Representation Sharing for Fast Object Detector Search and Beyond – Supplementary Material –

Yujie Zhong, Zelu Deng, Sheng Guo, Matthew R. Scott, and Weilin Huang

Malong LLC
{jaszhong, zeldeng, sheng, mscott, whuang}@malong.com

In this supplementary material, we first describe the implementation details and more ablation studies for object detection regarding the macro-structure of the detector search space, and the balance between width and depth of the searched architectures (Section 1). We then provide the implementation details for instance segmentation in Section 2, including the search and the training. Lastly, Section 3 provides some details of the search space and search method for image classification, as well as the results on the effect of decoupling RepShare.

1 Object Detection

1.1 Implementation Details

Architecture Search for the Subnetworks. The search takes $12.5k$ iterations with batch size 4. The initial learning rate is 0.004 and divided by 10 at iteration 10k. The size of the input images are resized such that the short side is 416 and the long side is less or equal to 693. The norm of the gradients are clipped to 20 to stabilize the search. During the search, we derive a discrete architecture every 2.5k iterations. The search process is terminated when the current derived architecture is the same as the previous one (i.e. 2.5k iterations before it). In our experiments, we find that the search mostly terminates at 12.5k iterations on VOC, and continuing the search does not change the derived architecture. Meanwhile, $L_{train}(w, \alpha)$, which is also considered as an indicator of the search process, flattens. Therefore, we use 12.5k iterations for all the experiments for consistency. To balance the efficiency and consistency between the search and detector retraining, we set $M = 1$ and $c' = 96$ for the search. Notably, $c = 256$ is used for all the experiments in this work, including the search and training.

Object detector training. Once we obtain the derived architectures, we train the whole detector on the MS-COCO *train2017*. For FCOS, we exactly follow the training strategy as in [7] for different backbone networks, including input image sizes, learning rate schedule and iterations. Similarly, the same training strategy as in [5] is adopted for RetinaNet. All the FCOS detectors (including vanilla and FAD) are trained using the improvements introduced in [7]. Note that the centerness in FCOS is predicted based on the first cell group. For detector training, we set $M = 2$ and $c' = 96$, unless specified.

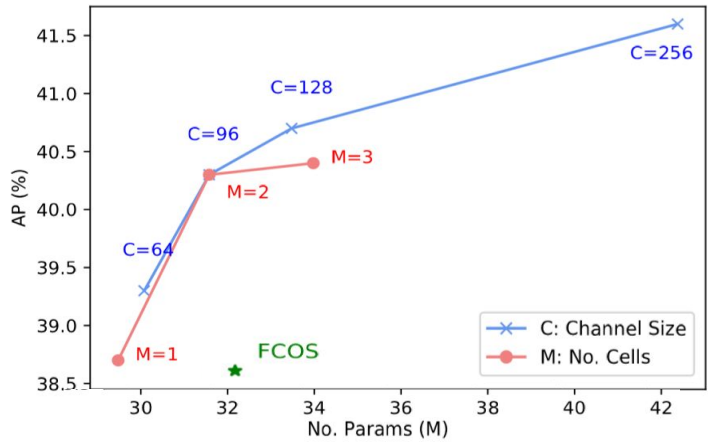

Fig. 1. FAD with different channel sizes and number of repeated cells. When channel size C varies, the number of repeated cells in the groups M is fixed to be 2; whereas C is fixed to be 96 when M varies. Results are obtained on the MS-COCO *minival* set, using ResNet-50 as the backbone.

1.2 More Ablation Studies on Object Detection

Ablation on the macro-structure of FAD. Searching and training a FAD with two parallel groups for classification and box regression (i.e. similar to RetinaNet [5] and FCOS [7]) achieve 40.0 AP, which is 0.3 lower than the sequential-group design in the main paper. We also experiment with the sequential groups in reverse order, i.e. classification is performed after the first group and regression after the second. This results in a drop of 0.1 AP. Therefore, the design of the macro-structure brings much smaller improvements comparing to the proposed search method.

