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 $10\%\sim90\%$ 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

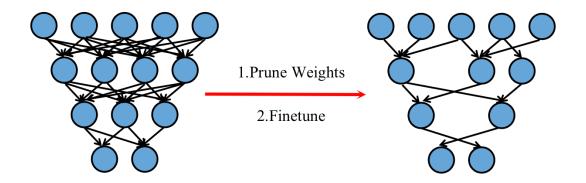


Figure 1: Training flow of non-structured sparsity.

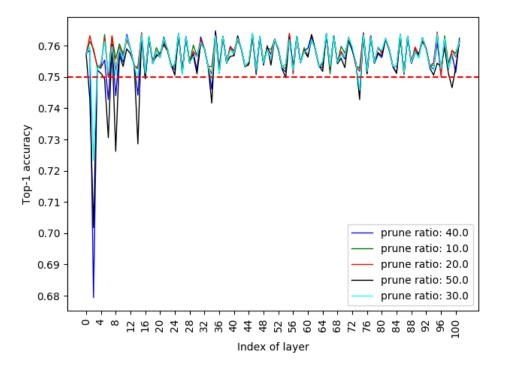


Figure 2: The curve of sensitive analysis result of prune ratio in $10\% \sim 50\%$, other prune ratios are omitted for clarity.

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. During the GEMM process, the multiplication result of INT8 activation with INT8 weights will be INT16(maximum=32767) in extremis, FP16 maximum is 65504, there exists some risk of overflow when we using a FP16 accumulation for INT8 multiplication. When we use the "fake quantization", Activation and weights are both in FP32(or FP16) format which envolves their quantization steps, these quantization steps decrease the magnitude of their multiplication result. As a result of this inaccurate simulation of quantized convolution, "fake quantization" evades the INT16 accumulation overflow risk during the FP16 GEMM.

In order to prove the FP16 accumulation's effectiveness, we have to use the activations and weights which has the correct magnitude(INT8 magnitude) in the FP16 GEMM. By dividing their quantizations step before we send them in to the FP16 GEMM and multiplying these steps after the FP16 GEMM, we design an accurate simulation of quantized convolution. In our proposal, we use the INT8 activation and INT6 weights, which decrease the theoretical maximum multiplication result from 32767 to 3937. During our experiments, the FP32 accumulation result can be inherited with slight variation(75.0022 -> 75.0762) when we switch to the FP16 accumulation with correct magnitude.

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

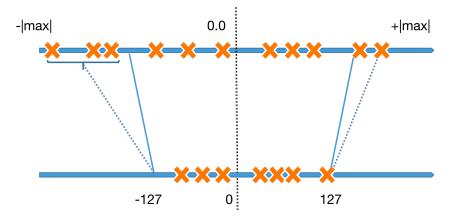


Figure 3: The diagram of symmetric KL-divergence based quantization while fixed-point number is 8.

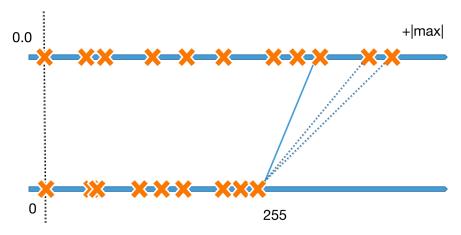


Figure 4: The diagram of asymmetric KL-divergence based quantization while fixed-point number is 8.

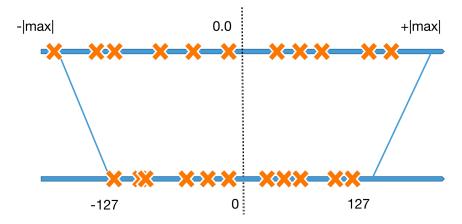


Figure 5: The diagram of asymmetric KL-divergence based quantization while fixed-point number is 8.

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.

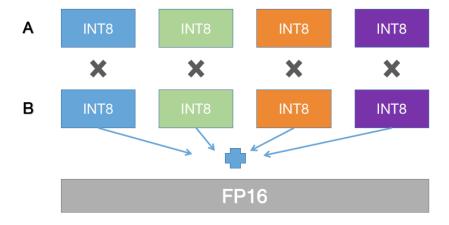


Figure 6: The diagram of fp16 accumulation matrix dot product

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.

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.

