Supplementary Material for "Resolution Switchable Networks for Runtime Efficient Image Recognition"

Yikai Wang[1⋆], Fuchun Sun[1], Duo Li[2], and Anbang Yao[2]

[1] Beijing National Research Center for Information Science and Technology (BNRist), State Key Lab on Intelligent Technology and Systems, Department of Computer Science and Technology, Tsinghua University
[2] Cognitive Computing Laboratory, Intel Labs China
{wangyk17@mails.,fcsun@}tsinghua.edu.cn, {duo.li,anbang.yao}@intel.com

1 Details of the Ensemble

As the importance scores $\boldsymbol{\alpha}$ for ensemble are essential to the MRED, we study their values w.r.t. different resolutions. We observe that the learned values of $\boldsymbol{\alpha}$ stay almost unchanged in the last three epochs. The final results w.r.t. the resolutions from 224×224 to 96×96 are $\{0.37, 0.29, 0.20, 0.12, 0.02\}$ for ResNet18, $\{0.32, 0.30, 0.23, 0.13, 0.02\}$ for ResNet50, and $\{0.41, 0.30, 0.19, 0.09, 0.01\}$ for MobileNetV2. In each network, the score w.r.t. the resolution 224×224 has the largest ratio, and the ratio decreases with the decrease of resolutions.

2 Visualization of BNs.

Fig. 1 visualizes BN parameters, including scale $\boldsymbol{\gamma}$, bias $\boldsymbol{\beta}$, mean $\boldsymbol{\mu}$ and variance $\boldsymbol{\sigma}$, in a parallel trained model on ResNet18. There are eight blocks and each has two Conv layers. We plot the channel-wise means of BN parameters of every first layer in the left four sub-figures and of every second layer in the right four sub-figures. We observe that BN parameters are likely to be arranged in the ascending order or the descending order of image resolutions.

3 Extension to Semantic Segmentation

Besides the experiments described in the main paper, we also apply our method to semantic segmentation to further validate its generalization ability to handle other visual recognition tasks beyond classification. We choose RefineNet [4], a typical semantic segmentation model which achieves state-of-the-art results on dataset NYUDv2 [5]. Following the original setting in [4], we use ResNet101 as the backbone network. A schematic framework for training a RS-Net for semantic segmentation is illustrated in Figure 2. During training, each logit outputted

⋆ This work was done when Yikai Wang was an intern at Intel Labs China, supervised by Anbang Yao who is responsible for correspondence.

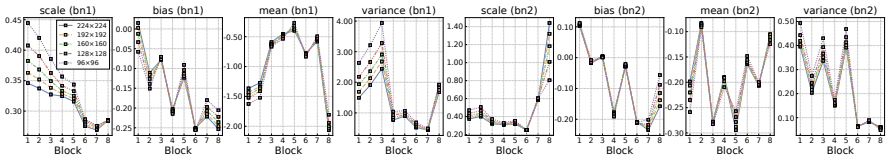

Fig. 1. BN parameters and statistics in ResNet18 blocks. The first layer of each block is shown in the left four sub-figures, and the second is shown in the right four sub-figures.

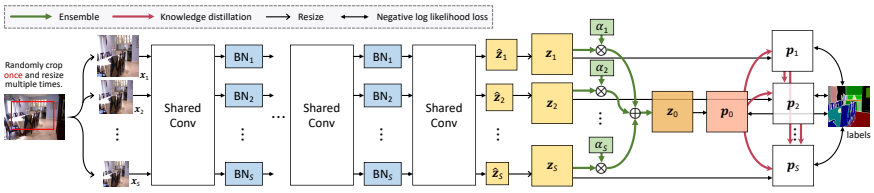

Fig. 2. Framework of training a RS-Net for semantic segmentation. Logits outputted by the last Conv layer are denoted as $\hat{\boldsymbol{z}}_1, \hat{\boldsymbol{z}}_2, \cdots, \hat{\boldsymbol{z}}_S$. We resize these logits to the same resolution of $\boldsymbol{x}_1$, which is the largest input resolution. We denote the resized logits as $\boldsymbol{z}_1, \boldsymbol{z}_2, \cdots, \boldsymbol{z}_S$. The ensemble logit $\boldsymbol{z}_0$ is learned as a weighted mean of the resized logits. During testing, each logit $\hat{\boldsymbol{z}}_s, s \in \{1, 2, \cdots, S\}$ is uniformly resized to the original image resolution for evaluation.

