MicroNet: Large-Scale Model Compression Competition

Mengze Zeng* Momenta

zengmengze@momenta.ai

Ziheng Wu* Momenta

wuziheng@momenta.ai

Jie Hu*

State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences
University of Chinese Academy of Sciences
Momenta

hujie@ios.ac.cn

1 Model Architecture

In the competition, we play in *ImageNet Classification* track. We employ MixConv [7] integrated with SE-module [3] and HardSwish [1] as basic block to form the entry model, named **ProfitableNet**. And the design of overall module arrangement borrows from off-the-shelf efficient mobile model in ProxylessNAS [2]. The schema of ProfitableNet is showed in Appendix A.

2 Sparsity

Benefitting from effective sparsity matrix computation and storage, non-structured pruning plays an important role in cost-efficient inference and storage. In this part, we use naive sparsity strategy by zero low-weight prarameters to complete non-structured pruning, keeping accuracy with the aid of finetuning with sparisity. The process is depicted in Fig. 1

Implementation of sparisity requires following procedures:

- ► Sensitivity Analysis
- ► Generating parameters mask
- ► Finetuning with mask

Sensitivity Analysis is a method to decide which layers need to be pruned and the degree of sparsity for these layers. For each layer with parameters (e.g. Convolution layer), we iteratively zero $10\% \sim 90\%$ with step 5% parameters whose absolute value are relatively small and report the accuracy on a special mini-train dataset split from whole train set for each percentage. After that, we can plot a sensitivity curve like in Fig. 2. By means of the curve, we can easily find which layers are sensitive to be pruned. Empirically, we set a unified lower bound of accuracy and prune the maximum percentage parameters whose test accuracy exceed the bound for each layer.

Generating parameters mask. After sensitivity analysis, we can generate a binary mask for each layer with parameters to record where set to zero. The generated masks are also indispensable during the following finetuning step.

^{*}Equal contribution

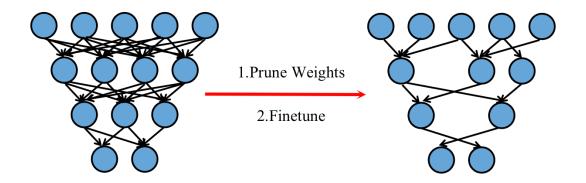


Figure 1: Training flow of non-structured sparsity.

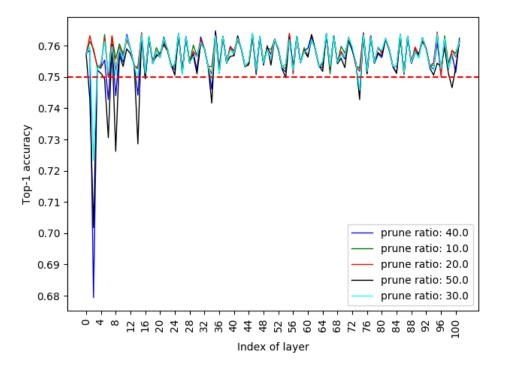


Figure 2: The curve of sensitive analysis result of prune ratio in $10\% \sim 50\%$, other prune ratios are omitted for clarity.

Finetuning with mask. In the last, we finetune the model with the generated masks for several epoch to recoup the loss in accuracy. Throughout the finetuning, the pruned parameters are never to be updated.

Following the above processes, we are able to prune model at about $30\%\sim50\%$ sparsity with neglectable loss in accuracy.

3 Quantization

Quantization is an another essential procedure which can largely accelerate inference.

Quantization for a floating-point vector *X* is able to be formulated:

$$x_{int} = clamp((round(\frac{x}{s}), INT_{min}, INT_{max})), x \in X$$

where x refers to an element in vector X, s refer to step of two adjacent fixed-point numbers, and INT_{min} and INT_{max} refer to corresponding lower bound and upper bound that fixed-point number can reach. Given a fixed-point number, the quantization process is to find the optimal step to minimize the error.

There are three different quantization approach used here:

- Symmetric KL-divergence based quantization
- Asymmetric KL-divergence based quantization
- Symmetric Max Value quantization

3.1 Symmetric KL-divergence based quantization

Symmetric KL-divergence based quantization is used for the activation which has negative responses (e.g. input, convolutional output without ReLU activation). The diagram is show in Fig. 3.

Pseudo code of calculating step

limitation: precision <= 10

```
SOURCE BINS = 2048
TARGET_BINS = 2^{(precison-1)}
bin[0], ..., bin[2047] = GenerateHist(fabs(Input))
For i in range (TARGET_BINS, SOURCE_BINS):
   reference_dist_P = [bin[0], ..., bin[i-1]]
   outliers_count = sum(bin[i] , bin[i+1] , ... , bin[SOURCE_BINS-1])
   reference_distribution_P[ i-1 ] += outliers_count
   P \neq sum(P) + normalize distribution P
   candidate_dist_Q = quantize[bin[0], ..., bin[i-1]] into TARGET_BINS
       levels
   expand candidate_distribution_Q to 'i' bins
   Q /= sum(Q) # normalize distribution Q
   divergence[i] = KL_divergence(reference_dist_P, candidate_dist_Q)
End For
Find the minimal KL_divergence called minKL
Find the max index 'h' for which divergence[k] <= tolerance * minKL
s = (h + 0.5) * (width of a source Tbin) / TARGET_BINS
return s
```

Hyperparameter

Precision: Specify the fixed-point number, such as int8 means precision=8.

Tolerance: Specify scale factor for minimal kl-divergence to find better step. We noticed that taking the step of minimal kl-divergence often obtain inferior result and enlarge this step can get better performance. So we introduce *tolerance* to relax the objective. Instead, we choose the max step in which kl-divergences less then *tolerance* * (minimal kl-divergence). In particular, we found that setting *tolerance* = 1.3 can achieve a good accuracy.

If a precision is given, the INT_{min} and INT_{max} can be calculated using the following formulation:

$$\begin{split} INT_{min} &= -2^{precision-1}\\ INT_{max} &= 2^{precision-1} - 1 \end{split}$$

3.2 Asymmetric KL-divergence based quantization

It is almost the same as Symmetric KL-divergence based quantization that customized for layer only has positive responses (e.g. ReLU or Sigmoid output). The diagram is exhibited in Fig. 4. In this

case, the lower bound and upper bound are defined:

$$INT_{min} = 0$$
$$INT_{max} = 2^{precision} - 1$$

3.3 Symmetric MaxValue quantization

The lower bound and upper bound formulation is the same as Symmetric KL-divergence based quantization. And the step calculation are formulated as: $s = max(fabs(X))/(2^{precision-1} - 1)$.

3.4 Accumulation in FP16

For common implementation of quantized matrix dot product, the accumulation part is still in FP32 or INT32 to maintain precision. In order to accelerate accumulation process, we find that FP16 is accurate enough in modern neural networks with normalization technology (e.g. BatchNorm, LayerNorm). In the competition, we implement convolution with GEMM by calling cublasGemmEx function with FP16 dataType and FP16 computeType in NVIDIA cublas engine.

3.5 Quantization in competition

In the competition, we use *fake quantization* throughout inference since lack of standard software support. Actually, we take floating-point number $x_{quant} = x_{int} * s$ as quantized number and use standard floating-point library to calculate. As mentioned above, we perform conversion after each convolution. If it followed by a ReLU or Sigmoid activation layer whose output are all non-negative, Asymmetric KL-divergence based quantization will be applied. Otherwise, Symmetric KL-divergence based quantization will be taken. As for convolutional kernels, we perform Symmetric MaxValue quantization. Notably, it is essential to quantize convolutional kernels in one layer separately which can largely reduce the accuracy loss (e.g. If the shape of convolutional kernels is $[c_{out}, c_{in}, k, k]$, c_{out} number of individual step need to be calculated). Besides, there are several special cases need to be pointed out:

- In residual-like block, we do not perform any fixed-point quantization closely followed the last convolution in residual branch. Instead, we quantize the feature maps after element-wise summation.
- In SE-module, we do not perform any fixed-point quantization on all feature maps as well. For the sake of reducing parameter storage, we conduct Symmetric MaxValue quantization for parameters of two FC layers.

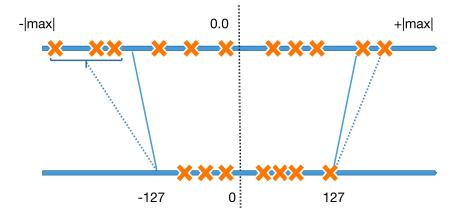


Figure 3: The diagram of symmetric KL-divergence based quantization while fixed-point number is 8.