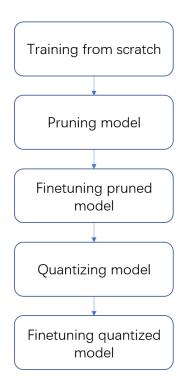


Figure 7: The training flow of our model

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 75.0762% 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:

```
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 **0.178852**. The parameter storage and math operations are as follows:

Parameter storage: 0.478411 M
Mul operations: 44.5208 M
Add operations: 83.6139 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.
- [4] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. In *CoRR*, 2016. 6
- [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. 6
- [6] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In CVPR, 2016. 6
- [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.

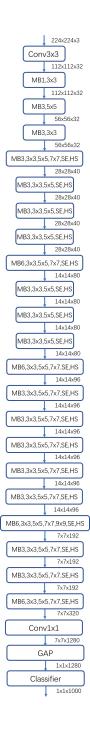


Figure 8: The overall schema of ProfitableNet.

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	-	6	0.5000	8	16	1.3046	2.6092	0.12
conv1/3x3_s2/ relu	ReLU	8	-	-	-	8	-	0.1004	-	-
conv2_0/ 3x3_dwconv	Conv	8	_	6	0.0000	8	16	0.9032	1.8063	0.07
conv2_0/3x3_dwconv/	ReLU	8	_	-	-	8	-	0.1004	-	-
relu	Reze							0.1001		
conv2_0/1x1_decrease	Conv	8	-	6	0.5000	8	16	0.8028	1.6056	0.07
conv3_0/1x1_increase	Conv	8	_	6	0.5000	8	16	1.2042	2.4084	0.12
conv3_0/1x1_increase/	ReLU	8	_	-	-	8	-	0.1505	2.1001	0.12
relu	RCLO	0		_	_		_	0.1303		
conv3_0/3x3_dwconv	Conv	8	_	6	0.0000	8	16	0.9408	1.8816	0.25
conv3_0/3x3_dwconv/	ReLU	8	_	-	-	8	-	0.0376	-	0.23
relu	RCLO	0		_	_		_	0.0370		
conv3_0/1x1_decrease	Conv	8	_	6	0.5000	8	16	0.6021	1.2042	0.21
conv3_1/1x1_increase	Conv	8	_	6	0.5000	8	16	1.2042	2.4084	0.43
conv3_1/1x1_increase/	ReLU	8	_	-	-	8	-	0.0753	2.1001	-
relu	Rele							0.0755		
conv3_1/3x3_dwconv	Conv	8	_	6	0.0000	8	16	0.6774	1.3548	0.21
conv3_1/3x3_dwconv/	ReLU	8	_	-	-	8	-	0.0753	1.55 10	0.21
relu	RCLO	0		_	_		_	0.0733		
conv3_1/1x1_decrease	Conv	8	_	6	0.5000	8	16	1.2042	2.4084	0.40
conv3_1/ elt_sum	Elt	8	16	-	0.5000	-	16	-	0.0502	- 0.10
conv4_0/1x1_increase	Conv	8	-	6	0.5000	8	16	1.2042	2.4084	0.43
conv4_0/1x1_increase/	HardSwish	8	_	-	0.3000	8	-	0.2258	2.4004	0.43
hardswish	Tiard5 wish	0	_	_	_	6	_	0.2236	_	_
conv4_0/3x3_dwconv/	Conv	8	_	6	0.5000	8	16	0.0251	0.0502	0.05
slice 0	Conv	0		0	0.3000		10	0.0231	0.0302	0.03
conv4_0/3x3_dwconv/	HardSwish	8	_	_	_	8	_	0.0188	_	_
slice_0/hardswish	Tiard5 wish	0		_	_		_	0.0100		_
conv4_0/3x3_dwconv/	Conv	8	_	6	0.5000	8	16	0.0753	0.1505	0.12
slice 1	Conv				0.5000		10	0.0755	0.1303	0.12
conv4_0/3x3_dwconv/	HardSwish	8	_	-	_	8	_	0.0188		_
slice_1/hardswish	Tital do wish							0.0100		
conv4_0/3x3_dwconv/	Conv	8	_	6	0.5000	8	16	0.1505	0.3011	0.21
slice_2	Conv				0.5000		10	0.1303	0.5011	0.21
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/	Scale	16	_	16	_	16	_	0.0000	_	0.00
SE_bn										
conv4_0/3x3_dwconv/	Conv	16	_	6	0.5000	16	16	0.0001	0.0001	0.07
SE_fc1										
conv4_0/3x3_dwconv/	ReLU	16	_	-	-	16	-	0.0000	_	_
SE_fc1/relu										
conv4_0/3x3_dwconv/	Conv	16	-	6	0.5000	16	16	0.0001	0.0001	0.07
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	-	6	0.5000	8	16	0.3763	0.7526	0.50
conv4_1/1x1_increase	Conv	8	-	6	0.5000	8	16	0.4704	0.9408	0.66
conv4_1/1x1_increase/	HardSwish	8	-	-	-	8	-	0.0706	-	-
hardswish										
conv4_1/3x3_dwconv/	Conv	8	-	6	0.5000	8	16	0.0470	0.0941	0.10
slice_0				-					·	
	1	1	1	1	i .			1		1

layername	type	input1	input2	weight	sparsity	mul	add	mulops(M)	addops(M)	storage(K)
conv4_1/3x3_dwconv/	HardSwish	8	-	-	-	8	-	0.0353	-	-
slice_0/hardswish										
conv4_1/3x3_dwconv/	Conv	8	-	6	0.5000	8	16	0.1411	0.2822	0.22
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/	Scale	16	-	16	-	16	-	0.0001	-	0.00
SE_bn										
conv4_1/3x3_dwconv/	Conv	16	-	6	0.5000	16	16	0.0002	0.0002	0.10
SE_fc1										
conv4_1/3x3_dwconv/	ReLU	16	-	-	-	16	-	0.0000	-	-
SE_fc1/relu										
conv4_1/3x3_dwconv/	Conv	16	-	6	0.5000	16	16	0.0002	0.0001	0.10
SE_fc2										
conv4_1/3x3_dwconv/	Sigmoid	16	-	-	-	16	16	0.0001	0.0001	-
SE_weights										
conv4_1/3x3_dwconv/	Scale	8	16	-	-	16	-	0.0470	-	-
scale										
conv4_1/1x1_decrease	Conv	8	-	6	0.5000	8	16	0.4704	0.9408	0.62
conv4_1/ elt_sum	Elt	8	16	-	-	-	16	-	0.0157	-
conv4_2/ 1x1_increase	Conv	8	-	6	0.5000	8	16	0.4704	0.9408	0.66
conv4_2/1x1_increase/	HardSwish	8	-	-	-	8	-	0.0706	-	-
hardswish										
conv4_2/3x3_dwconv/	Conv	8	-	6	0.5000	8	16	0.0470	0.0941	0.10
slice_0										