by the last Conv layer, denoted as $\hat{\boldsymbol{z}}_s, s \in \{1, 2, \cdots, S\}$, does not has the same resolution with its corresponding input $\boldsymbol{x}_s$. For example, if we choose the multi-resolution setting as $\mathbb{S} = \{352 \times 352, 224 \times 224, 96 \times 96\}$, resolutions of $\hat{\boldsymbol{z}}_1, \hat{\boldsymbol{z}}_2, \hat{\boldsymbol{z}}_3$ will be $88 \times 88, 56 \times 56, 24 \times 24$ respectively. We uniformly resize $\hat{\boldsymbol{z}}_s$ to the largest input resolution (for this example is 352×352) before the ensemble distillation process and calculating losses with labels. We do not use left-right flips or the multi-scale technique during testing for additional performance promotion, and each logit $\hat{\boldsymbol{z}}_s, s \in \{1, 2, \cdots, S\}$ is uniformly resized to the original image resolution before calculating evaluation metrics.

Results on NYUDv2 are shown in Table 1. Following [4], we train on RGB images with 40 classes, using the standard training and testing split with 795 and 654 images respectively. These results verify that our method can be applied to the semantic segmentation task, maintaining the resolution switchable ability while simultaneously improves performance. As far as we know, this is the first resolution switching attempt for semantic segmentation, realizing a selectable inference speed which is beneficial to efficient runtime model deployments. As we can see in Table 1, RS-Net especially achieves performance gains over I-Nets at low resolutions, e.g., with a significant IoU gain 14.0 at 96×96.

In Fig. 3, we compare our RS-Net with an individual model which is trained at 352×352. We evaluate performance at three resolutions during inference, for proving that our model has better robustness against various resolutions. Predictions w.r.t. 224×224 and 96×96 indicate that downsizing input resolution leads to quick performance drops for an individual model. In contrast, our RS-Net has milder performance drops toward downsizing the resolution.

Table 1. Results comparison for semantic segmentation based on RefineNet with ResNet101 as the backbone. We report pixel accuracies (%), mean accuracies (%) and IoU of individual models (I-Nets) and our RS-Net. Note that no left-right flips or multi-scale testing is performed. All experiments use the same data pre-processing methods and training settings.

Resolution	I-Nets (base)			Our RS-Net		
	Pixel Acc.	Mean Acc.	IoU	Pixel Acc.	Mean Acc.	IoU
352×352	72.3	56.7	43.9	**72.3** (+0.0)	**57.0** (+0.3)	**44.1** (+0.2)
224×224	69.6	52.2	40.5	**71.4** (+1.8)	**54.6** (+2.4)	**42.6** (+2.1)
96×96	50.4	26.5	18.1	**63.0** (+12.6)	**43.1** (+16.6)	**32.1** (+14.0)
Total Params	$118.20\text{M} \times 3 = 354.60\text{M}$			118.31M		