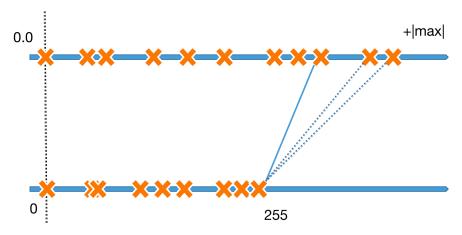


Figure 4: The diagram of asymmetric KL-divergence based quantization while fixed-point number is

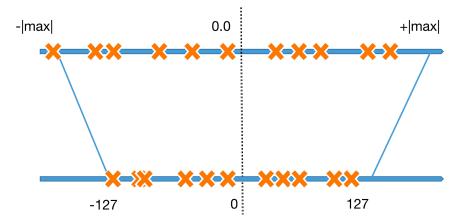


Figure 5: The diagram of asymmetric KL-divergence based quantization while fixed-point number is 8.

4 Training & Testing

4.1 Training

Our training schedule consists of five sequential steps that are training model from scratch, pruning model, finetuning pruned model, quantizing model and finetuning quantized model. The training flowchart is showed in Fig. 7.

Training from scratch

During training on ImageNet, we follow standard practice and perform data augmentation with random-size cropping [5] to 224×224 and random horizontal flipping. The mean channel subtraction are performed on input images. Notably, we replace the random-size cropping augmentation with random cropping from an image whose short edge is first resized to 256 in finetuning phase of sparsity and quantization. The model is optimized by synchronous SGD with momentum 0.9 and a mini-batch size of 256. The initial learning rate is set to 0.1 and is adjusted according to cosine annealing strategy [4]. In addition, a warmup strategy is used in the beginning 5 epoch where the learning rate increases linearly from 0 to 0.1. The model is trained for 300 epoch from scratch with weight decay 4e-5. Label-smoothing regularization [6] is used.

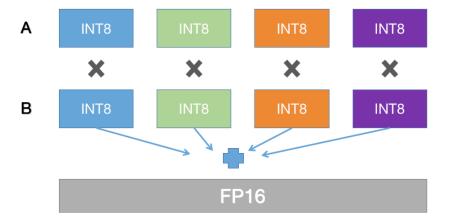


Figure 6: The diagram of fp16 accumulation matrix dot product

Pruning model and finetuning pruned model

We take sensitive analysis refer to Sec. 2 to prune previous well-trained model. In order to compensate the accuracy loss, we finetune the pruned model for 40 epochs with initial learning rate 1e-3 and decay by a factor of 0.1 at epoch 20. Other training settings are the same as training from scratch.

Quantizing model and finetuning quantized model

We randomly take 1000 images from training set as calibration set to calculate the appropriate steps of activation which need to be quantized. For the entry model, all the precision is set to 8 and the tolerance is set to 1.5. After that, we also require finetuning the quantized model to improve accuracy. We continue finetune it for 20 epochs with initial learning rate 5e-4, decayed by a factor of 0.2 at epoch 10.

4.2 Testing

When testing, we apply a centre crop evaluation on the validation set, where 224×224 pixels are cropped from each image whose shorter edge is first resized to 256. In the entry model, we eventually obtain 75.13% top-1 accuracy.

5 Scoring

The final score consists of two parts: (a) Parameter Storage (b) Math Operations. We implemented a tool for easily scoring. Once a model definition and sparsity file are given, the score can be figured out automatically.

5.1 Parameter Storage

In the entry model, we perform linear combination for all BatchNorm layers, integrating these parameters into their former convolutional layers or fully-connected layers. Here, we replace the last fully-connected layer (i.e. classifier layer) and fully-connected layers in SE-module with 1×1 convolutional layers. Consequently, all the convolutional kernels form the whole parameter set. And the size of parameter storage is affected by the degree of sparsity and fixed-point number of parameters. More concretely, the pseudo code of figuring out storage size is as follows:

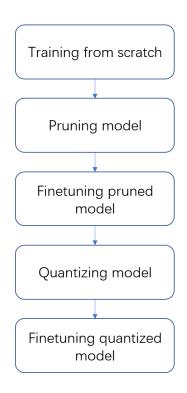


Figure 7: The training flow of our model

```
weight_count = c_out x c_in x k x k
storage = weight_count * weight_precision * (1 - layer_sparsity)
storage += weight_count // binary mask storage
if use_bias:
    storage += output_size * bias_precision;
storage /= 32
```

5.2 Math Operations

Math Operations consists of multiplication operations and addition operations. According to the competition rule, we demonstrate the process of calculating math operations in a convolution as follows:

```
mul_bit = max(input_precision, weight_precision);
add_bit = accumulation_precision
vector_length = c_in * k * k * (1 - layer_sparsity);
output_count = c_out * h_out * w_out
mul_bitops = vector_length * output_count * mul_bit;
add_bitops = (vector_length - 1) * output_count * add_bit;
if use_bias
    add_bitops += output_count * add_bit;
math_ops = (mul_bitops + add_bitops) / 32
```

Other operation's flops calculation are almost the same to Convolution Example.

5.3 Score report

The score of entry model is **0.203545**. The parameter storage and math operations are as follows:

Parameter storage: 0.652118 M
Mul operations: 44.3308 M
Add operations: 83.2402 M

More details of scoring each layer are exhibited in Appendix B.

References

- [1] Grace Chu Liang-Chieh Chen Bo Chen Mingxing Tan Weijun Wang Yukun Zhu Ruoming Pang Vijay Vasudevan Quoc V. Le Hartwig Adam Andrew Howard, Mark Sandler. Searching for mobilenetv3. In *ICCV*, 2019. 1
- [2] Han Cai, Ligeng Zhu, and Song Han. Proxylessnas: Direct neural architecture search on target task and hardware. In *ICLR*, 2019. 1
- [3] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In CVPR, 2018. 1
- [4] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. In *CoRR*, 2016. 5
- [5] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In CVPR, 2015. 5
- [6] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016. 5
- [7] Mingxing Tan and Quoc V. Le. Mixconv: Mixed depthwise convolutional kernels. *arXiv preprint arXiv:1907.09595*, 2019. 1

A Appendix: Model Schema

The Fig. 8 shows the details of the entry model.

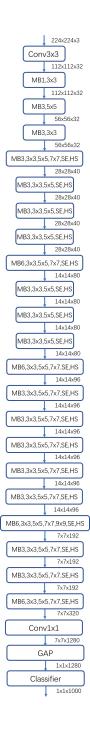


Figure 8: The overall schema of ProfitableNet.

B Scoring Details

The table ${\color{red}1}$ shows the details of scoring for each layer.

Table 1: score detail

layername	type	input1	input2	weight	sparsity	mul	add	mulops(M)	addops(M)	storage(K)
conv1/3x3_s2	Conv	8	-	8	0.50	8	16	1.3046	2.6092	0.15
conv1/3x3_s2/ relu	ReLU	8	-	-	-	8	-	0.1004	-	-
conv2_0/3x3_dwconv	Conv	8	_	8	0.00	8	16	0.9032	1.8063	0.09
conv2_0/3x3_dwconv/	ReLU	8	_	-	-	8	-	0.1004	-	-
relu	Reze							0.1001		
conv2_0/1x1_decrease	Conv	8	-	8	0.50	8	16	0.8028	1.6056	0.09
conv3_0/1x1_increase	Conv	8	_	8	0.50	8	16	1.2042	2.4084	0.14
conv3_0/1x1_increase/	ReLU	8	_	-	-	8	-	0.1505	2.1001	0.11
relu	RCLO	0		_	_		_	0.1303		_
conv3_0/ 3x3_dwconv	Conv	8	-	8	0.00	8	16	0.9408	1.8816	0.32
conv3_0/3x3_dwconv/	ReLU	8	_	-	-	8	-	0.0376	-	-
relu	Reze							0.0370		
conv3_0/1x1_decrease	Conv	8	_	8	0.50	8	16	0.6021	1.2042	0.26
conv3_1/1x1_increase	Conv	8	_	8	0.50	8	16	1.2042	2.4084	0.53
conv3_1/1x1_increase/	ReLU	8	_	_	-	8	-	0.0753	-	-
relu	l Reze							0.0755		
conv3_1/3x3_dwconv	Conv	8	-	8	0.00	8	16	0.6774	1.3548	0.26
conv3_1/3x3_dwconv/	ReLU	8	_	-	-	8	-	0.0753	-	-
relu						~		,		
conv3_1/1x1_decrease	Conv	8	-	8	0.50	8	16	1.2042	2.4084	0.50
conv3_1/ elt_sum	Elt	8	16	_	-	-	16	-	0.0502	-
conv4_0/1x1_increase	Conv	8	-	8	0.50	8	16	1.2042	2.4084	0.53
conv4_0/1x1_increase/	HardSwish	8	-	-	-	8	-	0.2258	-	-
hardswish	Tan do Wisir							0.2200		
conv4_0/3x3_dwconv/	Conv	8	_	8	0.50	8	16	0.0251	0.0502	0.06
slice 0					0.00			0.000		
conv4_0/3x3_dwconv/	HardSwish	8	_	_	_	8	_	0.0188	_	_
slice_0/hardswish								0.000		
conv4_0/3x3_dwconv/	Conv	8	_	8	0.50	8	16	0.0753	0.1505	0.14
slice 1										
conv4_0/3x3_dwconv/	HardSwish	8	_	-	-	8	-	0.0188	_	_
slice_1/hardswish										
conv4_0/3x3_dwconv/	Conv	8	-	8	0.50	8	16	0.1505	0.3011	0.26
slice_2										
conv4_0/3x3_dwconv/	HardSwish	8	-	-	-	8	-	0.0188	-	-
slice_2/hardswish										
conv4_0/3x3_dwconv/	Pool	8	-	-	-	8	16	0.0000	0.0376	-
SE_global_pool										
conv4_0/3x3_dwconv/	Conv	16	-	8	0.50	16	16	0.0001	0.0001	0.09
SE_fc1										
conv4_0/3x3_dwconv/	ReLU	16	-	-	-	16	-	0.0000	-	-
SE_fc1/relu										
conv4_0/3x3_dwconv/	Conv	16	-	8	0.50	16	16	0.0001	0.0001	0.09
SE_fc2										
conv4_0/3x3_dwconv/	Sigmoid	16	-	-	-	16	16	0.0001	0.0000	-
SE_weights										
conv4_0/3x3_dwconv/	Scale	8	16	-	-	16	-	0.0376	-	-
scale										
conv4_0/1x1_decrease	Conv	8	-	8	0.50	8	16	0.3763	0.7526	0.62
conv4_1/1x1_increase	Conv	8	-	8	0.50	8	16	0.4704	0.9408	0.81
conv4_1/1x1_increase/	HardSwish	8	-	-	-	8	-	0.0706	-	-
hardswish										
conv4_1/3x3_dwconv/	Conv	8	-	8	0.50	8	16	0.0470	0.0941	0.11
slice_0										
conv4_1/3x3_dwconv/	HardSwish	8	-	-	-	8	-	0.0353	-	-
slice_0/hardswish										