conv4_2/3x3_dwconv/	HardSwish	8	-	-	-	8	-	0.0353	-	-
slice_0/hardswish										
conv4_2/3x3_dwconv/	Conv	8	-	6	0.5000	8	16	0.1411	0.2822	0.22
slice_1										
conv4_2/3x3_dwconv/	HardSwish	8	-	-	-	8	-	0.0353	-	-
slice_1/hardswish										
conv4_2/3x3_dwconv/	Pool	8	-	-	-	8	16	0.0000	0.0470	-
SE_global_pool										
conv4_2/3x3_dwconv/	Scale	16	-	16	-	16	-	0.0001	-	0.00
SE_bn										
conv4_2/3x3_dwconv/	Conv	16	-	6	0.5000	16	16	0.0002	0.0002	0.10
SE_fc1										
conv4_2/3x3_dwconv/	ReLU	16	-	-	-	16	-	0.0000	-	-
SE_fc1/relu										
conv4_2/3x3_dwconv/	Conv	16	-	6	0.5000	16	16	0.0002	0.0001	0.10
SE_fc2										
conv4_2/3x3_dwconv/	Sigmoid	16	-	-	-	16	16	0.0001	0.0001	-
SE_weights							<u> </u>			
conv4_2/3x3_dwconv/	Scale	8	16	-	-	16	-	0.0470	-	-
scale						L	L			
conv4_2/1x1_decrease	Conv	8	-	6	0.5000	8	16	0.4704	0.9408	0.62
conv4_2/ elt_sum	Elt	8	16	-	-	-	16	-	0.0157	-
conv4_3/1x1_increase	Conv	8	-	6	0.5000	8	16	0.4704	0.9408	0.66
conv4_3/1x1_increase/	HardSwish	8	-	-	-	8	-	0.0706	-	-
hardswish										
conv4_3/3x3_dwconv/	Conv	8	-	6	0.5000	8	16	0.0470	0.0941	0.10
slice_0										
conv4_3/3x3_dwconv/	HardSwish	8	-	-	-	8	-	0.0353	-	-
slice_0/hardswish										
conv4_3/3x3_dwconv/	Conv	8	-	6	0.5000	8	16	0.1411	0.2822	0.22
slice_1										
conv4_3/3x3_dwconv/	HardSwish	8	-	-	-	8	-	0.0353	-	-
slice_1/hardswish										
conv4_3/3x3_dwconv/	Pool	8	-	-	-	8	16	0.0000	0.0470	-
SE_global_pool										
· .	1		1	1	1			1		

layername	type	input1	input2	weight	sparsity	mul	add	mulops(M)	addops(M)	storage(K)
conv4_3/3x3_dwconv/	Scale	16	-	16	-	16	-	0.0001	-	0.00
SE_bn										
conv4_3/3x3_dwconv/ SE_fc1	Conv	16	-	6	0.5000	16	16	0.0002	0.0002	0.10
conv4_3/3x3_dwconv/ SE_fc1/relu	ReLU	16	-	-	-	16	-	0.0000	-	-
conv4_3/3x3_dwconv/ SE_fc2	Conv	16	-	6	0.5000	16	16	0.0002	0.0001	0.10
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	-	6	0.5000	8	16	0.4704	0.9408	0.62
conv4_3/ elt_sum	Elt	8	16	-	-	-	16	-	0.0157	-
conv5_0/1x1_increase	Conv	8	-	6	0.5000	8	16	0.9408	1.8816	1.32
conv5_0/1x1_increase/ hardswish	HardSwish	8	-	-	-	8	-	0.1411	-	-
conv5_0/3x3_dwconv/ slice_0	Conv	8	-	6	0.5000	8	16	0.0157	0.0314	0.13
conv5_0/3x3_dwconv/ slice_0/hardswish	HardSwish	8	-	-	-	8	-	0.0118	-	-
conv5_0/3x3_dwconv/ slice_1	Conv	8	-	6	0.5000	8	16	0.0470	0.0941	0.29
conv5_0/3x3_dwconv/	HardSwish	8	-	-	-	8	-	0.0118	-	-
slice_1/hardswish conv5_0/3x3_dwconv/	Conv	8	-	6	0.5000	8	16	0.0941	0.1882	0.53
slice_2 conv5_0/3x3_dwconv/	HardSwish	8	-	-	-	8	-	0.0118	-	-
slice_2/hardswish conv5_0/3x3_dwconv/	Pool	8	_	_	_	8	16	0.0001	0.0234	
SE_global_pool conv5_0/3x3_dwconv/	Scale	16	_	16	_	16	-	0.0001	-	0.00
SE_bn										
conv5_0/3x3_dwconv/ SE_fc1	Conv	16	-	6	0.5000	16	16	0.0009	0.0009	0.45
conv5_0/3x3_dwconv/ SE_fc1/relu	ReLU	16	-	-	-	16	-	0.0000	-	-
conv5_0/3x3_dwconv/ SE_fc2	Conv	16	-	6	0.5000	16	16	0.0008	0.0007	0.45
conv5_0/3x3_dwconv/ SE_weights	Sigmoid	16	-	-	-	16	16	0.0002	0.0001	-
conv5_0/3x3_dwconv/	Scale	8	16	-	-	16	-	0.0235	-	-
scale	Comy	0			0.5000	0	16	0.4704	0.0409	2.44
conv5_0/ 1x1_decrease conv5_1/ 1x1_increase	Conv Conv	8	-	6	0.5000	8	16 16	0.4704 0.4704	0.9408 0.9408	2.44 2.52
conv5_1/1x1_increase/	HardSwish	8	-	-	-	8	-	0.4704	-	- 2.32
hardswish conv5_1/3x3_dwconv/	Conv	8	-	6	0.5000	8	16	0.0235	0.0470	0.20
slice_0 conv5_1/3x3_dwconv/	HardSwish	8	-	-	-	8	-	0.0176	-	
slice_0/hardswish conv5_1/3x3_dwconv/	Conv	8	_	6	0.5000	8	16	0.0706	0.1411	0.43
slice_1 conv5_1/3x3_dwconv/	HardSwish							0.0176		
slice_1/hardswish		8	-	-	-	8	-		-	-
conv5_1/3x3_dwconv/ SE_global_pool	Pool	8	-	-	-	8	16	0.0001	0.0234	-
conv5_1/3x3_dwconv/ SE_bn	Scale	16	-	16	-	16	-	0.0001	-	0.00
conv5_1/3x3_dwconv/ SE_fc1	Conv	16	-	6	0.5000	16	16	0.0009	0.0009	0.45
	L	L	1	l	L		L			

layername	type	input1	input2	weight	sparsity	mul	add	mulops(M)	addops(M)	storage(K)
conv5_1/3x3_dwconv/	ReLU	16	-	-	-	16	-	0.0000	-	-
SE_fc1/relu		1.6			0.5000	1.6	1.6	0.0000	0.0007	0.45
conv5_1/3x3_dwconv/ SE_fc2	Conv	16	-	6	0.5000	16	16	0.0008	0.0007	0.45
conv5_1/3x3_dwconv/ SE_weights	Sigmoid	16	-	-	-	16	16	0.0002	0.0001	-
conv5_1/3x3_dwconv/	Scale	8	16	-	-	16	-	0.0235	-	-
scale conv5_1/1x1_decrease	Conv	8	_	6	0.5000	8	16	0.4704	0.9408	2.44
conv5_1/ elt_sum	Elt	8	16	-	-	-	16	-	0.0078	-
conv5_2/ 1x1_increase	Conv	8	-	6	0.5000	8	16	0.4704	0.9408	2.52
conv5_2/1x1_increase/	HardSwish	8		-	-	8		0.4704	-	-
hardswish	HardSwish	8	-	-	-	8	-	0.0333	-	-
