Supplementary Material
ELASTIC: Improving CNNs with Dynamic Scaling Policies

Huiyu Wang[1] Aniruddha Kembhavi[2] Ali Farhadi[2,3,4] Alan Yuille[1] Mohammad Rastegari[2,4]
[1]Johns Hopkins University [2]PRIOR @ Allen Institute for AI
[3]University of Washington [4]Xnor.ai
huiyu@jhu.edu {anik,mohammadr}@allenai.org ali@cs.uw.edu alan.l.yuille@gmail.com

This supplementary material contains the following:

A. **sElastic**: a method to reduce flops while still reducing error rates

B. Details of the Elastic architecture and comparisons to current models

C. Screenshots of a demo visualizing different scale policies for ImageNet val images

D. Semantic segmentation qualitative results

In addition, our code and pre-trained models are available here: https://github.com/allenai/elastic
And our project website is online here: https://prior.allenai.org/projects/elastic

A. sElastic (simple Elastic)

A simple way of augmenting current models with Elastic is directly replacing bottlenecks by Elastic bottlenecks. This leads to models with less FLOPs and exactly the same number of parameters, which we refer to as sElastic (simple Elastic). This is in comparison to Elastic models that maintain the number of FLOPs and parameters. As shown in Table 1, sElastic already outperforms some of the original models, with less FLOPs. Note that DLA-X60+sElastic in Table 1 is equivalent to DLA-X60+Elastic (in Table 2 in the original paper), i.e. we do not add/remove layers in different scales.

Model	# Params	FLOPs	Top-1	Top-5
ResNext50	25.0M	4.2B	22.2	-
ResNext50*	25.0M	4.2B	22.23	6.25
ResNext50+sElastic	25.0M	3.4B	**22.03**	**6.07**
ResNeXt50+Elastic	25.2M	4.2B	21.56	5.83
DLA-X60	17.6M	3.6B	21.8	-
DLA-X60*	17.6M	3.6B	21.92	6.03
DLA-X60+sElastic	17.6M	3.2B	**21.25**	**5.71**
DLA-X60+Elastic	17.6M	3.2B	21.25	5.71
DLA-X102	26.8M	6.0B	21.5	-
DLA-X102+sElastic	26.8M	5.0B	**21.0**	**5.66**
DLA-X102+Elastic	24.9M	6.0B	20.71	5.38

Table 1: **Error rates for sElastic on the ImageNet validation set.** sElastic models with reduced FLOPs already perform better than some of the original models. We also provide the Elastic versions from the original paper as a reference.

B. Elastic Architecture Details

SElastic already outperforms original models. However, only applying downsamplings equivalently shifts computation from low level to higher level, which could cause lack of low level features to support high level processing. Also, sElastic reduces FLOPs so that its accuracy is not fairly comparable with the original model. For these two reasons, we rearrange computation distribution in each resolution, and this leads to our final Elastic model.

Consider ResNeXt-50 as an example. The original model assigns [3, 4, 6, 3] blocks respectively to [56, 28, 14, 7] four scales. As shown in Table 2, sElastic simply replaces original bottlenecks with Elastic bottlenecks. In Elastic, we roughly match the scale distribution of the original model by assigning [6, 8, 5, 3] blocks to those resolutions, as shown in Table 2. Note that half of each block processes information at a higher level. This modification also leads to matched number of parameters, and matched number of FLOPs. For ResNeXt101, we use a block design of [12, 14, 20, 3]. DenseNet+Elastic and DLA+Elastic architectures are shown respectively in Table 3 and Table 4. Note that these block designs were picked to match the original number of parameters and FLOPs, so we didn't tune them as hyper-parameters. Tuning them could probably lead to even lower error rates.