layername	type	input1	input2	weight	sparsity	mul	add	mulops(M)	addops(M)	storage(K)
conv4_1/3x3_dwconv/	Conv	8	-	8	0.50	8	16	0.1411	0.2822	0.26
slice_1										
conv4_1/3x3_dwconv/	HardSwish	8	-	-	-	8	-	0.0353	-	-
slice_1/hardswish										
conv4_1/3x3_dwconv/	Pool	8	-	-	-	8	16	0.0000	0.0470	-
SE_global_pool										
conv4_1/3x3_dwconv/	Conv	16	-	8	0.50	16	16	0.0002	0.0002	0.13
SE_fc1										
conv4_1/3x3_dwconv/	ReLU	16	-	-	-	16	-	0.0000	-	-
SE_fc1/relu										
conv4_1/3x3_dwconv/	Conv	16	-	8	0.50	16	16	0.0002	0.0001	0.13
SE_fc2								0.0004	0.0004	
conv4_1/3x3_dwconv/	Sigmoid	16	-	-	-	16	16	0.0001	0.0001	-
SE_weights								0.0450		
conv4_1/3x3_dwconv/	Scale	8	16	-	-	16	-	0.0470	-	-
scale		0		0	0.50		1.6	0.4704	0.0400	0.77
conv4_1/1x1_decrease	Conv	8	-	8	0.50	8	16	0.4704	0.9408	0.77
conv4_1/ elt_sum	Elt	8	16	-	-	-	16	-	0.0157	-
conv4_2/1x1_increase	Conv	8	-	8	0.50	8	16	0.4704	0.9408	0.81
conv4_2/1x1_increase/	HardSwish	8	-	-	-	8	-	0.0706	-	-
hardswish					0.50		1.0	0.0450	0.0041	0.11
conv4_2/3x3_dwconv/	Conv	8	-	8	0.50	8	16	0.0470	0.0941	0.11
slice_0	II IC : :							0.0252		
conv4_2/3x3_dwconv/	HardSwish	8	-	-	-	8	-	0.0353	-	-
slice_0/hardswish					0.50		1.0	0.1411	0.2022	0.26
conv4_2/3x3_dwconv/	Conv	8	-	8	0.50	8	16	0.1411	0.2822	0.26
slice_1	11 10 11	0				0		0.0252		
conv4_2/3x3_dwconv/	HardSwish	8	-	-	-	8	-	0.0353	-	-
slice_1/hardswish								0.0000	0.0450	
conv4_2/3x3_dwconv/	Pool	8	-	-	-	8	16	0.0000	0.0470	-
SE_global_pool		1.6		0	0.50	16	16	0.0002	0.0002	0.12
conv4_2/3x3_dwconv/	Conv	16	-	8	0.50	16	16	0.0002	0.0002	0.13
SE_fc1	ReLU	16				16		0.0000	-	
conv4_2/3x3_dwconv/	ReLU	10	-	-	-	16	-	0.0000	-	-
SE_fc1/relu conv4_2/3x3_dwconv/	Conv	16		8	0.50	16	16	0.0002	0.0001	0.13
SE_fc2	Conv	10	-	0	0.30	10	10	0.0002	0.0001	0.13
conv4_2/3x3_dwconv/	Sigmoid	16	_	_	_	16	16	0.0001	0.0001	_
SE_weights	Signiola	10	-	-	-	10	10	0.0001	0.0001	-
conv4_2/3x3_dwconv/	Scale	8	16	_		16	_	0.0470	-	_
scale	Scale	0	10	-	-	10	-	0.0470	-	-
conv4_2/1x1_decrease	Conv	0		8	0.50	8	16	0.4704	0.9408	0.77
conv4_2/ 1x1_decrease conv4_2/ elt_sum	Elt	8	- 16	0	0.30	0	16	0.4704	0.9408	0.77
		8	10	0	0.50	- 0		0.4704		0.91
conv4_3/1x1_increase	Conv HardSwish	_	-	8	0.50	8	16	0.4704 0.0706	0.9408	0.81
conv4_3/1x1_increase/ hardswish	naruSwish	8	-	_	_	8	-	0.0706	-	-
nardswish conv4_3/3x3_dwconv/	Conv	8		8	0.50	8	16	0.0470	0.0941	0.11
	Conv	8	_	8	0.50	8	10	0.0470	0.0941	0.11
slice_0 conv4_3/3x3_dwconv/	HardSwish	8				8		0.0353	-	
	naruswish	0	-	-	-	0	-	0.0555	-	-
slice_0/hardswish conv4_3/3x3_dwconv/	Conv	8		8	0.50	8	16	0.1411	0.2822	0.26
slice_1	Conv	0	-	0	0.50	0	10	0.1411	0.2822	0.20
conv4_3/3x3_dwconv/	HardSwish	8		_		8	-	0.0353	_	_
slice_1/hardswish	11aruSWISII	0	_	_	_	0	_	0.0333	-	-
	Pool	8				8	16	0.0000	0.0470	
conv4_3/3x3_dwconv/	POOL	8	-	-	-	8	16	0.0000	0.0470	-
SE_global_pool conv4_3/3x3_dwconv/	Conv	16		8	0.50	16	16	0.0002	0.0002	0.13
SE_fc1	COIIV	10	-	0	0.50	10	10	0.0002	0.0002	0.13
conv4_3/3x3_dwconv/	ReLU	16			_	16	-	0.0000	_	
SE_fc1/relu	KELU	10	_	_	_	10	_	0.0000	-	-
conv4_3/3x3_dwconv/	Conv	16	-	8	0.50	16	16	0.0002	0.0001	0.13
SE_fc2	Conv	10	_	0	0.50	10	10	0.0002	0.0001	0.13
3E_IC2										