conv5_2/3x3_dwconv/ slice_0	Conv	8	-	6	0.5000	8	16	0.0235	0.0470	0.20
conv5_2/3x3_dwconv/	HardSwish	8	_	_	_	8	-	0.0176	_	_
slice_0/hardswish			_							
conv5_2/3x3_dwconv/ slice_1	Conv	8	-	6	0.5000	8	16	0.0706	0.1411	0.43
conv5_2/3x3_dwconv/	HardSwish	8	_	-	-	8	-	0.0176	-	-
slice_1/hardswish conv5_2/3x3_dwconv/	Pool	8				8	16	0.0001	0.0234	
SE_global_pool		8	-	-	-	8	10		0.0234	-
conv5_2/3x3_dwconv/ SE_bn	Scale	16	-	16	-	16	-	0.0001	-	0.00
conv5_2/3x3_dwconv/	Conv	16	-	6	0.5000	16	16	0.0009	0.0009	0.45
SE_fc1 conv5_2/3x3_dwconv/	ReLU	16	-	-	-	16	-	0.0000	-	-
SE_fc1/relu		16			0.5000	16	16		0.0007	0.45
conv5_2/3x3_dwconv/ SE_fc2	Conv	16	-	6	0.5000	16	16	0.0008	0.0007	0.45
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	_	6	0.5000	8	16	0.4704	0.9408	2.44
conv5_2/ elt_sum	Elt	8	16	U	0.3000	-	16	0.4704	0.0078	2.44
conv5_3/ 1x1_increase	Conv		-	6	0.5000	8	16	0.4704	0.0078	2.52
conv3_3/ 1x1_increase		8								
conv5_3/1x1_increase/ hardswish	HardSwish	8	-	-	-	8	-	0.0353	-	-
conv5_3/3x3_dwconv/	Conv	8	-	6	0.5000	8	16	0.0235	0.0470	0.20
slice_0 conv5_3/3x3_dwconv/	HardSwich	8	_	_	_	8		0.0176	_	_
slice_0/hardswish	Haluswish	0	_	_	_		_		-	-
conv5_3/3x3_dwconv/ slice_1	Conv	8	-	6	0.5000	8	16	0.0706	0.1411	0.43
conv5_3/3x3_dwconv/	HardSwish	8	-	-	-	8	-	0.0176	-	-
slice_1/hardswish conv5_3/3x3_dwconv/	Pool	8	_	_	_	8	16	0.0001	0.0234	-
SE_global_pool										
conv5_3/3x3_dwconv/ SE_bn	Scale	16	-	16	-	16	-	0.0001	-	0.00
conv5_3/3x3_dwconv/ SE_fc1	Conv	16	-	6	0.5000	16	16	0.0009	0.0009	0.45
conv5_3/3x3_dwconv/ SE_fc1/relu	ReLU	16	-	-	-	16	-	0.0000	-	-
conv5_3/3x3_dwconv/ SE_fc2	Conv	16	-	6	0.5000	16	16	0.0008	0.0007	0.45
conv5_3/3x3_dwconv/	Sigmoid	16	-	-	-	16	16	0.0002	0.0001	-
SE_weights conv5_3/3x3_dwconv/	Scale	8	16	-	-	16	-	0.0235	-	-
scale										

Summon S	layername	type	input1	input2	weight	sparsity	mul	add	mulops(M)	addops(M)	storage(K)
Convo O/181 increase Conv	conv5_3/1x1_decrease			-							
Conv6 O/134 increase hardSwish 8	conv5_3/ elt_sum	Elt	8	16	-	-	-	16	-	0.0078	-
hardswish	conv6_0/ 1x1_increase	Conv	8	-	6	0.5000	8	16	0.9408	1.8816	5.04
Come G/333 dwconv Conv S	conv6_0/1x1_increase/	HardSwish	8	-	-	-	8	-	0.0706	-	-
Since_O HardSwish S	hardswish										
Comedic Come	conv6_0/3x3_dwconv/	Conv	8	-	6	0.5000	8	16	0.0314	0.0627	0.26
Sice_OhracIswish	slice_0										
Sice_OhracIswish	conv6_0/3x3_dwconv/	HardSwish	8	-	-	-	8	-	0.0235	-	-
Common C											
Sice_1		Conv	8	-	6	0.5000	8	16	0.0941	0.1882	0.58
Common C											
Sice_Li/hardswish		HardSwish	8	-	_	_	8	-	0.0235	_	_
Comv6_0/33_d dvconv/ Comv Second Comv Com									*******		
Slice_2		Conv	8	_	6	0.5000	8	16	0.1882	0.3763	1.06
Second Company Compa									0.1000	3.0.0	
Sice_2Dhardswish		HardSwish	8	_	_	_	8	-	0.0235	_	_
Conv. Conv		Tital GO WISH							0.0233		
SE_global_pool		Pool	8	_	_	_	8	16	0.0001	0.0468	_
Convolution		1 001						10	0.0001	0.0100	
SE_bn	conv6 0/3v3 dwconv/	Scale	16	_	16	_	16	_	0.0002	_	0.00
Conv6_0/3x3_dwconv/ Conv 16		Scare	10	_	10		10	_	0.0002	_	0.00
SE_fcI		Conv	16		6	0.5000	16	16	0.0036	0.0036	1.80
Conv6_0/3x3_dwconv/ Conv 16 - - - - 16 - 0.0000 - - -		Conv	10	_	0	0.5000	10	10	0.0030	0.0030	1.60
SE_fcl/relu		Dalii	16				16		0.0000		
Conv6_0/3x3_dwconv/ Conv 16		ReLU	10	_	_	_	10	_	0.0000	-	_
SE_fc2		Conv	16		6	0.5000	16	16	0.0026	0.0024	1.90
Convo. O/3x3_dwconv/ Sigmoid 16 - - - 16 16 0.0005 0.0002 -		Conv	10	_	U	0.3000	10	10	0.0030	0.0034	1.60
SE_weights		Ciamoid	16				16	16	0.0005	0.0002	
Conv6_0/3x3_dwconv/ Scale 8		Signiola	10	_	-	_	10	10	0.0003	0.0002	-
Scale		0 1	0	1.0			1.6		0.0470		
Conv6_0/		Scale	8	10	-	_	10	-	0.0470	-	-
Conv6_1/1x1_increase		Comy	0		-	0.5000	0	16	1 1200	2.2570	5 01
Conv6_1/1x1_increase/ hardswish S											
hardswish conv6_1/3x3_dwconv/ conv 8 - 6 0.5000 8 16 0.0188 0.0376 0.16							1				
Conv6_1/3x3_dwconv/ Conv S -		HardSwish	8	-	-	-	8	-	0.0423	-	-
Slice_0 Oonv6_1/3x3_dwconv/ HardSwish 8 - - - 8 - 0.0141 - - -			0			0.5000	0	16	0.0100	0.0276	0.16
Conv6_1/3x3_dwconv/ Sice_0/hardswish Sice_0/hardswish Sice_0/hardswish Sice_1 Conv6_1/3x3_dwconv/ Sice_1 Conv6_1/3x3_dwconv/ Sice_1 Conv6_1/3x3_dwconv/ Sice_1 Sice_1 Conv6_1/3x3_dwconv/ Sice_1/hardswish Sice_1/hardswish Sice_1/hardswish Sice_1/hardswish Sice_1/hardswish Sice_1/hardswish Sice_2/hardswish Sice_1/hardswish Sice_1/h		Conv	8	-	6	0.5000	8	16	0.0188	0.0376	0.16
Slice_0/hardswish Conv		11 10 11	0						0.0141		
