Searching for MobileNetV3

PR-SMARCLE
발표자 : 심동현
2021.07.22

SMARCLE

Recall

What Should be the Appropriate Model for Mobile Devices?

Desirable Property

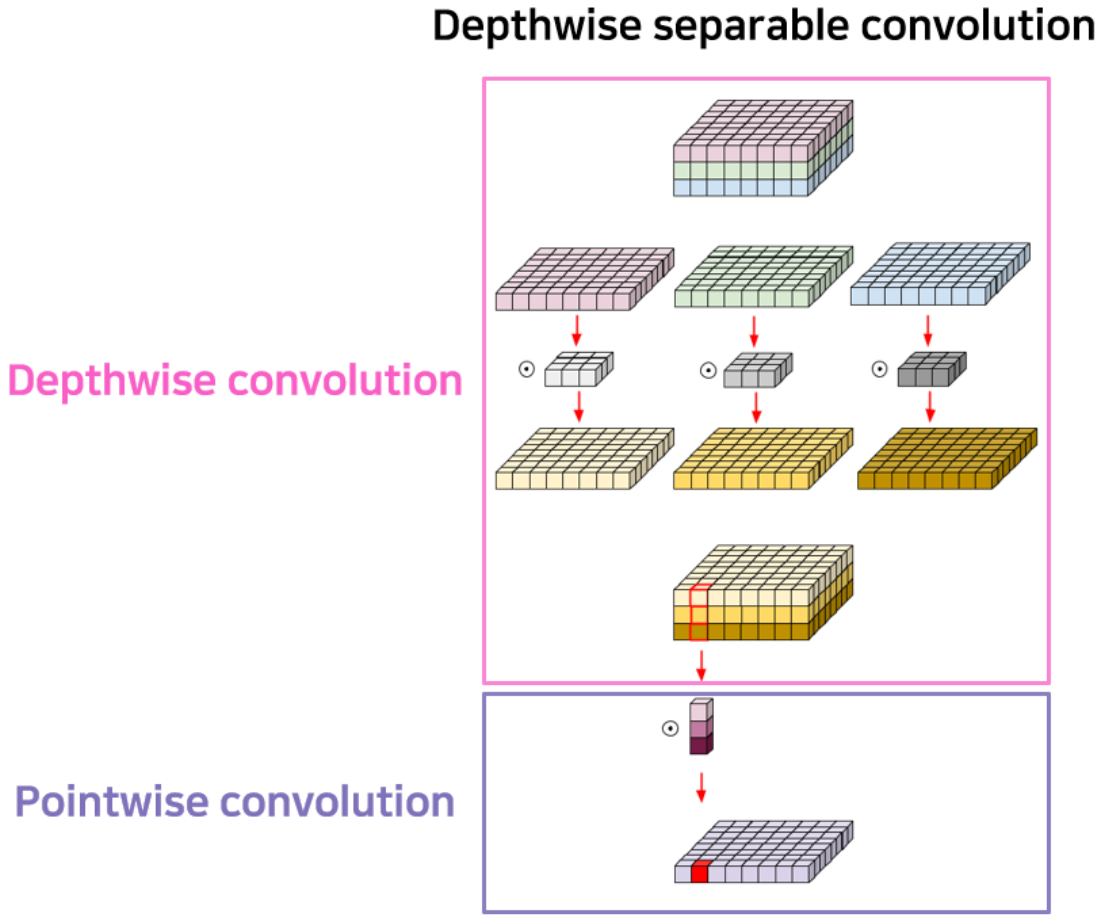
- Sufficiently High Accuracy
- Low Computational Complexity
- Low Energy Consumption
- Small Size of Model
- ~~Cool Name (e.g. YOLO V3)~~