layername	type	input1	input2	weight	sparsity	mul	add	mulops(M)	addops(M)	storage(K)
conv4_3/3x3_dwconv/	Sigmoid	16	-	-	-	16	16	0.0001	0.0001	-
SE_weights										
conv4_3/3x3_dwconv/	Scale	8	16	-	-	16	-	0.0470	-	-
scale conv4_3/1x1_decrease	Conv	8		8	0.50	8	16	0.4704	0.9408	0.77
conv4_3/ elt_sum	Elt	8	- 16	0	- 0.30	-	16	-	0.9408	-
conv5_0/1x1_increase	Conv	8	-	8	0.50	8	16	0.9408	1.8816	1.62
conv5_0/1x1_increase/	HardSwish	8	-	-	-	8	-	0.1411	-	-
hardswish	Tiaras wish							0.1111		
conv5_0/3x3_dwconv/ slice_0	Conv	8	-	8	0.50	8	16	0.0157	0.0314	0.15
conv5_0/3x3_dwconv/ slice_0/hardswish	HardSwish	8	-	-	-	8	-	0.0118	-	-
conv5_0/3x3_dwconv/ slice_1	Conv	8	-	8	0.50	8	16	0.0470	0.0941	0.35
conv5_0/3x3_dwconv/ slice_1/hardswish	HardSwish	8	-	-	-	8	-	0.0118	-	-
conv5_0/3x3_dwconv/ slice_2	Conv	8	-	8	0.50	8	16	0.0941	0.1882	0.65
conv5_0/3x3_dwconv/ slice_2/hardswish	HardSwish	8	-	-	-	8	-	0.0118	-	-
conv5_0/3x3_dwconv/ SE_global_pool	Pool	8	-	-	-	8	16	0.0001	0.0234	-
conv5_0/3x3_dwconv/ SE_fc1	Conv	16	-	8	0.50	16	16	0.0009	0.0009	0.56
conv5_0/3x3_dwconv/ SE_fc1/relu	ReLU	16	-	-	-	16	-	0.0000	-	-
conv5_0/3x3_dwconv/ SE_fc2	Conv	16	-	8	0.50	16	16	0.0008	0.0007	0.56
conv5_0/3x3_dwconv/ SE_weights	Sigmoid	16	-	-	-	16	16	0.0002	0.0001	-
conv5_0/3x3_dwconv/ scale	Scale	8	16	-	-	16	-	0.0235	-	-
conv5_0/ 1x1_decrease	Conv	8	-	8	0.50	8	16	0.4704	0.9408	3.04
conv5_1/1x1_increase	Conv	8	-	8	0.50	8	16	0.4704	0.9408	3.12
conv5_1/1x1_increase/ hardswish	HardSwish	8	-	-	-	8	-	0.0353	-	-
conv5_1/3x3_dwconv/ slice_0	Conv	8	-	8	0.50	8	16	0.0235	0.0470	0.23
conv5_1/3x3_dwconv/ slice_0/hardswish	HardSwish	8	-	-	-	8	-	0.0176	-	-
conv5_1/3x3_dwconv/ slice_1	Conv	8	-	8	0.50	8	16	0.0706	0.1411	0.53
conv5_1/3x3_dwconv/ slice_1/hardswish	HardSwish	8	-	-	-	8	-	0.0176	-	-
conv5_1/3x3_dwconv/ SE_global_pool	Pool	8	-	-	-	8	16	0.0001	0.0234	-
conv5_1/3x3_dwconv/ SE_fc1	Conv	16	-	8	0.50	16	16	0.0009	0.0009	0.56
conv5_1/3x3_dwconv/ SE_fc1/relu	ReLU	16	-	-	-	16	-	0.0000	-	-
conv5_1/3x3_dwconv/ SE_fc2	Conv	16	-	8	0.50	16	16	0.0008	0.0007	0.56
conv5_1/3x3_dwconv/ SE_weights	Sigmoid	16	-	-	-	16	16	0.0002	0.0001	-
conv5_1/3x3_dwconv/ scale	Scale	8	16	-	-	16	-	0.0235	-	-
conv5_1/1x1_decrease	Conv	8	-	8	0.50	8	16	0.4704	0.9408	3.04
conv5_1/ elt_sum	Elt	8	16	-	-	-	16	-	0.0078	-
conv5_2/1x1_increase	Conv	8	-	8	0.50	8	16	0.4704	0.9408	3.12
conv5_2/1x1_increase/ hardswish	HardSwish	8	-	-	-	8	-	0.0353	-	-

layername	type	input1	input2	weight	sparsity	mul	add	mulops(M)	addops(M)	storage(K)
conv5_2/3x3_dwconv/	Conv	8	-	8	0.50	8	16	0.0235	0.0470	0.23
slice_0								0.01=1		
conv5_2/3x3_dwconv/	HardSwish	8	-	-	-	8	-	0.0176	-	-
slice_0/hardswish conv5_2/3x3_dwconv/	Conv	8	_	8	0.50	8	16	0.0706	0.1411	0.53
slice_1			-	8	0.50	8	10		0.1411	0.55
conv5_2/3x3_dwconv/ slice_1/hardswish	HardSwish	8	-	-	-	8	-	0.0176	-	-
conv5_2/3x3_dwconv/	Pool	8	-	-	-	8	16	0.0001	0.0234	-
SE_global_pool conv5_2/3x3_dwconv/	Conv	16	-	8	0.50	16	16	0.0009	0.0009	0.56
SE_fc1 conv5_2/3x3_dwconv/	ReLU	16	-	_	-	16	_	0.0000	-	
SE_fc1/relu										
conv5_2/3x3_dwconv/ SE_fc2	Conv	16	-	8	0.50	16	16	0.0008	0.0007	0.56
conv5_2/3x3_dwconv/ SE_weights	Sigmoid	16	-	-	-	16	16	0.0002	0.0001	-
conv5_2/3x3_dwconv/ scale	Scale	8	16	-	-	16	-	0.0235	-	-
conv5_2/ 1x1_decrease	Conv	8	_	8	0.50	8	16	0.4704	0.9408	3.04
conv5_2/ elt_sum	Elt	8	16	-	-	-	16	-	0.0078	-
conv5_3/ 1x1_increase	Conv	8	-	8	0.50	8	16	0.4704	0.9408	3.12
conv5_3/1x1_increase/	HardSwish	8	-	-	-	8	-	0.0353	-	-
hardswish	Tanas Wish							0.0000		
conv5_3/3x3_dwconv/ slice_0	Conv	8	-	8	0.50	8	16	0.0235	0.0470	0.23
conv5_3/3x3_dwconv/	HardSwish	8	-	_	-	8	-	0.0176	-	-
slice_0/hardswish conv5_3/3x3_dwconv/	Conv	8	-	8	0.50	8	16	0.0706	0.1411	0.53
slice_1 conv5_3/3x3_dwconv/	HardSwish	8	-	_	_	8	_	0.0176	_	_
slice_1/hardswish										
conv5_3/3x3_dwconv/ SE_global_pool	Pool	8	-	-	-	8	16	0.0001	0.0234	-
conv5_3/3x3_dwconv/ SE_fc1	Conv	16	-	8	0.50	16	16	0.0009	0.0009	0.56
conv5_3/3x3_dwconv/ SE_fc1/relu	ReLU	16	-	-	-	16	-	0.0000	-	-
conv5_3/3x3_dwconv/	Conv	16	-	8	0.50	16	16	0.0008	0.0007	0.56
SE_fc2 conv5_3/3x3_dwconv/	Sigmoid	16	_	_	_	16	16	0.0002	0.0001	_
SE_weights							10		0.0001	
conv5_3/3x3_dwconv/ scale	Scale	8	16			16	_	0.0235	<u>-</u>	
conv5_3/1x1_decrease	Conv	8	-	8	0.50	8	16	0.4704	0.9408	3.04
conv5_3/ elt_sum	Elt	8	16	-	-	-	16	-	0.0078	-
conv6_0/1x1_increase	Conv	8	-	8	0.50	8	16	0.9408	1.8816	6.24
conv6_0/1x1_increase/	HardSwish	8	-	-	-	8	-	0.0706	-	-
hardswish conv6_0/3x3_dwconv/	Com	8		ρ	0.50	0	16	0.0314	0.0627	0.20
slice_0	Conv		-	8	0.50	8	16		0.0627	0.30
conv6_0/3x3_dwconv/ slice_0/hardswish	HardSwish	8	-	-	-	8	-	0.0235	-	-
conv6_0/3x3_dwconv/ slice_1	Conv	8	-	8	0.50	8	16	0.0941	0.1882	0.70
conv6_0/3x3_dwconv/ slice_1/hardswish	HardSwish	8	-	-	-	8	-	0.0235	-	-
conv6_0/3x3_dwconv/	Conv	8	-	8	0.50	8	16	0.1882	0.3763	1.30
slice_2 conv6_0/3x3_dwconv/	HardSwish	8	-	-	-	8	-	0.0235	-	-
slice_2/hardswish										