Conv6_1/3x3_dwconv/ slice_1		HardSwish	8	-	-	-	8	-	0.0141	-	-
Slice_I Conv6_1/3x3_dwconv/ Scale Slice_I/hardswish Slic		G	0			0.5000		1.0	0.0564	0.1120	0.25
Conv6_1/3x3_dwconv/ Scale Seconv6_1/3x3_dwconv/ SE_fc1/relu Conv6_1/3x3_dwconv/ SE_fc2 Conv6_1/3x3_dwconv/ Sigmoid Seconv6_1/3x3_dwconv/ Sigmoid Scale Seconv6_1/3x3_dwconv/ Seconv6_1/3x3		Conv	8	-	6	0.5000	8	16	0.0564	0.1129	0.35
Slice_1/hardswish		11 10 11	0						0.0141		
conv6_1/3x3_dwconv/ slice_2 Conv 8 - 6 0.5000 8 16 0.1129 0.2258 0.64 conv6_1/3x3_dwconv/ slice_2/hardswish HardSwish 8 - - - 8 - 0.0141 - - conv6_1/3x3_dwconv/ SE_global_pool Pool 8 - - - 8 16 0.0001 0.0281 - conv6_1/3x3_dwconv/ SE_bn Scale 16 - 16 - 16 - 0.0001 - 0.00 sE_fc1 conv6_1/3x3_dwconv/ SE_fc1/relu ReLU 16 - - - 16 0.0003 0.0013 0.0012 0.65 sE_fc2 conv6_1/3x3_dwconv/ SE_weights Sigmoid 16 - - - - 16 0.0003 0.0001 - - conv6_1/3x3_dwconv/ SE_weights Scale 8 16 - - - - - - - - - - - <td></td> <td>HardSwish</td> <td>8</td> <td>-</td> <td>-</td> <td>-</td> <td>8</td> <td>-</td> <td>0.0141</td> <td>-</td> <td>-</td>		HardSwish	8	-	-	-	8	-	0.0141	-	-
Slice_2 Slic			0			0.5000		16	0.1120	0.2250	0.64
conv6_1/3x3_dwconv/ slice_2/hardswish HardSwish 8 - - - 8 - 0.0141 - - conv6_1/3x3_dwconv/ SE_global_pool Pool 8 - - 8 16 0.0001 0.0281 - conv6_1/3x3_dwconv/ SE_bn Scale 16 - 16 - 16 - 0.0001 - 0.00 SE_bn - 6 0.5000 16 16 0.0013 0.0013 0.065 SE_fc1 -		Conv	8	-	6	0.5000	8	16	0.1129	0.2258	0.64
Slice_2/hardswish		II. 10							0.0141		
conv6_1/3x3_dwconv/ SE_global_pool Pool 8 - - - 8 16 0.0001 0.0281 - conv6_1/3x3_dwconv/ SE_bn Scale 16 - 16 - 16 - 0.0001 - 0.00 SE_bn Conv6_1/3x3_dwconv/ Conv 16 - 6 0.5000 16 16 0.0013 0.0013 0.65 SE_fc1 Conv6_1/3x3_dwconv/ ReLU 16 - - - 16 - 0.0000 - - - - SE_fc1/relu Conv6_1/3x3_dwconv/ Sigmoid 16 - <td< td=""><td></td><td>HardSwish</td><td>8</td><td>-</td><td>-</td><td>-</td><td> 8</td><td> -</td><td>0.0141</td><td>-</td><td>-</td></td<>		HardSwish	8	-	-	-	8	-	0.0141	-	-
SE_global_pool Conv6_1/3x3_dwconv/ Scale 16 - 16 - 16 - 0.0001 - 0.000 SE_bn Conv6_1/3x3_dwconv/ Conv 16 - 6 0.5000 16 16 0.0013 0.0013 0.0013 0.65 SE_fc1 Conv6_1/3x3_dwconv/ ReLU 16 - - - 16 - 0.0000 - - - SE_fc1/relu Conv6_1/3x3_dwconv/ Conv 16 - 6 0.5000 16 16 0.0013 0.0012 0.65 SE_fc2 Conv6_1/3x3_dwconv/ Sigmoid 16 - - - 16 16 0.0003 0.0001 - SE_weights Conv6_1/3x3_dwconv/ Scale 8 16 - - 16 - 0.0282 - - -		D 1	0				0	17	0.0001	0.0201	
conv6_1/3x3_dwconv/ SE_bn Scale 16 - 16 - 16 - 0.0001 - 0.00 SE_bn Conv6_1/3x3_dwconv/ SE_fc1 Conv 16 - 6 0.5000 16 16 0.0013 0.0013 0.65 SE_fc1 Conv6_1/3x3_dwconv/ SE_fc2 ReLU 16 - - - 16 - 0.0000 - - - 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 - - -		Pool	8	-	-	_	8	16	0.0001	0.0281	-
SE_bn Conv6_1/3x3_dwconv/ Conv 16 - 6 0.5000 16 16 0.0013 0.0013 0.0013 0.65 SE_fc1 conv6_1/3x3_dwconv/ ReLU 16 - - - 16 - 0.0000 - - - SE_fc1/relu conv6_1/3x3_dwconv/ Conv 16 - 6 0.5000 16 16 0.0013 0.0012 0.65 SE_fc2 conv6_1/3x3_dwconv/ Sigmoid 16 - - - 16 16 0.0003 0.0001 - SE_weights conv6_1/3x3_dwconv/ Scale 8 16 - - 16 - 0.0282 - - scale 16 - - - 16 - 0.0282 - -		0.1	1.0		1.0		16		0.0001		0.00
conv6_1/3x3_dwconv/ Conv 16 - 6 0.5000 16 16 0.0013 0.0013 0.0013 0.65 SE_fc1 conv6_1/3x3_dwconv/ ReLU 16 - - - 16 - 0.0000 - - - SE_fc1/relu conv6_1/3x3_dwconv/ Conv 16 - 6 0.5000 16 16 0.0013 0.0012 0.65 SE_fc2 conv6_1/3x3_dwconv/ Sigmoid 16 - - - 16 16 0.0003 0.0001 - SE_weights conv6_1/3x3_dwconv/ Scale 8 16 - - 16 - 0.0282 - - scale 16 - - - 16 - 0.0282 - -		Scale	16	-	16	-	16	-	0.0001	-	0.00
SE_fc1 conv6_1/3x3_dwconv/ ReLU 16 - - - 16 - 0.0000 - - - SE_fc1/relu conv6_1/3x3_dwconv/ Conv 16 - 6 0.5000 16 16 0.0013 0.0012 0.65 SE_fc2 conv6_1/3x3_dwconv/ Sigmoid 16 - - - 16 16 0.0003 0.0001 - SE_weights conv6_1/3x3_dwconv/ Scale 8 16 - - 16 - 0.0282 - - scale - - - 16 - 0.0282 - -		C	17			0.5000	16	1.0	0.0012	0.0012	0.65
conv6_1/3x3_dwconv/ SE_fc1/relu ReLU 16 - - - 16 - 0.0000 -		Conv	16	-	6	0.5000	16	16	0.0013	0.0013	0.65
SE_fc1/relu Conv6_1/3x3_dwconv/ Conv 16 - 6 0.5000 16 16 0.0013 0.0012 0.65 SE_fc2 conv6_1/3x3_dwconv/ Sigmoid 16 - - - 16 16 0.0003 0.0001 - SE_weights conv6_1/3x3_dwconv/ Scale 8 16 - - 16 - 0.0282 - - scale - - - - 16 - 0.0282 - - -		D ***	1.				1.		0.0000		
conv6_1/3x3_dwconv/ Conv 16 - 6 0.5000 16 16 0.0013 0.0012 0.65 SE_fc2 conv6_1/3x3_dwconv/ Sigmoid 16 - - - 16 16 0.0003 0.0001 - SE_weights conv6_1/3x3_dwconv/ Scale 8 16 - - 16 - 0.0282 - - scale scale - - - 16 - 0.0282 - -		ReLU	16	-	-	-	16	-	0.0000	-	-