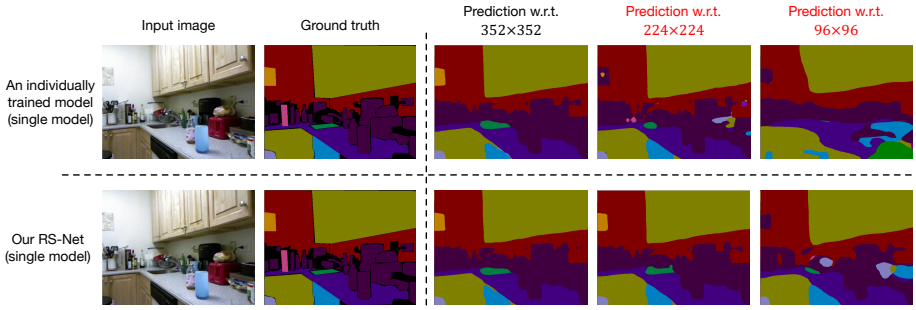

Fig. 3. Performance comparison of an individual model and our RS-Net. The rightmost four sub-figures (with titles marked in red) verify that our model can better maintain the performance when input resolution at inference is downsized for the sake of saving inference time.

4 Comparison with FixRes

Although FixRes [7] and our work have both considered resolution adaptation, they are different in motivation and design. FixRes focuses on improving accuracy by operating models at much higher resolution at test time, relying on manual fine-tuning for adaptation and test-time augmentations. However, our method focuses on efficient and flexible resolution adaptation at test time without additional latency such as fine-tuning. The accuracy comparison is possible as we both consider experiments on ResNet-50. In Fig.5 of FixRes and Table 5 of its supplementary material, for a ResNet-50 model trained with 224×224 images, the top-1 accuracy drops 9.4% from 224×224 (77.1%) to 128×128 (67.7%), while ours merely drops 3.0% from 224×224 (79.3%) to 128×128 (76.3%) (Table 1 of our main paper). Therefore, our method can better suppress the accuracy drop when input image resolution is downsized, which is beneficial to the model deployment in a resource-constrained platform. A more detailed comparison is shown in Table 2, which indicates that our model has much better performance at low resolutions, saving a large amount of FLOPs but even achieving higher

Table 2. Top-1 accuracies (%) comparison at different testing resolutions. Our model has better performance especially at low resolutions, which means being able to achieve better accuracy-efficiency trade-offs at runtime. Note that the accuracy at 224×224 of RS-Net has already surpassed all results of FixRes.

Model \ Resolution	64	128	224	288	352	384	448
FixRes [7]	41.7	67.7	77.1	78.5	**78.9**	**79.0**	**78.4**
Our RS-Net	**61.1**	**76.3**	**79.3**	**79.2**	78.1	77.4	75.8
Multiply-Adds	338M	1.35G	4.14G	6.84G	10.22G	12.17G	16.56G

performance. For example, our accuracy at 224×224 surpasses the top accuracy of FixRes at 384×384, needing only 34% FLOPs. We conjecture that under the permission of training resources, our RS-Net has the potential to achieve better performance by adding larger resolutions (e.g. 384×384 or larger) for training.

5 Discrepancy and Interaction Effects

We conduct an additional contrast experiment for verifying our analysis in Section 3.2 of our main paper, where we propose that the multi-resolution interaction effects are highly correlated with the train-test discrepancy, which is a kind of distribution shift caused by different data pre-processing methods during training and testing. As a conclusion of our analysis, on account of the multi-resolution parallel training, accuracies at higher resolutions tend to be further improved, but the accuracy at the low resolution tends to be reduced. In this part, we try to reduce the train-test discrepancy and observe if such interaction effects are weakened.

The concept of the train-test discrepancy itself is revealed by [7]. We first re-explain this discrepancy, based on our experiment setting as a specific example. In Section 4.1 of our paper, we mention that during training, we randomly crop the data for augmentation with an area ratio[3] uniformly sampled in $[0.08, 1.0]$, which is a standard setting following [2,6,7,3]. Therefore the expectation of area ratio for training is $(0.08 + 1)/2 = 0.54$. During testing, we first resize images to the target resolution divided by 0.875 (following [3,1]), and then crop the central regions with the target resolution. Therefore the expectation of area ratio for testing is $0.875^2 \approx 0.77$, which is larger than during training. As a larger crop means a smaller apparent object size, so on average, the apparent object size in testing is smaller than in training, which is the so-called train-test discrepancy [7]. Note that the parameters $[0.08, 1.0]$ and 0.875 are not always adopted by all image recognition works, but the train-test discrepancy typically exists (in different degrees) [7].