stage	ResNeXt50	ResNeXt50+sElastic	ResNeXt50+Elastic
conv1	7×7, 64, stride 2, 3×3 max pool, stride 2		
conv2 56×56	$\left[\begin{array}{c} 1\times1, 128 \\ 3\times3, 128, C{=}32 \\ 1\times1, 256 \end{array}\right] \times 3$	$\left[\begin{array}{ccc} & & 2\times\text{down}, 28\times28 \\ 1\times1, 64 & & 1\times1, 64 \\ 3\times3, 64, C{=}16 & + & 3\times3, 64, C{=}16 \\ 1\times1, 256 & & 1\times1, 256 \\ & & 2\times\text{up}, 56\times56 \end{array}\right] \times 3$	$\left[\begin{array}{ccc} & & 2\times\text{down}, 28\times28 \\ 1\times1, 64 & & 1\times1, 64 \\ 3\times3, 64, C{=}16 & + & 3\times3, 64, C{=}16 \\ 1\times1, 256 & & 1\times1, 256 \\ & & 2\times\text{up}, 56\times56 \end{array}\right] \times 6$
conv3 28×28	$\left[\begin{array}{c} 1\times1, 256 \\ 3\times3, 256, C{=}32 \\ 1\times1, 512 \end{array}\right] \times 4$	$\left[\begin{array}{ccc} & & 2\times\text{ down}, 14\times14 \\ 1\times1, 128 & & 1\times1, 128 \\ 3\times3, 128, C{=}16 & + & 3\times3, 128, C{=}16 \\ 1\times1, 512 & & 1\times1, 512 \\ & & 2\times\text{ up}, 28\times28 \end{array}\right] \times 4$	$\left[\begin{array}{ccc} & & 2\times\text{ down}, 14\times14 \\ 1\times1, 128 & & 1\times1, 128 \\ 3\times3, 128, C{=}16 & + & 3\times3, 128, C{=}16 \\ 1\times1, 512 & & 1\times1, 512 \\ & & 2\times\text{ up}, 28\times28 \end{array}\right] \times 8$
conv4 14×14	$\left[\begin{array}{c} 1\times1, 512 \\ 3\times3, 512, C{=}32 \\ 1\times1, 1024 \end{array}\right] \times 6$	$\left[\begin{array}{ccc} & & 2\times\text{ down}, 7\times7 \\ 1\times1, 256 & & 1\times1, 256 \\ 3\times3, 256, C{=}16 & + & 3\times3, 256, C{=}16 \\ 1\times1, 1024 & & 1\times1, 1024 \\ & & 2\times\text{ up}, 14\times14 \end{array}\right] \times 6$	$\left[\begin{array}{ccc} & & 2\times\text{ down}, 7\times7 \\ 1\times1, 256 & & 1\times1, 256 \\ 3\times3, 256, C{=}16 & + & 3\times3, 256, C{=}16 \\ 1\times1, 1024 & & 1\times1, 1024 \\ & & 2\times\text{ up}, 14\times14 \end{array}\right] \times 5$
conv5 7×7	$\left[\begin{array}{c} 1\times1, 1024 \\ 3\times3, 1024, C{=}32 \\ 1\times1, 2048 \end{array}\right] \times 3$		
1×1	global average pool, 1000-d fc, softmax		
Params.	25.0×10^6	25.0×10^6	25.2×10^6
FLOPs	4.2×10^9	3.4×10^9	4.2×10^9

Table 2: **ResNeXt50 vs. ResNeXt50+sElastic vs. ResNeXt50+Elastic.** ResNeXt50+Elastic employs two resolutions in each block, and keeps output resolution high for more blocks, compared with ResNeXt50.

C. Scale policy demo

Apart from Figure 1 and Figure 6 in the main paper, we made an interactive HTML based demo of our learned scale policy, that allows a user to explore images in the validation set and their scale policies. In the following screenshots we show some images where ResNeXt50+Elastic improves over the original ResNeXt50 on ImageNet validation set. Figures 1 and 2 show two screenshots. Each screenshot shows images with their classes, their scale policy visualizations, and their scale policy scores at all layers. The user can search through images and sort these images by their categories or their scale policy score at any layer. We refer interested reader to section 4.1.1 of the main paper for the definition of scale policy score and more discussions on different scale policies.