SE_fc2 conv6_1/3x3_dwconv/ Sigmoid 16 - - - 16 16 0.0003 0.0001 - SE_weights conv6_1/3x3_dwconv/ Scale 8 16 - - 16 - 0.0282 - - scale scale - - - - - - - - -						0.70			0.0015	0.0015	6.55
conv6_1/3x3_dwconv/ Sigmoid 16 - - - 16 16 0.0003 0.0001 - SE_weights conv6_1/3x3_dwconv/ Scale 8 16 - - 16 - 0.0282 - - scale scale -<		Conv	16	-	6	0.5000	16	16	0.0013	0.0012	0.65
SE_weights											
conv6_1/3x3_dwconv/ Scale 8 16 16 - 0.0282 scale		Sigmoid	16	-	-	-	16	16	0.0003	0.0001	-
scale	SE_weights										
		Scale	8	16	-	-	16	-	0.0282	- 7	-
conv6_1/1x1_decrease Conv 8 - 6 0.5000 8 16 0.6774 1.3548 3.50											
	conv6_1/1x1_decrease	Conv	8	-	6	0.5000	8	16	0.6774	1.3548	3.50

layername	type	input1	input2	weight	sparsity	mul	add	mulops(M)	addops(M)	storage(K)
conv6_1/ elt_sum	Elt	8	16	-	-	-	16	-	0.0094	-
conv6_2/1x1_increase	Conv	8	-	6	0.5000	8	16	0.6774	1.3548	3.60
conv6_2/1x1_increase/	HardSwish	8	-	-	-	8	-	0.0423	-	-
hardswish										
conv6_2/3x3_dwconv/	Conv	8	-	6	0.5000	8	16	0.0188	0.0376	0.16
slice_0								0.000		
conv6_2/3x3_dwconv/	HardSwish	8	_	_	-	8	-	0.0141	_	_
slice_0/hardswish	Tiaras wish							0.0111		
conv6_2/3x3_dwconv/	Conv	8	-	6	0.5000	8	16	0.0564	0.1129	0.35
slice_1	Conv				0.5000		10	0.0501	0.1129	0.55
conv6_2/3x3_dwconv/	HardSwish	8	_	_	_	8	-	0.0141	_	_
slice_1/hardswish	Tiard5 Wish	0	_	_		"		0.0141	_	_
conv6_2/3x3_dwconv/	Conv	8	_	6	0.5000	8	16	0.1129	0.2258	0.64
slice_2	Conv	0	_	0	0.5000	"	10	0.1127	0.2230	0.04
conv6_2/3x3_dwconv/	HardSwish	8	_	_	_	8	-	0.0141	_	_
slice_2/hardswish	HaluSwisii	0	_	_	_	0	-	0.0141	-	-
	D 1	0				0	16	0.0001	0.0201	
conv6_2/3x3_dwconv/	Pool	8	-	-	-	8	16	0.0001	0.0281	-
SE_global_pool	0.1	1.6		16		16		0.0001		0.00
conv6_2/3x3_dwconv/	Scale	16	-	16	-	16	-	0.0001	-	0.00
SE_bn					0.7000			0.0013	0.0013	
conv6_2/3x3_dwconv/	Conv	16	-	6	0.5000	16	16	0.0013	0.0013	0.65
SE_fc1										
conv6_2/3x3_dwconv/	ReLU	16	-	-	-	16	-	0.0000	-	-
SE_fc1/relu										
conv6_2/3x3_dwconv/	Conv	16	-	6	0.5000	16	16	0.0013	0.0012	0.65
SE_fc2										
conv6_2/3x3_dwconv/	Sigmoid	16	-	-	-	16	16	0.0003	0.0001	-
SE_weights										
conv6_2/3x3_dwconv/	Scale	8	16	-	-	16	-	0.0282	-	-
scale										
conv6_2/1x1_decrease	Conv	8	-	6	0.5000	8	16	0.6774	1.3548	3.50
conv6_2/ elt_sum	Elt	8	16	-	-	-	16	-	0.0094	-
conv6_3/ 1x1_increase	Conv	8	-	6	0.5000	8	16	0.6774	1.3548	3.60
conv6_3/1x1_increase/	HardSwish	8	-	-	-	8	-	0.0423	_	_
hardswish								***************************************		
conv6_3/3x3_dwconv/	Conv	8	_	6	0.5000	8	16	0.0188	0.0376	0.16
slice_0	Conv				0.2000		10	0.0100	0.0270	0.10
conv6_3/3x3_dwconv/	HardSwish	8	-	-	-	8	-	0.0141	_	_
slice_0/hardswish	Tital dis Wish							0.0111		
conv6_3/3x3_dwconv/	Conv	8	_	6	0.5000	8	16	0.0564	0.1129	0.35
slice_1	Conv	0	-	U	0.3000	0	10	0.0304	0.1129	0.55
conv6_3/3x3_dwconv/	HardSwish	8		_		8		0.0141	_	_
slice_1/hardswish	HaluSwisii	0	-	_	-	0	-	0.0141	-	-
	C	0		-	0.5000	0	16	0.1120	0.2250	0.64
conv6_3/3x3_dwconv/	Conv	8	-	6	0.5000	8	16	0.1129	0.2258	0.64
slice_2	II 10 . 1	0	1			0		0.0141		
conv6_3/3x3_dwconv/	HardSwish	8	-	-	-	8	-	0.0141	-	-
slice_2/hardswish	D 1						1.5	0.0004	0.0001	
conv6_3/3x3_dwconv/	Pool	8	-	-	-	8	16	0.0001	0.0281	-
SE_global_pool										
conv6_3/3x3_dwconv/	Scale	16	-	16	-	16	-	0.0001	-	0.00
SE_bn										
conv6_3/3x3_dwconv/	Conv	16	-	6	0.5000	16	16	0.0013	0.0013	0.65
SE_fc1			<u></u>	<u> </u>		<u></u>				
conv6_3/3x3_dwconv/	ReLU	16	-	-	-	16	-	0.0000	-	-
SE_fc1/relu										
conv6_3/3x3_dwconv/	Conv	16	-	6	0.5000	16	16	0.0013	0.0012	0.65
SE_fc2										
conv6_3/3x3_dwconv/	Sigmoid	16	-	-	-	16	16	0.0003	0.0001	-
SE_weights										
conv6_3/3x3_dwconv/	Scale	8	16	-	-	16	-	0.0282	-	-
scale										
conv6_3/1x1_decrease	Conv	8	-	6	0.5000	8	16	0.6774	1.3548	3.50
John J. J. Tal_decrease	LONV				0.5000		1.0	0.0771	1.55 10	3.30

layername	type	input1	input2	weight	sparsity	mul	add	mulops(M)	addops(M)	storage(K)
conv6_3/ elt_sum	Elt	8	16	-	-	-	16	-	0.0094	-
conv6_4/ 1x1_increase	Conv	8	-	6	0.5000	8	16	0.6774	1.3548	3.60
conv6_4/1x1_increase/	HardSwish	8	-	-	-	8	-	0.0423	-	-
hardswish										
conv6_4/3x3_dwconv/	Conv	8	-	6	0.5000	8	16	0.0188	0.0376	0.16
slice_0										
conv6_4/3x3_dwconv/	HardSwish	8	-	-	-	8	-	0.0141	-	-
slice_0/hardswish										
conv6_4/3x3_dwconv/	Conv	8	-	6	0.5000	8	16	0.0564	0.1129	0.35