layername	type	input1	input2	weight	sparsity	mul	add	mulops(M)	addops(M)	storage(K)
conv6_0/3x3_dwconv/	Pool	8	-	-	-	8	16	0.0001	0.0468	-
SE_global_pool										
conv6_0/3x3_dwconv/ SE_fc1	Conv	16	-	8	0.50	16	16	0.0036	0.0036	2.25
conv6_0/3x3_dwconv/ SE_fc1/relu	ReLU	16	-	-	-	16	-	0.0000	-	-
conv6_0/3x3_dwconv/	Conv	16	-	8	0.50	16	16	0.0036	0.0034	2.25
SE_fc2 conv6_0/3x3_dwconv/	Sigmoid	16	_	_	_	16	16	0.0005	0.0002	
SE_weights										
conv6_0/3x3_dwconv/ scale	Scale	8	16	-	-	16	-	0.0470	-	-
conv6_0/1x1_decrease	Conv	8	-	8	0.50	8	16	1.1290	2.2579	7.25
conv6_1/1x1_increase	Conv	8	-	8	0.50	8	16	0.6774	1.3548	4.46
conv6_1/1x1_increase/ hardswish	HardSwish	8	-	-	-	8	-	0.0423	-	-
conv6_1/3x3_dwconv/	Conv	8	-	8	0.50	8	16	0.0188	0.0376	0.18
slice_0	HardSwish	8				0		0.0141		
conv6_1/3x3_dwconv/ slice_0/hardswish			-	-	-	8	-	0.0141	-	-
conv6_1/3x3_dwconv/ slice_1	Conv	8	-	8	0.50	8	16	0.0564	0.1129	0.42
conv6_1/3x3_dwconv/ slice_1/hardswish	HardSwish	8	-	-	-	8	-	0.0141	-	-
conv6_1/3x3_dwconv/ slice_2	Conv	8	-	8	0.50	8	16	0.1129	0.2258	0.78
conv6_1/3x3_dwconv/	HardSwish	8	-	-	-	8	-	0.0141	-	-
slice_2/hardswish conv6_1/3x3_dwconv/	Pool	8	-	-	-	8	16	0.0001	0.0281	-
SE_global_pool conv6_1/3x3_dwconv/	Conv	16	-	8	0.50	16	16	0.0013	0.0013	0.81
SE_fc1										
conv6_1/3x3_dwconv/ SE_fc1/relu	ReLU	16	-	-	-	16	-	0.0000	-	-
conv6_1/3x3_dwconv/ SE_fc2	Conv	16	-	8	0.50	16	16	0.0013	0.0012	0.81
conv6_1/3x3_dwconv/ SE_weights	Sigmoid	16	-	-	-	16	16	0.0003	0.0001	-
conv6_1/3x3_dwconv/ scale	Scale	8	16	-	-	16	-	0.0282	-	-
conv6_1/1x1_decrease	Conv	8	-	8	0.50	8	16	0.6774	1.3548	4.37
conv6_1/ elt_sum	Elt	8	16	-	-	-	16	-	0.0094	-
conv6_2/1x1_increase	Conv	8	-	8	0.50	8	16	0.6774	1.3548	4.46
conv6_2/1x1_increase/ hardswish	HardSwish	8	-	-	-	8	-	0.0423	-	-
conv6_2/3x3_dwconv/	Conv	8	-	8	0.50	8	16	0.0188	0.0376	0.18
slice_0 conv6_2/3x3_dwconv/	HardSwish	8	-	-	-	8	-	0.0141	-	-
slice_0/hardswish conv6_2/3x3_dwconv/	Conv	8	_	8	0.50	8	16	0.0564	0.1129	0.42
slice_1			-	o	0.50				0.1129	
conv6_2/3x3_dwconv/ slice_1/hardswish	HardSwish	8	-	-	-	8	-	0.0141	-	-
conv6_2/3x3_dwconv/ slice 2	Conv	8	-	8	0.50	8	16	0.1129	0.2258	0.78
conv6_2/3x3_dwconv/ slice_2/hardswish	HardSwish	8	-	-	-	8	-	0.0141	-	-
conv6_2/3x3_dwconv/ SE_global_pool	Pool	8	-	-	_	8	16	0.0001	0.0281	-
conv6_2/3x3_dwconv/ SE_fc1	Conv	16	-	8	0.50	16	16	0.0013	0.0013	0.81
	1	l	1	1		1	1			

layername	type	input1	input2	weight	sparsity	mul	add	mulops(M)	addops(M)	storage(K)
conv6_2/3x3_dwconv/	ReLU	16	-	-	-	16	-	0.0000	-	-
SE_fc1/relu										
conv6_2/3x3_dwconv/ SE_fc2	Conv	16	-	8	0.50	16	16	0.0013	0.0012	0.81
conv6_2/3x3_dwconv/ SE_weights	Sigmoid	16	-	-	-	16	16	0.0003	0.0001	-
conv6_2/3x3_dwconv/ scale	Scale	8	16	-	-	16	-	0.0282	-	-
conv6_2/ 1x1_decrease	Conv	8	_	8	0.50	8	16	0.6774	1.3548	4.37
conv6_2/ elt_sum	Elt	8	16	-	-	-	16	-	0.0094	-
conv6_3/1x1_increase	Conv	8	-	8	0.50	8	16	0.6774	1.3548	4.46
conv6_3/1x1_increase/ hardswish	HardSwish	8	-	-	-	8	-	0.0423	-	-
conv6_3/3x3_dwconv/ slice_0	Conv	8	-	8	0.50	8	16	0.0188	0.0376	0.18
conv6_3/3x3_dwconv/	HardSwish	8	-	-	-	8	-	0.0141	-	-
slice_0/hardswish conv6_3/3x3_dwconv/	Conv	8	-	8	0.50	8	16	0.0564	0.1129	0.42
slice_1 conv6_3/3x3_dwconv/	HardSwish	8	-	-	-	8	-	0.0141	-	-
slice_1/hardswish conv6_3/3x3_dwconv/	Conv	8	-	8	0.50	8	16	0.1129	0.2258	0.78
slice_2 conv6_3/3x3_dwconv/	HardSwish	8	_	_	-	8	_	0.0141	-	-
slice_2/hardswish conv6_3/3x3_dwconv/	Pool	8		_		8	16	0.0001	0.0281	
SE_global_pool			-	-	-					-
conv6_3/3x3_dwconv/ SE_fc1	Conv	16	-	8	0.50	16	16	0.0013	0.0013	0.81
conv6_3/3x3_dwconv/ SE_fc1/relu	ReLU	16	-	-	-	16	-	0.0000	-	-
conv6_3/3x3_dwconv/ SE_fc2	Conv	16	-	8	0.50	16	16	0.0013	0.0012	0.81
conv6_3/3x3_dwconv/ SE_weights	Sigmoid	16	-	-	-	16	16	0.0003	0.0001	-
conv6_3/3x3_dwconv/ scale	Scale	8	16	-	-	16	-	0.0282	-	-
conv6_3/1x1_decrease	Conv	8	_	8	0.50	8	16	0.6774	1.3548	4.37
conv6_3/ elt_sum	Elt	8	16	-	-	-	16	-	0.0094	-
conv6_4/ 1x1_increase	Conv	8	-	8	0.50	8	16	0.6774	1.3548	4.46
conv6_4/1x1_increase/	HardSwish	8	-	-	-	8	-	0.0423	-	-
hardswish										
conv6_4/3x3_dwconv/ slice_0	Conv	8	-	8	0.50	8	16	0.0188	0.0376	0.18
conv6_4/3x3_dwconv/ slice_0/hardswish	HardSwish	8	-	-	-	8	-	0.0141	-	-
conv6_4/3x3_dwconv/ slice_1	Conv	8	-	8	0.50	8	16	0.0564	0.1129	0.42
conv6_4/3x3_dwconv/	HardSwish	8	-	-	-	8	-	0.0141	-	-
slice_1/hardswish conv6_4/3x3_dwconv/	Conv	8	-	8	0.50	8	16	0.1129	0.2258	0.78
slice_2 conv6_4/3x3_dwconv/	HardSwish	8	-	-	-	8	-	0.0141	-	-
slice_2/hardswish conv6_4/3x3_dwconv/	Pool	8	-	-	-	8	16	0.0001	0.0281	
SE_global_pool					0.55					
conv6_4/3x3_dwconv/ SE_fc1	Conv	16	-	8	0.50	16	16	0.0013	0.0013	0.81
conv6_4/3x3_dwconv/ SE_fc1/relu	ReLU	16	-	_	-	16	-	0.0000	-	-
conv6_4/3x3_dwconv/ SE_fc2	Conv	16	-	8	0.50	16	16	0.0013	0.0012	0.81
20_102	1	l .	I	l	1	1	l	<u> </u>	1	