D. Semantic segmentation results

Some visualizations of our semantic segmentation results are shown in Figure 3, demonstrating that Elastic segments scale-challenging objects well on PASCAL VOC.

stage	DenseNet201	DenseNet201+Elastic
conv1	7×7, 64, stride 2, 3×3 max pool, stride 2	
conv2 56×56	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 32 \end{bmatrix} \times 6$	$\begin{bmatrix} \begin{matrix} 1\times1, 64 \\ 3\times3, 32 \end{matrix} + \begin{matrix} 2\times\text{down}, 28\times28 \\ 1\times1, 64 \\ 3\times3, 32 \\ 2\times\text{up}, 56\times56 \end{matrix} \end{bmatrix} \times 10$
trans1	1×1 conv, 2×2 average pool, stride 2	
conv3 28×28	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 32 \end{bmatrix} \times 12$	$\begin{bmatrix} \begin{matrix} 1\times1, 64 \\ 3\times3, 32 \end{matrix} + \begin{matrix} 2\times\text{down}, 14\times14 \\ 1\times1, 64 \\ 3\times3, 32 \\ 2\times\text{up}, 28\times28 \end{matrix} \end{bmatrix} \times 20$
trans2	1×1 conv, 2×2 average pool, stride 2	
conv4 14×14	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 32 \end{bmatrix} \times 48$	$\begin{bmatrix} \begin{matrix} 1\times1, 64 \\ 3\times3, 32 \end{matrix} + \begin{matrix} 2\times\text{down}, 7\times7 \\ 1\times1, 64 \\ 3\times3, 32 \\ 2\times\text{up}, 14\times14 \end{matrix} \end{bmatrix} \times 40$
trans3	1×1 conv, 2×2 average pool, stride 2	
conv5 7×7	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 32 \end{bmatrix} \times 32$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 32 \end{bmatrix} \times 30$
1×1	global average pool, 1000-d fc, softmax	
Params.	20.0×10^6	19.5×10^6
FLOPs	4.4×10^9	4.2×10^9

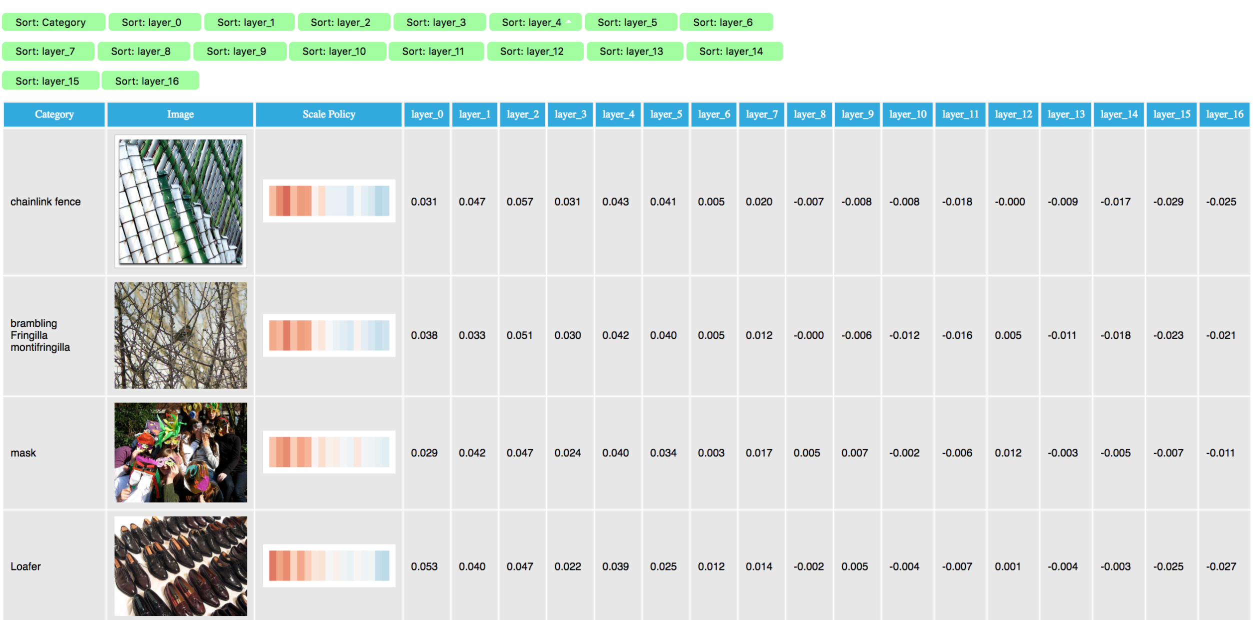
Table 3: **DenseNet201 vs. DenseNet201+Elastic.** DenseNet+Elastic follows a similar modification as ResNeXt+Elastic, i.e. two resolutions in each block and more blocks in high resolutions.