Channel size and the number of cells. The capacity of the derived architecture searched by FAD can vary in two directions: the channel size c' in the transformation blocks and the number of repeated cells M in both groups. For a better understanding on their effects, we take FAD on FCOS with ResNet-50 [2] as a base model and vary c' and M. Figure 1 shows the performance trend of both directions. The two curves grow fast at the beginning and then increase in a slower pace, as we expect. Nevertheless, all the FAD models outperform FCOS. As the model with $M = 2$ has the best trade-off between performance and model size, we fix $M = 2$ throughout the object detection experiments in the main paper.

2 Implementation Details for Instance Segmentation

Architecture search for the mask head. The search is performed using Mask R-CNN [1] on the MS-COCO *train2017* set, which is randomly split into two halves: one for optimizing the architecture α and the other for learning the

Table 1. Ablation on the decoupling. Test error (lower is better) on CIFAR-10. *Shared Trans.* refers to the transformations that correspond to the shared representations (i.e. t_1 and t_2 in Figure 2 in the main paper), and *Other Trans.* denotes the t_3 to t_5.

Method	Decouple	Shared Trans. (%)	Other Trans. (%)	Params (M)	Test Error (%)
w/o RepShare	-	87.1	12.9	3.3	2.93 ± 0.11
w RepShare	✗	30.0	70.0	3.8	3.15 ± 0.12
	✓	85.7	14.3	3.2	2.98 ± 0.08

network weights w. We set M, c' and c to be 1, 64 and 256, respectively. The search takes 180k iterations, with an initial learning rate of 0.02 and decreased by 10 at the 120k and 160k iteration. The weight decay is 0.00001. The rest of search details are the same as object detector search. The search for mask head architecture requires 2.6 GPU-days. We use the same strategy as that for object detectors to derive the searched architecture.

Segmentation network training. For training Mask FAD with the searched architecture, we exactly follow [1]. For a fair comparison, we set $M = 2$ and $c' = 96$, which results in a similar capacity of Mask R-CNN. We also transfer the searched architecture to a more recent segmentation network, Mask Scoring R-CNN [3] (MS R-CNN). The same training scheme as in [3] is adopted for MS FAD.

3 Image Classification

To further demonstrate the effectiveness of decoupling the transformations from the shared representations, we conduct ablation study on CIFAR-10 [4].

3.1 Search Space and Search Method

We follow the experimental details in [6], but with a different search space. For normal cells, the candidate operations are t_1 to t_5, while the reduction cell only has two options: max/avg-pooling. 'Skip-connect' is included to both cells. Note that the goal of the experiments is not to search for better architectures for image classification. Instead, by constructing this simpler and smaller search space, we can illustrate the decoupling effect more clearly. To reduce the variance, we search for 4 architectures with or without the decoupling, and train for 5 runs each.

3.2 Results and Discussion

Table 1 reports the percentage of two groups of transformations being chosen in the derived architectures and the mean error with standard deviation. With the decoupling, t_1 and t_2 are more likely to appear after the derivation, since they

are no longer strictly tied to the shared representations. More importantly, the performance is better with the decoupling. We see that the coupling effect harms the search quality for image classification but not much for object detection and instance segmentation. We think that it is because t_1 and t_2 are not as favored in those two tasks as in image classification, since the two tasks benefit more from the combination of larger RFs.

References

1. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. In: Proceedings of the IEEE international conference on computer vision. pp. 2961–2969 (2017)
2. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
3. Huang, Z., Huang, L., Gong, Y., Huang, C., Wang, X.: Mask scoring R-CNN. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6409–6418 (2019)
4. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
5. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision. pp. 2980–2988 (2017)
6. Liu, H., Simonyan, K., Yang, Y.: DARTS: Differentiable architecture search. arXiv preprint arXiv:1806.09055 (2018)
7. Tian, Z., Shen, C., Chen, H., He, T.: Fcos: Fully convolutional one-stage object detection. arXiv preprint arXiv:1904.01355 (2019)