

L^2-Starting Point (L^2-SP) is introduced in the transfer learning and domain adaptation area as starting point as the reference (SPAR) [43]. The authors use L^2 regularization as a starting point to preserve the knowledge of the teacher network pre-trained on the source dataset, while training the student network. This L^2 SP is also utilized in many transfer learning methods [44,45]. We experiment using this L^2-SP loss instead of the CRD loss.

CRD loss vs. L^2-SP loss. We can see in Table 6 that the use of L^2-SP results in a slightly better performance than CRD. However, to use the L^2-SP loss, the teacher and student networks must be of the architecture in terms of size, while CRD has no such limitation. Due to this advantage, we choose the CRD loss for our proposed method, but the L^2-SP loss may also be a good option.

References

39. Yun, S., Han, D., Oh, S.J., Chun, S., Choe, J., Yoo, Y.: Cutmix: Regularization strategy to train strong classifiers with localizable features. In: Proceedings of the IEEE International Conference on Computer Vision. (2019) 6023–6032
40. Hendrycks, D., Mu, N., Cubuk, E.D., Zoph, B., Gilmer, J., Lakshminarayanan, B.: Augmix: A simple data processing method to improve robustness and uncertainty. arXiv preprint arXiv:1912.02781 (2019)
41. Yamada, Y., Iwamura, M., Akiba, T., Kise, K.: Shakedrop regularization for deep residual learning. IEEE Access **7** (2019) 186126–186136
42. Han, D., Kim, J., Kim, J.: Deep pyramidal residual networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2017) 5927–5935
43. Li, X., Grandvalet, Y., Davoine, F.: Explicit inductive bias for transfer learning with convolutional networks. arXiv preprint arXiv:1802.01483 (2018)
44. Li, X., Xiong, H., Wang, H., Rao, Y., Liu, L., Huan, J.: Delta: Deep learning transfer using feature map with attention for convolutional networks. arXiv preprint arXiv:1901.09229 (2019)
45. Jeon, H., Bang, Y., Kim, J., Woo, S.S.: T-gd: Transferable gan-generated images detection framework. arXiv preprint arXiv:2008.04115 (2020)