Name	Block	Stage 1	Stage 2	Stage 3	Stage 4	Stage 5	Stage 6	Params.	FLOPs
DLA-X60	Split32	16	32	1-128	2-256	3-512	1-1024	17.7×10^6	3.6×10^9
DLA-X60+Elastic	Split32+Elastic	16	32	1-128	2-256	3-512	1-1024	17.7×10^6	3.2×10^9
DLA-X102	Split32	16	32	1-128	3-256	4-512	1-1024	26.8×10^6	6.0×10^9
DLA-X102+sElastic	Split32+Elastic	16	32	1-128	3-256	4-512	1-1024	26.8×10^6	5.0×10^9
DLA-X102+Elastic	Split50+Elastic	16	32	3-128	3-256	3-512	1-1024	24.9×10^6	6.0×10^9

Table 4: **DLA model architectures.** Following DLA, we show our DLA classification architectures in the table. Split32 means a ResNeXt bottleneck with 32 paths while Split50 means a ResNeXt bottleneck with 50 paths. Stages 3 to 6 show d-n where d is the aggregation depth and n is the number of channels.

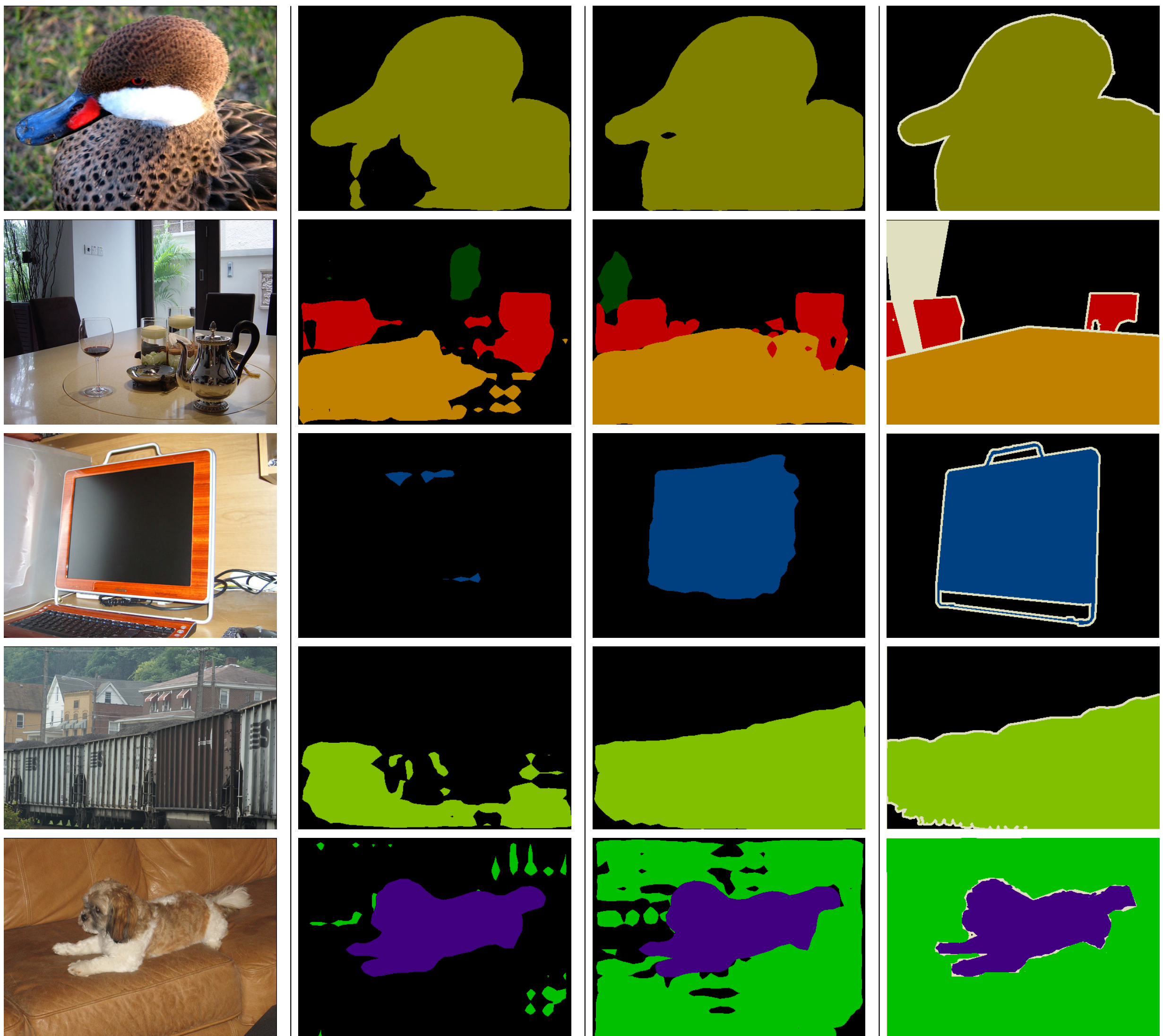
Category	Image	Scale Policy	layer_0	layer_1	layer_2	layer_3	layer_4	layer_5	layer_6	layer_7	layer_8	layer_9	layer_10	layer_11	layer_12	layer_13	layer_14	layer_15	layer_16
candle taper wax light			0.088	0.051	0.024	-0.010	-0.003	-0.008	-0.007	-0.009	-0.018	-0.018	-0.009	-0.013	-0.014	-0.007	-0.004	-0.013	-0.023
envelope			0.033	0.035	0.014	-0.007	0.005	-0.005	-0.001	-0.002	-0.012	-0.013	-0.014	-0.015	-0.010	-0.008	-0.007	-0.016	-0.014
window shade			0.063	0.024	0.028	0.005	0.005	-0.002	0.001	-0.002	-0.010	-0.013	-0.016	-0.018	-0.008	-0.012	-0.016	-0.029	-0.024