layername	type	input1	input2	weight	sparsity	mul	add	mulops(M)	addops(M)	storage(K)
conv6_4/3x3_dwconv/	Sigmoid	16	-	-	-	16	16	0.0003	0.0001	-
SE_weights										
conv6_4/3x3_dwconv/	Scale	8	16	-	-	16	-	0.0282	-	-
scale										
conv6_4/1x1_decrease	Conv	8	-	8	0.50	8	16	0.6774	1.3548	4.37
conv6_4/ elt_sum	Elt	8	16	-	-	-	16	-	0.0094	-
conv6_5/ 1x1_increase	Conv	8	-	8	0.50	8	16	0.6774	1.3548	4.46
conv6_5/1x1_increase/	HardSwish	8	-	-	-	8	-	0.0423	-	-
hardswish					0.70			0.0100	0.025	0.10
conv6_5/3x3_dwconv/	Conv	8	-	8	0.50	8	16	0.0188	0.0376	0.18
slice_0	11 10 11	0				0		0.0141		
conv6_5/3x3_dwconv/	HardSwish	8	-	-	-	8	-	0.0141	-	-
slice_0/hardswish	Conv	8		8	0.50	8	16	0.0564	0.1129	0.42
conv6_5/3x3_dwconv/	Conv	8	-	8	0.50	8	16	0.0364	0.1129	0.42
slice_1 conv6_5/3x3_dwconv/	HardSwish	8	_	_		8	-	0.0141	_	_
slice_1/hardswish	HaluSwisii	0	_	_	-	0	-	0.0141	-	-
conv6_5/3x3_dwconv/	Conv	8	_	8	0.50	8	16	0.1129	0.2258	0.78
slice_2	Conv	0	_	0	0.50	0	10	0.1129	0.2236	0.76
conv6_5/3x3_dwconv/	HardSwish	8	_	_	_	8	_	0.0141	_	_
slice_2/hardswish	Taruswish	0	_	_	_	0	-	0.0141	_	-
conv6_5/3x3_dwconv/	Pool	8	_	_	-	8	16	0.0001	0.0281	_
SE_global_pool	1 001		_	_			10	0.0001	0.0201	_
conv6_5/3x3_dwconv/	Conv	16	_	8	0.50	16	16	0.0013	0.0013	0.81
SE_fc1	Conv	10			0.50	10	10	0.0013	0.0015	0.01
conv6_5/3x3_dwconv/	ReLU	16	_	_	-	16	-	0.0000	-	_
SE_fc1/relu	1620	10				10		0.0000		
conv6_5/3x3_dwconv/	Conv	16	-	8	0.50	16	16	0.0013	0.0012	0.81
SE_fc2								0.000		
conv6_5/3x3_dwconv/	Sigmoid	16	-	-	-	16	16	0.0003	0.0001	_
SE_weights								0.000		
conv6_5/3x3_dwconv/	Scale	8	16	-	-	16	-	0.0282	-	-
scale										
conv6_5/1x1_decrease	Conv	8	-	8	0.50	8	16	0.6774	1.3548	4.37
conv6_5/ elt_sum	Elt	8	16	-	-	-	16	-	0.0094	-
conv7_0/ 1x1_increase	Conv	8	-	8	0.45	8	16	1.4676	2.9353	9.62
conv7_0/1x1_increase/	HardSwish	8	-	-	-	8	-	0.0847	-	-
hardswish										
conv7_0/3x3_dwconv/	Conv	8	-	8	0.45	8	16	0.0071	0.0141	0.29
slice_0										
conv7_0/3x3_dwconv/	HardSwish	8	-	-	-	8	-	0.0053	-	-
slice_0/hardswish										
conv7_0/3x3_dwconv/	Conv	8	-	8	0.45	8	16	0.0229	0.0459	0.68
slice_1										
conv7_0/3x3_dwconv/	HardSwish	8	-	-	-	8	-	0.0053	-	-
slice_1/hardswish					0.45			2 2 4 7 2	0.001=	
conv7_0/3x3_dwconv/	Conv	8	-	8	0.45	8	16	0.0459	0.0917	1.26
slice_2	1110 : 1	0	1			0		0.0052		
conv7_0/3x3_dwconv/	HardSwish	8	_	-	_	8	-	0.0053	-	-
slice_2/hardswish conv7_0/3x3_dwconv/	Comm	8		8	0.45	8	16	0.0776	0.1552	2.04
slice_3	Conv	0	-	0	0.43	0	10	0.0776	0.1332	2.04
conv7_0/3x3_dwconv/	HardSwish	8	_	_	_	8	-	0.0053	_	-
slice_3/hardswish	11atus Wisil	0	_	_	_	6	_	0.0033	_	-
conv7_0/3x3_dwconv/	Pool	8	_	_	_	8	16	0.0001	0.0138	_
SE_global_pool	1 001	8	_	_	_	"	10	0.0001	0.0136	-
conv7_0/3x3_dwconv/	Conv	16	_	8	0.50	16	16	0.0052	0.0052	3.24
SE_fc1	Conv	10	_		0.50	10	10	0.0032	0.0052	3.24
conv7_0/3x3_dwconv/	ReLU	16	-	_	_	16	-	0.0000	_	_
SE_fc1/relu	ROLO	10				10		0.0000		
conv7_0/3x3_dwconv/	Conv	16	-	8	0.50	16	16	0.0052	0.0049	3.24
SE_fc2					3.20		10	2.0052		<u>-</u> .
	1	1	I	I .	<u>I</u>	1		<u>I</u>		

