Shape Adaptor: A Learnable Resizing Module (Supplementary Material)

Shikun Liu[1], Zhe Lin[2], Yilin Wang[2], Jianming Zhang[2], Federico Perazzi[2], and Edward Johns[1]

[1] Department of Computing, Imperial College London
[2] Adobe Research

A General Formation of Shape Adaptors

In our paper, we formulated shape adaptors as a two-branch design, where input feature maps are processed by two branches, with the output feature maps then being recombined. Shape adaptors can be extended into a multi-branch design, into a more general manner. Each shape adaptor module is then composed with $K \geq 2$ resizing layers $F_{i=1:K}$, with fixed reshaping factors $r_{i=1:K} > 0$, and the corresponding learnable scaling weight parameters $\alpha_{i=1:K} \in (0,1)$.

We first define the set of reshaping factors r_i, and scaling weights α_i in resizing layers F_i:

$$\boldsymbol{r} = \{r_{i=1:K} \,|\, r_i > 0, \, \exists m, n : r_m \neq r_n\}, \boldsymbol{\alpha} = \left\{ \alpha_{i=1:K} \,\middle|\, \sum_{i=1}^{K} \alpha_i = 1, \, \alpha_i > 0 \right\}. \tag{1}$$

The general system for a shape adaptor module is formulated as follows:

$$\texttt{ShapeAdaptor}(x, \boldsymbol{\alpha}, \mathbf{r}) = \sum_{i=1}^{K} \alpha_i \cdot G\left(F_i\left(x, r_i\right), \; \frac{s(\boldsymbol{\alpha})}{r_i}\right), \tag{2}$$

with $s(\boldsymbol{\alpha})$ satisfies

$$s(\boldsymbol{\alpha})_{\alpha_k \to 1} = r_k, \quad \text{and} \quad s(\boldsymbol{\alpha}) \mapsto \mathcal{R}. \tag{3}$$

Then, the module's learnable reshaping factor mapped from scaling weights $\boldsymbol{\alpha} \to s(\boldsymbol{\alpha})$, is defined in the search space interval $\mathcal{R} = (\min(\boldsymbol{r}), \max(\boldsymbol{r}))$.

The weighted generalised mean:

$$s_0(\boldsymbol{\alpha}) = \prod_{i=1}^{K} r_i^{\alpha_i}, \quad \text{and} \quad s_p(\boldsymbol{\alpha}) = \left(\sum_{i=1}^{K} \alpha_i r_i^p\right)^{1/p}, \, p \neq 0 \tag{4}$$

are examples of suitable reshaping function design.

The general design for multi-branch shape adaptors can be inserted into more complicated networks architectures, such as ResNeXt [5] and Xception [2]. It can also be seen as a direct enhancement to spatial pyramid pooling [3,1], and U-Net [4], to enable them propagate context information from different, rather than the same feature dimensions.

B The Complete Hyper-Parameter Table

In this section, for reproducibility, we present a detailed list of hyper-parameter choices, across all networks and datasets evaluated in Table 1.

	Small Datasets: [32 × 32]			Fine-Grained Datasets: [224 × 224]			ImageNet: [224 × 224]		
	VGG-16	ResNet-50	MobileNetv2	VGG-16	ResNet-50	MobileNetv2	VGG-16	ResNet-50	MobileNetv2
A - Learning Rate	0.1			0.1			0.1		
A - Optimiser	SGD with 0.9 momentum			SGD with 0.9 momentum			SGD with 0.9 momentum		
A - Scheduler	Cosine Annealing			Cosine Annealing			Cosine Annealing		
A - Update Step	20			20			1500		
A - Number	Eq. 2: $\log_2(D^{in}/2)$ (4 for [32 × 32] images, 6 for [224 × 224] images)								
A - Initialisation	Eq. 4 with $D^{out} = 8$								
A - Location	Uniformly distributed (across all layers except for the last layer)								
A - Search Space	$(0.5, 1.0)$ (for every shape adaptor module)								
W - Learning Rate	0.1			0.01			0.1	0.1	0.05
W - Optimiser	SGD with 0.9 momentum			SGD with 0.9 momentum			SGD with 0.9 momentum		
W - Weight Decay	$5 \cdot 10^{-4}$	$5 \cdot 10^{-4}$	$4 \cdot 10^{-5}$	$5 \cdot 10^{-4}$	$5 \cdot 10^{-4}$	$4 \cdot 10^{-5}$	$5 \cdot 10^{-4}$	$5 \cdot 10^{-4}$	$4 \cdot 10^{-5}$
W - Scheduler	Cosine Annealing			Cosine Annealing			Cosine Annealing		
Batch Size	128			8			32 (per GPU) for 8 GPUs		
Epochs	200			200			120		

Table 1: The complete hyper-parameter applied to reproduce Table 1.

C Negative Results

- **Other choices in shape adaptor search space.** We experimented with shape adaptors in the reshaping range $(0.25, 1)$, which we found to converge to a similar overall network shape, but with degraded performance compared to the current setting. We also experimented with shape adaptors with the reshaping range in $(0.5, 2.0)$, which we found to have very unstable learning dynamics, and often with out of memory issues.
- **Other choices in reshaping function design.** We evaluated shape adaptors with the reshaping function $s(\alpha) = \frac{1}{\alpha/r_1 + (1-\alpha)/r_2}$, a weighted harmonic mean, which we found to have no improvements compared to the current setting.
- **Other optimisation methods.** We experimented with updating network shape parameters and weight parameters based on a different sample in the training dataset, which we found to have a degraded performance compared to the current setting.
- **Learning shape with prior structure knowledge.** We have experimented with directly replacing human-designed resizing layers with shape adaptors, which we found to have a minor effect on final performance compared to the current setting.
- **Alternative shape adaptor design in a residual cell.** We have experimented with an alternate design of the residual cell, with $[1 \times 1]$ convolution layer as the identity branch, and with the weight layer as the resizing branch. The final performance with such design achieved worse performance compared to the current setting.

References

1. Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017.
2. François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017.
3. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 37(9):1904–1916, 2015.
4. Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.
5. Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017.