We alleviate the discrepancy by modifying $[0.08, 1.0]$ to $[0.3, 1.0]$, because the expectation of area ratio for training becomes $(0.3 + 1)/2 = 0.65$, which is closer to 0.77 in testing. Results of parallel training (without MRED) are illustrated

[3] The area ratio means the ratio of the cropped image area to the original image area.

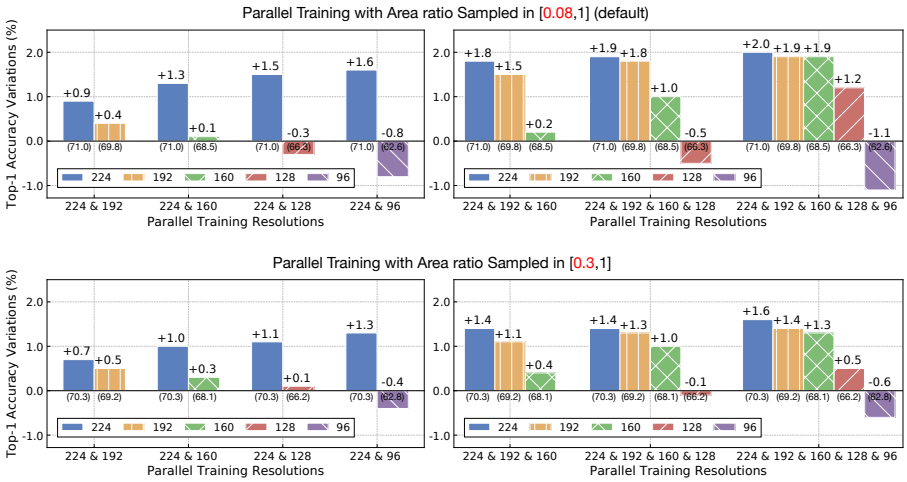

Fig. 4. Absolute top-1 accuracy variations (%) (compared with individual models) of parallel trainings, based on ResNet18, with two settings of the area ratio. The top-1 accuracy (%) of each individual model (from I-96 to I-224) is written in the bracket, which is used as the baseline. We use single numbers to represent the image resolutions.

in Fig. 4, including top-1 accuracy variations over each individual model. We can see that by alleviating the discrepancy, interaction effects are weakened, as accuracy gains at high resolutions and accuracy drops at the lowest resolution are both alleviated. Besides, in Fig. 4, we also provide the accuracy of each individual model (see each number in the bracket). We observe that sampling the area ratio in $[0.08, 1.0]$ has better overall performance than $[0.3, 1.0]$, which also indicates why $[0.08, 1.0]$ is a more popular choice for training on ImageNet.

References

1. Hoffer, E., Weinstein, B., Hubara, I., Ben-Nun, T., Hoefler, T., Soudry, D.: Mix & match: training convnets with mixed image sizes for improved accuracy, speed and scale resiliency. arXiv preprint arXiv:1908.08986 (2019)
2. Huang, G., Liu, Z., van der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: CVPR (2017)
3. Li, D., Zhou, A., Yao, A.: Hbonet: Harmonious bottleneck on two orthogonal dimensions. In: ICCV (2019)
4. Lin, G., Milan, A., Shen, C., Reid, I.D.: Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In: CVPR (2017)
5. Nathan Silberman, Derek Hoiem, P.K., Fergus, R.: Indoor segmentation and support inference from rgbd images. In: ECCV (2012)
6. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S.E., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: CVPR (2015)
7. Touvron, H., Vedaldi, A., Douze, M., Jégou, H.: Fixing the train-test resolution discrepancy. In: NeurIPS (2019)