Second Control Contr	layername	type	input1	input2	weight	sparsity	mul	add	mulops(M)	addops(M)	storage(K)
Conv7_O/181_decrease		Sigmoid		-	-	-					-
Scale	SE_weights										
CONTY		Scale	8	16	-	-	16	-	0.0141	-	-
Som File Conv Section Sect											
Convolution				-	_			1			
hardswish				-	8	0.45	1	16		1.4818	18.95
Convolution		HardSwish	8	-	-	-	8	-	0.0212	-	-
Sice_0											
Convolution		Conv	8	-	8	0.45	8	16	0.0094	0.0188	0.39
Silice_D/hardswish	slice_0	11 10 11	0						0.0071		
Comv 1/3x3 dwconv Conv Second Conv Second Conv Second Conv Second Conv Second Second Conv Second Secon		HardSwish	8	-	-	-	8	-	0.0071	-	-
Sice 1			0		0	0.45	0	16	0.0206	0.0612	0.01
Compton Comp		Conv	0	-	0	0.43	0	10	0.0306	0.0012	0.91
Silice_Jhardswish		HardCwich	0				0		0.0071		
Conv 1/3x3 dwconv Conv S - S 0.45 S 16 0.0612 0.1223 1.68		naruswish	0	-	-	-	0	-	0.0071	-	-
Sice 2		Conv	Q		Q	0.45	Q	16	0.0612	0.1223	1.68
Compt Comp		Conv	0	_	0	0.43	8	10	0.0012	0.1223	1.06
Sice_2Dhardswish		HardSwish	8	_	_	_	8	<u> </u>	0.0071		_
Conv_T_I/3x3_dwconv/ Pool 8		Tital do wish							0.0071		
SE_global_pool		Pool	8	_	-	-	8	16	0.0001	0.0138	_
Conv		1 001						10	0.0001	0.0100	
SE_fc1		Conv	16	-	8	0.50	16	16	0.0052	0.0052	3.24
Conv7_1/3x3_dwconv/ SE_Ic1/relu Conv SE_Ic1/relu Conv SE_Ic1/relu Conv SE_Ic2 Scale Scale											
SE_fel/relu		ReLU	16	-	-	-	16	-	0.0000	-	-
SE_fc2											
SE_fc2		Conv	16	-	8	0.50	16	16	0.0052	0.0049	3.24
SE_weights											
Conv7_1/3x3_dwconv/ Scale 8	conv7_1/3x3_dwconv/	Sigmoid	16	-	-	-	16	16	0.0006	0.0003	-
Scale	SE_weights										
Conv7_1/1x1_decrease	conv7_1/3x3_dwconv/	Scale	8	16	-	-	16	-	0.0141	-	-
Conv7_1/elt_sum											
conv7_2/1x1_increase Conv 8 - 8 0.45 8 16 0.7409 1.4818 18.95 conv7_2/1x1_increase/ hardswish HardSwish 8 - - - 8 - 0.0212 - - conv7_2/3x3_dwconv/ slice_0 Conv 8 - 8 0.45 8 16 0.0094 0.0188 0.39 slice_0 Conv7_2/3x3_dwconv/ slice_0/hardswish HardSwish 8 - - - 8 - 0.0071 -					8	0.45	8		0.7432		18.76
Conv7_2/1x1_increase/ hardswish				16					-		
hardswish				-	8	0.45		16		1.4818	18.95
Conv7_2/3x3_dwconv/ Sice_0 S		HardSwish	8	-	-	-	8	-	0.0212	-	-
Slice_0											
Conv7_2/3x3_dwconv/ Sice_0/hardswish		Conv	8	-	8	0.45	8	16	0.0094	0.0188	0.39
Slice_0/hardswish Conv S - S 0.45 S 16 0.0306 0.0612 0.91											
conv7_2/3x3_dwconv/ slice_1 Conv 8 - 8 0.45 8 16 0.0306 0.0612 0.91 conv7_2/3x3_dwconv/ slice_1/hardswish HardSwish 8 - - - 8 - 0.0071 - - conv7_2/3x3_dwconv/ slice_2/hardswish Conv 8 - - - 8 - 0.0071 - - conv7_2/3x3_dwconv/ sE_global_pool HardSwish 8 - - - 8 16 0.0001 0.0138 - conv7_2/3x3_dwconv/ SE_fc1/relu Conv 16 - 8 0.50 16 16 0.0052 0.0052 3.24 conv7_2/3x3_dwconv/ SE_fc2 ReLU 16 - - - 16 - 0.0000 - - conv7_2/3x3_dwconv/ SE_weights Sigmoid 16 - - - 16 16 0.0006 0.0003 - conv7_2/3x3_dwconv/ scale Scale 8 16		HardSwish	8	-	-	-	8	-	0.0071	-	-
Slice_I						0.45		1.6	0.0207	0.0612	0.01
conv7_2/3x3_dwconv/ slice_1/hardswish HardSwish 8 - - 8 - 0.0071 - - conv7_2/3x3_dwconv/ slice_2 Conv 8 - 8 0.45 8 16 0.0612 0.1223 1.68 conv7_2/3x3_dwconv/ slice_2/hardswish B - - 8 - 0.0071 - - - conv7_2/3x3_dwconv/ sE_global_pool Pool 8 - - 8 16 0.0001 0.0138 - conv7_2/3x3_dwconv/ sE_fc1 Conv 16 - 8 0.50 16 16 0.0052 0.0052 3.24 SE_fc1/relu Conv7_2/3x3_dwconv/ sE_weights Conv 16 - 8 0.50 16 16 0.0052 0.0049 3.24 SE_weights Conv7_2/3x3_dwconv/ scale Sigmoid 16 - - - 16 0.0006 0.0003 -		Conv	8	-	8	0.45	8	16	0.0306	0.0612	0.91
slice_I/hardswish Conv7_2/3x3_dwconv/ slice_2 Conv 8 - 8 0.45 8 16 0.0612 0.1223 1.68 slice_2 conv7_2/3x3_dwconv/ slice_2/hardswish 8 - - - 8 - 0.0071 - - conv7_2/3x3_dwconv/ slice_2/hardswish Pool 8 - - - 8 16 0.0001 0.0138 - conv7_2/3x3_dwconv/ slice_2/hardswish Conv 16 - 8 16 0.0001 0.0138 - conv7_2/3x3_dwconv/ slice_2/hardswish Conv 16 - 8 0.50 16 16 0.0052 0.0052 3.24 SE_fc1 SE_fc1 SE_fc1/relu -		HondS:-1	O				n		0.0071		
conv7_2/3x3_dwconv/ slice_2 Conv 8 - 8 0.45 8 16 0.0612 0.1223 1.68 conv7_2/3x3_dwconv/ slice_2/hardswish HardSwish 8 - - - 8 - 0.0071 - - conv7_2/3x3_dwconv/ SE_global_pool Pool 8 - - - 8 16 0.0001 0.0138 - conv7_2/3x3_dwconv/ SE_fc1 Conv 16 - 8 0.50 16 16 0.0052 0.0052 3.24 SE_fc1/relu conv7_2/3x3_dwconv/ Conv 16 - 8 0.50 16 16 0.0052 0.0049 3.24 SE_fc2 conv7_2/3x3_dwconv/ Sigmoid 16 - - - 16 16 0.0006 0.0003 - scale 8 16 - - 16 - 0.0141 - -		naruSwish	ð	_	_	_	8	_	0.00/1	-	-
Slice_2 Slic		Conv	Q		Q	0.45	Q	16	0.0612	0.1223	1.69
conv7_2/3x3_dwconv/ slice_2/hardswish HardSwish 8 - - - 8 - 0.0071 - <t< td=""><td></td><td>Conv</td><td>0</td><td>_</td><td>0</td><td>0.43</td><td> °</td><td>10</td><td>0.0012</td><td>0.1223</td><td>1.00</td></t<>		Conv	0	_	0	0.43	°	10	0.0012	0.1223	1.00
Slice_2/hardswish	conv7 2/3v3 dwconv/	HardSwich	8	_	_	_	8		0.0071		_
conv7_2/3x3_dwconv/ SE_global_pool Pool 8 - - - 8 16 0.0001 0.0138 - conv7_2/3x3_dwconv/ SE_fc1 Conv 16 - 8 0.50 16 16 0.0052 0.0052 3.24 conv7_2/3x3_dwconv/ SE_fc1/relu ReLU 16 - - - 16 - 0.0000 - - - conv7_2/3x3_dwconv/ SE_weights Conv 16 - 8 0.50 16 16 0.0052 0.0049 3.24 conv7_2/3x3_dwconv/ Scale Sigmoid 16 - - - 16 16 0.0006 0.0003 -		11ai di S W 15ll	0	_	_	_	"	-	0.0071	-	_
SE_global_pool Conv7_2/3x3_dwconv/ Conv 16 - 8 0.50 16 16 0.0052 0.0052 3.24 SE_fc1 conv7_2/3x3_dwconv/ ReLU 16 - - - 16 - 0.0000 - - - SE_fc1/relu conv7_2/3x3_dwconv/ Conv 16 - 8 0.50 16 16 0.0052 0.0049 3.24 SE_fc2 conv7_2/3x3_dwconv/ Sigmoid 16 - - - 16 16 0.0006 0.0003 - SE_weights conv7_2/3x3_dwconv/ Scale 8 16 - - 16 - 0.0141 - - scale 8 16 - - 16 - 0.0141 - -		Pool	8	_	_	_	8	16	0.0001	0.0138	_
conv7_2/3x3_dwconv/ Conv 16 - 8 0.50 16 16 0.0052 0.0052 3.24 SE_fc1 conv7_2/3x3_dwconv/ ReLU 16 - - - 16 - 0.0000 - - SE_fc1/relu conv7_2/3x3_dwconv/ Conv 16 - 8 0.50 16 16 0.0052 0.0049 3.24 SE_fc2 conv7_2/3x3_dwconv/ Sigmoid 16 - - - 16 16 0.0006 0.0003 - SE_weights conv7_2/3x3_dwconv/ Scale 8 16 - - 16 - 0.0141 - - scale 8 16 - - 16 - 0.0141 - -		1501						10	0.0001	0.0150	
SE_fc1 conv7_2/3x3_dwconv/ ReLU 16 - - - 16 - 0.0000 -		Conv	16	-	8	0.50	16	16	0.0052	0.0052	3.24
conv7_2/3x3_dwconv/ SE_fc1/relu ReLU 16 - - - 16 - 0.0000 - - - - SE_fc1/relu - <		232.									- -
SE_fc1/relu Conv7_2/3x3_dwconv/ Conv 16 - 8 0.50 16 16 0.0052 0.0049 3.24 SE_fc2 conv7_2/3x3_dwconv/ Sigmoid 16 - - - 16 16 0.0006 0.0003 - SE_weights conv7_2/3x3_dwconv/ Scale 8 16 - - 16 - 0.0141 - - scale scale 8 16 - - 16 - 0.0141 - -		ReLU	16	-	-	-	16	-	0.0000	-	-
conv7_2/3x3_dwconv/ Conv 16 - 8 0.50 16 16 0.0052 0.0049 3.24 SE_fc2 conv7_2/3x3_dwconv/ Sigmoid 16 - - - 16 16 0.0006 0.0003 - SE_weights conv7_2/3x3_dwconv/ Scale 8 16 - - 16 - 0.0141 - - scale scale 8 16 - - 16 - 0.0141 - -											
SE_fc2 Conv7_2/3x3_dwconv/ Sigmoid 16 - - - 16 16 0.0006 0.0003 - SE_weights conv7_2/3x3_dwconv/ Scale 8 16 - - 16 - 0.0141 - - scale scale - - - - 0.0141 - -		Conv	16	-	8	0.50	16	16	0.0052	0.0049	3.24
conv7_2/3x3_dwconv/ Sigmoid 16 - - 16 16 0.0006 0.0003 - SE_weights conv7_2/3x3_dwconv/ Scale 8 16 - - 16 - 0.0141 - - scale scale -<	SE_fc2										
SE_weights	conv7_2/3x3_dwconv/	Sigmoid	16	-	-	-	16	16	0.0006	0.0003	-
conv7_2/3x3_dwconv/	SE_weights										
	conv7_2/3x3_dwconv/	Scale	8	16	-	-	16	-	0.0141	-	-
conv7_2/ 1x1_decrease											
	conv7_2/ 1x1_decrease	Conv	8	-	8	0.45	8	16	0.7432	1.4865	18.76