slice_1										
conv6_4/3x3_dwconv/	HardSwish	8	_	-	_	8	-	0.0141	_	_
slice_1/hardswish								0.00		
conv6_4/3x3_dwconv/	Conv	8	_	6	0.5000	8	16	0.1129	0.2258	0.64
slice 2	Conv				0.5000		10	0.1125	0.2230	0.01
conv6_4/3x3_dwconv/	HardSwish	8	_	_	_	8	-	0.0141	_	_
slice_2/hardswish	HaiuSwisii	0	_	_	_	8	-	0.0141	-	_
conv6_4/3x3_dwconv/	Pool	8				8	16	0.0001	0.0281	_
	Pool	0	-	-	-	°	10	0.0001	0.0281	-
SE_global_pool	0.1	1.6		16		16		0.0001		0.00
conv6_4/3x3_dwconv/	Scale	16	-	16	-	16	-	0.0001	-	0.00
SE_bn										
conv6_4/3x3_dwconv/	Conv	16	-	6	0.5000	16	16	0.0013	0.0013	0.65
SE_fc1										
conv6_4/3x3_dwconv/	ReLU	16	-	-	-	16	-	0.0000	-	-
SE_fc1/relu										
conv6_4/3x3_dwconv/	Conv	16	-	6	0.5000	16	16	0.0013	0.0012	0.65
SE_fc2										
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	Scare		10			10		0.0202		
conv6_4/ 1x1_decrease	Conv	8	_	6	0.5000	8	16	0.6774	1.3548	3.50
conv6_4/ elt_sum	Elt	8	16	-	-	-	16	-	0.0094	-
conv6_5/ 1x1_increase	Conv	8	-	6	0.5000	8	16	0.6774	1.3548	3.60
conv6_5/1x1_increase/	HardSwish	8	_	-		8	-	0.0774	-	
	naruswisii	0	-	-	-	°	-	0.0423	-	-
hardswish		0			0.5000		16	0.0100	0.0276	0.16
conv6_5/3x3_dwconv/	Conv	8	-	6	0.5000	8	16	0.0188	0.0376	0.16
slice_0										
conv6_5/3x3_dwconv/	HardSwish	8	-	-	-	8	-	0.0141	-	-
slice_0/hardswish										
conv6_5/3x3_dwconv/	Conv	8	-	6	0.5000	8	16	0.0564	0.1129	0.35
slice_1										
conv6_5/3x3_dwconv/	HardSwish	8	-	-	-	8	-	0.0141	-	-
slice_1/hardswish										
conv6_5/3x3_dwconv/	Conv	8	-	6	0.5000	8	16	0.1129	0.2258	0.64
slice_2										
conv6_5/3x3_dwconv/	HardSwish	8	_	-	-	8	-	0.0141	-	-
slice_2/hardswish						_				
conv6_5/3x3_dwconv/	Pool	8	_	_	_	8	16	0.0001	0.0281	_
SE_global_pool	1001					"	10	0.0001	0.0201	
conv6_5/3x3_dwconv/	Scale	16	_	16	_	16	_	0.0001	_	0.00
	Scale	10	_	10	_	10	-	0.0001	-	0.00
SE_bn conv6_5/3x3_dwconv/	Conv	16		6	0.5000	16	16	0.0013	0.0013	0.65
	Conv	10	-	0	0.3000	10	10	0.0013	0.0013	0.65
SE_fc1						1.5		0.0000		
conv6_5/3x3_dwconv/	ReLU	16	-	-	-	16	-	0.0000	-	-
SE_fc1/relu										
conv6_5/3x3_dwconv/	Conv	16	-	6	0.5000	16	16	0.0013	0.0012	0.65
SE_fc2										
conv6_5/3x3_dwconv/	Sigmoid	16	-	-	-	16	16	0.0003	0.0001	-
SE_weights										
conv6_5/3x3_dwconv/	Scale	8	16	-	-	16	-	0.0282	-	-
scale										
conv6_5/ 1x1_decrease	Conv	8	-	6	0.5000	8	16	0.6774	1.3548	3.50
tonio_o/ ixi_decrease	23111		1		0.5000		10	0.0771	1.55 10	5.50

layername	type	input1	input2	weight	sparsity	mul	add	mulops(M)	addops(M)	storage(K)
conv6_5/ elt_sum	Elt	8	16	-	-	-	16	-	0.0094	-
conv7_0/ 1x1_increase	Conv	8	-	6	0.4500	8	16	1.4676	2.9353	7.72
conv7_0/1x1_increase/	HardSwish	8	_	_	-	8	-	0.0847	-	
hardswish										
conv7_0/3x3_dwconv/	Conv	8	_	6	0.4500	8	16	0.0071	0.0141	0.25
slice_0	Conv				0.1500		10	0.0071	0.0111	0.23
conv7_0/3x3_dwconv/	HardSwish	8	_	_	_	8	-	0.0053	_	
slice_0/hardswish	Haluswish	0	_	_	_	0	_	0.0055	-	-
	C	8		6	0.4500	8	1.6	0.0229	0.0459	0.56
conv7_0/3x3_dwconv/	Conv	8	-	0	0.4500	8	16	0.0229	0.0459	0.56
slice_1	11 10 11	0				0		0.0052		
conv7_0/3x3_dwconv/	HardSwish	8	-	-	-	8	-	0.0053	-	-
slice_1/hardswish					0.1500			0.0150	0.001=	
conv7_0/3x3_dwconv/	Conv	8	-	6	0.4500	8	16	0.0459	0.0917	1.02
slice_2										
conv7_0/3x3_dwconv/	HardSwish	8	-	-	-	8	-	0.0053	-	-
slice_2/hardswish										
conv7_0/3x3_dwconv/	Conv	8	-	6	0.4500	8	16	0.0776	0.1552	1.64
slice_3										
conv7_0/3x3_dwconv/	HardSwish	8	-	-	-	8	-	0.0053	-	-
slice_3/hardswish										
conv7_0/3x3_dwconv/	Pool	8	-	-	-	8	16	0.0001	0.0138	-
SE_global_pool										
conv7_0/3x3_dwconv/	Scale	16	-	16	_	16	-	0.0003	_	0.00
SE_bn	Jeane	10		10		10		0.0000		0.00
conv7_0/3x3_dwconv/	Conv	16	-	6	0.5000	16	16	0.0052	0.0052	2.59
SE_fc1	Conv	10	_	0	0.5000	10	10	0.0032	0.0032	2.59
conv7_0/3x3_dwconv/	ReLU	16	_	_	_	16	-	0.0000	_	
	ReLU	10	-	-	_	10	-	0.0000	-	-
SE_fc1/relu		1.6		-	0.5000	16	1.6	0.0052	0.0040	2.50
conv7_0/3x3_dwconv/	Conv	16	-	6	0.5000	16	16	0.0052	0.0049	2.59
SE_fc2	ļ							0.000	0.000	
conv7_0/3x3_dwconv/	Sigmoid	16	-	-	-	16	16	0.0006	0.0003	-
SE_weights										
conv7_0/3x3_dwconv/	Scale	8	16	-	-	16	-	0.0141	-	-
scale										
conv7_0/ 1x1_decrease	Conv	8	-	6	0.4500	8	16	0.7432	1.4865	14.96
conv7_1/1x1_increase	Conv	8	-	6	0.4500	8	16	0.7409	1.4818	15.15
conv7_1/1x1_increase/	HardSwish	8	-	-	-	8	-	0.0212	-	-
hardswish										
conv7_1/3x3_dwconv/	Conv	8	-	6	0.4500	8	16	0.0094	0.0188	0.33