spotlight spot			0.064	0.047	0.018	-0.010	0.006	-0.003	-0.005	-0.004	-0.013	-0.016	-0.012	-0.014	-0.012	-0.007	-0.009	-0.018	-0.022

Figure 1: **Screenshots of the scale policy demo.** Examples of low scale scores at layer 4. These images usually contain a simple pattern.



Category	Image	Scale Policy	layer_0	layer_1	layer_2	layer_3	layer_4	layer_5	layer_6	layer_7	layer_8	layer_9	layer_10	layer_11	layer_12	layer_13	layer_14	layer_15	layer_16
chainlink fence			0.031	0.047	0.057	0.031	0.043	0.041	0.005	0.020	-0.007	-0.008	-0.008	-0.018	-0.000	-0.009	-0.017	-0.029	-0.025
brambling Fringilla montifringilla			0.038	0.033	0.051	0.030	0.042	0.040	0.005	0.012	-0.000	-0.006	-0.012	-0.016	0.005	-0.011	-0.018	-0.023	-0.021
mask			0.029	0.042	0.047	0.024	0.040	0.034	0.003	0.017	0.005	0.007	-0.002	-0.006	0.012	-0.003	-0.005	-0.007	-0.011
Loafer			0.053	0.040	0.047	0.022	0.039	0.025	0.012	0.014	-0.002	0.005	-0.004	-0.007	0.001	-0.004	-0.003	-0.025	-0.027

Figure 2: **Screenshots of the scale policy demo.** Examples of high scale scores at layer 4. These images require detailed processing at high resolutions.



(a) Validation Images (b) ResNeXt101 (c) ResNeXt101+Elastic (d) Ground truth

Figure 3: **Semantic segmentation results on PASCAL VOC.** Elastic improves most on scale-challenging images.