	input1	input2	weight	sparsity	mul	add	mulops(M)	addops(M)	storage(K)
Elt	8	16	-	-	-	16	-	0.0047	-
Conv		-	8	0.45	8	16		2.9635	37.90
HardSwish	8	-	-	-	8	-	0.0423	-	-
Conv	8	-	8	0.45	8	16	0.0188	0.0376	0.78
HardSwish	8	-	-	-	8	-	0.0141	-	-
Conv	8	-	8	0.45	8	16	0.0612	0.1223	1.81
HardSwish	8	-	-	-	8	-	0.0141	-	-
Conv	8	-	8	0.45	8	16	0.1223	0.2446	3.37
HardSwish	8	-	-	-	8	-	0.0141	-	-
Pool	8	-	-	-	8	16	0.0003	0.0276	-
Conv	16	-	8	0.50	16	16	0.0207	0.0207	12.96
ReLU	16	-	-	-	16	-	0.0000	-	-
Conv	16	-	8	0.50	16	16	0.0207	0.0202	12.96
Sigmoid	16	-	-	-	16	16	0.0012	0.0006	-
Scale	8	16	-	-	16	-	0.0282	-	-
Conv	8	-	8	0.45	8	16	2.4814	4.9627	62.37
Conv	8	-	8	0.50	8	16	2.5088	5.0176	64.64
ReLU	8	-	-	-	8	-	0.0157	-	-
Pool	8	-	-	-	8	16	0.0003	0.0307	-
Conv	8	-	8	0.50	8	16	0.1600	0.3200	200.50
	Conv HardSwish Conv HardSwish Conv HardSwish Conv HardSwish Conv HardSwish Pool Conv ReLU Conv Sigmoid Scale Conv Conv ReLU Pool	Conv 8 HardSwish 8 Conv 8 HardSwish 8 Conv 8 HardSwish 8 Conv 8 HardSwish 8 Conv 16 ReLU 16 Conv 16 Sigmoid 16 Scale 8 Conv 8 Conv 8 Conv 8 Conv 8 Conv 8 ReLU 8 Pool 8	Conv 8 - HardSwish 8 - Conv 8 - HardSwish 8 - Conv 8 - HardSwish 8 - Conv 8 - Pool 8 - Conv 16 - Conv 16 - Sigmoid 16 - Scale 8 16 Conv 8 - Conv 8 - ReLU 8 - Pool 8 -	Conv 8 - 8 HardSwish 8 - - Conv 8 - 8 HardSwish 8 - - Conv 8 - 8 HardSwish 8 - - Conv 8 - - Pool 8 - - Conv 16 - 8 ReLU 16 - - Conv 16 - 8 Sigmoid 16 - - Scale 8 16 - Conv 8 - 8 Conv 8 - 8 ReLU 8 - - Pool 8 - -	Conv 8 - 8 0.45 HardSwish 8 - - - Conv 8 - 8 0.45 HardSwish 8 - - - Conv 8 - 8 0.45 HardSwish 8 - - - Pool 8 - - - Pool 8 - - - Conv 16 - 8 0.50 ReLU 16 - - - Conv 16 - 8 0.50 Sigmoid 16 - - - Scale 8 16 - - Conv 8 - 8 0.45 Conv 8 - 8 0.50 ReLU 8 - - - - Pool 8 - - -	Conv 8 - 8 0.45 8 HardSwish 8 - - 8 0.45 8 Conv 8 - 8 0.45 8 HardSwish 8 - - - 8 Conv 8 - 8 0.45 8 HardSwish 8 - - - 8 Pool 8 - - - 8 Conv 16 - 8 0.50 16 ReLU 16 - - - 16 Sigmoid 16 - 8 0.50 16 Scale 8 16 - - 16 Conv 8 - 8 0.45 8 Conv 8 - 8 0.45 8 Conv 8 - 8 0.50 8 ReLU 8 - <td>Conv 8 - 8 0.45 8 16 HardSwish 8 - - - 8 - Conv 8 - 8 0.45 8 16 HardSwish 8 - - - 8 - Conv 8 - 8 0.45 8 16 HardSwish 8 - - - 8 - Pool 8 - - - 8 16 HardSwish 8 - - - 8 - Pool 8 - - - 8 - Pool 8 - - - 8 16 Conv 16 - - - 16 16 ReLU 16 - - - 16 - Conv 8 - 8 0.45 8 16<!--</td--><td>Conv 8 - 8 0.45 8 16 1.4818 HardSwish 8 - - - 8 - 0.0423 Conv 8 - 8 0.45 8 16 0.0188 HardSwish 8 - - - 8 - 0.0141 Conv 8 - 8 0.45 8 16 0.0612 HardSwish 8 - - - 8 - 0.0141 Conv 8 - 8 0.45 8 16 0.1223 HardSwish 8 - - - 8 - 0.0141 Pool 8 - - - 8 - 0.0141 Pool 8 - - - 8 16 0.0003 Conv 16 - 8 0.50 16 16 0.0207 Sigmoid<td>Conv 8 - 8 0.45 8 16 1.4818 2.9635 HardSwish 8 - - - 8 - 0.0423 - Conv 8 - 8 0.45 8 16 0.0188 0.0376 HardSwish 8 - - - 8 - 0.0141 - Conv 8 - 8 0.45 8 16 0.0612 0.1223 HardSwish 8 - - - 8 - 0.0141 - Conv 8 - 8 0.45 8 16 0.1223 0.2446 HardSwish 8 - - 8 16 0.1223 0.2446 HardSwish 8 - - 8 16 0.0223 0.2446 HardSwish 8 - - - 8 16 0.0003 0.0276 <</td></td></td>	Conv 8 - 8 0.45 8 16 HardSwish 8 - - - 8 - Conv 8 - 8 0.45 8 16 HardSwish 8 - - - 8 - Conv 8 - 8 0.45 8 16 HardSwish 8 - - - 8 - Pool 8 - - - 8 16 HardSwish 8 - - - 8 - Pool 8 - - - 8 - Pool 8 - - - 8 16 Conv 16 - - - 16 16 ReLU 16 - - - 16 - Conv 8 - 8 0.45 8 16 </td <td>Conv 8 - 8 0.45 8 16 1.4818 HardSwish 8 - - - 8 - 0.0423 Conv 8 - 8 0.45 8 16 0.0188 HardSwish 8 - - - 8 - 0.0141 Conv 8 - 8 0.45 8 16 0.0612 HardSwish 8 - - - 8 - 0.0141 Conv 8 - 8 0.45 8 16 0.1223 HardSwish 8 - - - 8 - 0.0141 Pool 8 - - - 8 - 0.0141 Pool 8 - - - 8 16 0.0003 Conv 16 - 8 0.50 16 16 0.0207 Sigmoid<td>Conv 8 - 8 0.45 8 16 1.4818 2.9635 HardSwish 8 - - - 8 - 0.0423 - Conv 8 - 8 0.45 8 16 0.0188 0.0376 HardSwish 8 - - - 8 - 0.0141 - Conv 8 - 8 0.45 8 16 0.0612 0.1223 HardSwish 8 - - - 8 - 0.0141 - Conv 8 - 8 0.45 8 16 0.1223 0.2446 HardSwish 8 - - 8 16 0.1223 0.2446 HardSwish 8 - - 8 16 0.0223 0.2446 HardSwish 8 - - - 8 16 0.0003 0.0276 <</td></td>	Conv 8 - 8 0.45 8 16 1.4818 HardSwish 8 - - - 8 - 0.0423 Conv 8 - 8 0.45 8 16 0.0188 HardSwish 8 - - - 8 - 0.0141 Conv 8 - 8 0.45 8 16 0.0612 HardSwish 8 - - - 8 - 0.0141 Conv 8 - 8 0.45 8 16 0.1223 HardSwish 8 - - - 8 - 0.0141 Pool 8 - - - 8 - 0.0141 Pool 8 - - - 8 16 0.0003 Conv 16 - 8 0.50 16 16 0.0207 Sigmoid <td>Conv 8 - 8 0.45 8 16 1.4818 2.9635 HardSwish 8 - - - 8 - 0.0423 - Conv 8 - 8 0.45 8 16 0.0188 0.0376 HardSwish 8 - - - 8 - 0.0141 - Conv 8 - 8 0.45 8 16 0.0612 0.1223 HardSwish 8 - - - 8 - 0.0141 - Conv 8 - 8 0.45 8 16 0.1223 0.2446 HardSwish 8 - - 8 16 0.1223 0.2446 HardSwish 8 - - 8 16 0.0223 0.2446 HardSwish 8 - - - 8 16 0.0003 0.0276 <</td>	Conv 8 - 8 0.45 8 16 1.4818 2.9635 HardSwish 8 - - - 8 - 0.0423 - Conv 8 - 8 0.45 8 16 0.0188 0.0376 HardSwish 8 - - - 8 - 0.0141 - Conv 8 - 8 0.45 8 16 0.0612 0.1223 HardSwish 8 - - - 8 - 0.0141 - Conv 8 - 8 0.45 8 16 0.1223 0.2446 HardSwish 8 - - 8 16 0.1223 0.2446 HardSwish 8 - - 8 16 0.0223 0.2446 HardSwish 8 - - - 8 16 0.0003 0.0276 <