slice_0										
conv7_1/3x3_dwconv/	HardSwish	8	-	-	_	8	-	0.0071	_	-
slice_0/hardswish										
conv7_1/3x3_dwconv/	Conv	8	_	6	0.4500	8	16	0.0306	0.0612	0.74
slice_1	Conv				0.1500		10	0.0300	0.0012	0.71
conv7_1/3x3_dwconv/	HardSwish	8		_	_	8	-	0.0071	_	_
slice_1/hardswish	Tiaius wisii	0	_	_	_	0	_	0.0071	-	-
	Conv	8		6	0.4500	8	16	0.0612	0.1223	1.36
conv7_1/3x3_dwconv/	Conv	٥	-	U	0.4300	°	16	0.0012	0.1223	1.30
slice_2	1110 : 1	0				0		0.0071		
conv7_1/3x3_dwconv/	HardSwish	8	_	-	-	8	-	0.0071	-	-
slice_2/hardswish								0.0001	0.0122	
conv7_1/3x3_dwconv/	Pool	8	-	-	-	8	16	0.0001	0.0138	-
SE_global_pool										
conv7_1/3x3_dwconv/	Scale	16	-	16	-	16	-	0.0003	-	0.00
SE_bn						<u></u>				
conv7_1/3x3_dwconv/	Conv	16	-	6	0.5000	16	16	0.0052	0.0052	2.59
SE_fc1										
conv7_1/3x3_dwconv/	ReLU	16	-	-	-	16	-	0.0000	-	-
SE_fc1/relu										
conv7_1/3x3_dwconv/	Conv	16	-	6	0.5000	16	16	0.0052	0.0049	2.59
SE_fc2		_		_		~	~			
conv7_1/3x3_dwconv/	Sigmoid	16	-	_	-	16	16	0.0006	0.0003	_
SE_weights	Signiola	10				10	10	0.0000	0.0003	
JL_weights	L		1		1		l			

layername	type	input1	input2	weight	sparsity	mul	add	mulops(M)	addops(M)	storage(K)
conv7_1/3x3_dwconv/	Scale	8	16	-	-	16	-	0.0141	-	-
scale		0			0.4500	0	17	0.7422	1 4065	14.06
conv7_1/1x1_decrease conv7_1/elt_sum	Conv Elt	8	16	6	0.4500	8 -	16 16	0.7432	1.4865 0.0047	14.96
conv7_2/ 1x1_increase	Conv	8	-	6	0.4500	8	16	0.7409	1.4818	15.15
conv7_2/1x1_increase/	HardSwish	8	_	-	-	8	-	0.0212	-	-
hardswish										
conv7_2/3x3_dwconv/	Conv	8	-	6	0.4500	8	16	0.0094	0.0188	0.33
slice_0										
conv7_2/3x3_dwconv/	HardSwish	8	-	-	-	8	-	0.0071	-	-
slice_0/hardswish conv7_2/3x3_dwconv/	Conv	8		6	0.4500	8	16	0.0306	0.0612	0.74
slice_1	Conv	٥	-	0	0.4300	°	10	0.0300	0.0012	0.74
conv7_2/3x3_dwconv/	HardSwish	8	_	_	_	8	-	0.0071	_	_
slice_1/hardswish										
conv7_2/3x3_dwconv/	Conv	8	-	6	0.4500	8	16	0.0612	0.1223	1.36
slice_2										
conv7_2/3x3_dwconv/	HardSwish	8	-	-	-	8	-	0.0071	-	-
slice_2/hardswish conv7_2/3x3_dwconv/	Pool	8				8	16	0.0001	0.0138	_
SE_global_pool	Pool	8	-	-	-	8	16	0.0001	0.0138	-
conv7_2/3x3_dwconv/	Scale	16	_	16	_	16	-	0.0003	_	0.00
SE_bn	Seare	10		10		10		0.0002		0.00
conv7_2/3x3_dwconv/	Conv	16	-	6	0.5000	16	16	0.0052	0.0052	2.59
SE_fc1										
conv7_2/3x3_dwconv/	ReLU	16	-	-	-	16	-	0.0000	-	-
SE_fc1/relu	Comm	16		-	0.5000	16	16	0.0052	0.0040	2.50
conv7_2/3x3_dwconv/ SE_fc2	Conv	16	-	6	0.5000	16	16	0.0052	0.0049	2.59
conv7_2/3x3_dwconv/	Sigmoid	16	_	_	_	16	16	0.0006	0.0003	_
SE_weights	Sigmoid	10				10	10	0.0000	0.0003	
conv7_2/3x3_dwconv/	Scale	8	16	-	-	16	-	0.0141	-	-
scale										
conv7_2/1x1_decrease	Conv	8	-	6	0.4500	8	16	0.7432	1.4865	14.96
conv7_2/ elt_sum	Elt	8	16	-	- 0.4500	-	16	- 1 4010	0.0047	-
conv8_0/1x1_increase	Conv	8	-	6	0.4500	8	16	1.4818	2.9635	30.30
conv8_0/1x1_increase/ hardswish	HardSwish	8	-	-	-	8	-	0.0423	-	-
conv8_0/3x3_dwconv/	Conv	8	_	6	0.4500	8	16	0.0188	0.0376	0.66
slice_0	Conv				0.1500		10	0.0100	0.0370	0.00
conv8_0/3x3_dwconv/	HardSwish	8	-	-	-	8	-	0.0141	-	-
slice_0/hardswish										
conv8_0/3x3_dwconv/	Conv	8	-	6	0.4500	8	16	0.0612	0.1223	1.48
slice_1	II 10 : 1	0				0		0.0141		
conv8_0/3x3_dwconv/ slice_1/hardswish	HardSwish	8	-	-	-	8	-	0.0141	-	-
conv8_0/3x3_dwconv/	Conv	8	_	6	0.4500	8	16	0.1223	0.2446	2.72
slice_2					000		10	0.1220	0.2	
conv8_0/3x3_dwconv/	HardSwish	8	-	-	-	8	-	0.0141	-	-
slice_2/hardswish										
conv8_0/3x3_dwconv/	Pool	8	-	-	-	8	16	0.0003	0.0276	-
SE_global_pool conv8_0/3x3_dwconv/	C1-	16		16		16		0.0006	_	0.00
SE_bn	Scale	16	-	16	-	16	-	0.0006	-	0.00
conv8_0/3x3_dwconv/	Conv	16	_	6	0.5000	16	16	0.0207	0.0207	10.37
SE_fc1					0.5000	10		5.0207	5.0207	10.57
conv8_0/3x3_dwconv/	ReLU	16	-	-	-	16	-	0.0000	-	-
SE_fc1/relu										
conv8_0/3x3_dwconv/	Conv	16	-	6	0.5000	16	16	0.0207	0.0202	10.37
SE_fc2	C:	17				17	17	0.0012	0.0007	
conv8_0/3x3_dwconv/ SE_weights	Sigmoid	16	-	-	-	16	16	0.0012	0.0006	-
DE_weights	<u> </u>				<u> </u>					

layername	type	input1	input2	weight	sparsity	mul	add	mulops(M)	addops(M)	storage(K)
conv8_0/3x3_dwconv/	Scale	8	16	-	-	16	-	0.0282	-	-
scale										
conv8_0/1x1_decrease	Conv	8	-	6	0.4500	8	16	2.4814	4.9627	49.70
conv9/1x1	Conv	8	-	6	0.4500	8	16	2.7597	5.5194	55.68
conv9/1x1/ relu	ReLU	8	-	-	-	8	-	0.0157	-	-
avepool/ 7x7	Pool	8	-	-	-	8	16	0.0003	0.0307	-
classifier	Conv	8	-	6	0.7000	8	16	0.0960	0.1920	112.50