SMARCLE

Recall

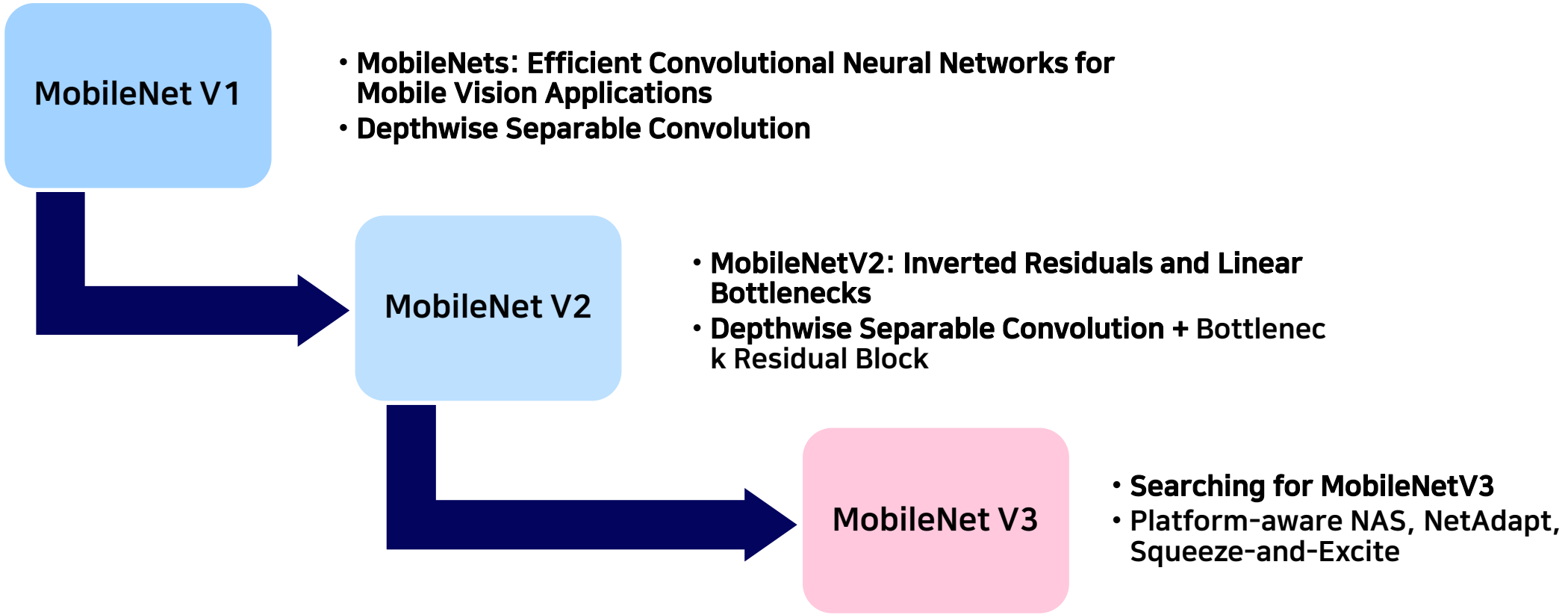
Abstract

We present a class of efficient models called MobileNets for mobile and embedded vision applications. MobileNets are based on a streamlined architecture that uses depth-wise separable convolutions to build light weight deep neural networks. We introduce two simple global hyper-parameters that efficiently trade off between latency and accuracy. These hyper-parameters allow the model builder to choose the right sized model for their application based on the constraints of the problem. We present extensive experiments on resource and accuracy tradeoffs and show strong performance compared to other popular models on ImageNet classification. We then demonstrate the effective-ness of MobileNets across a wide range of applications and use cases including object detection, finegrain classifica-tion, face attributes and large scale geo-localization.



Key Idea: Depthwise Separable Convolution

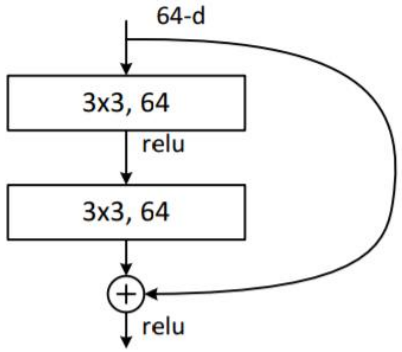
Before Launching in to the MobileNet V3



Before Launching in to the MobileNet V3

Quick review : MobileNet V2

Bottleneck Structure



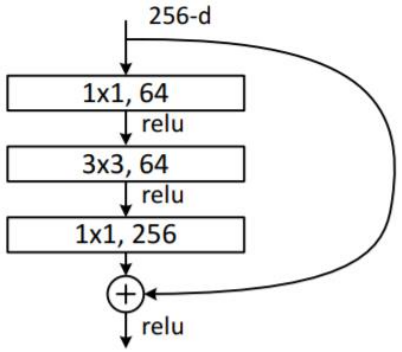
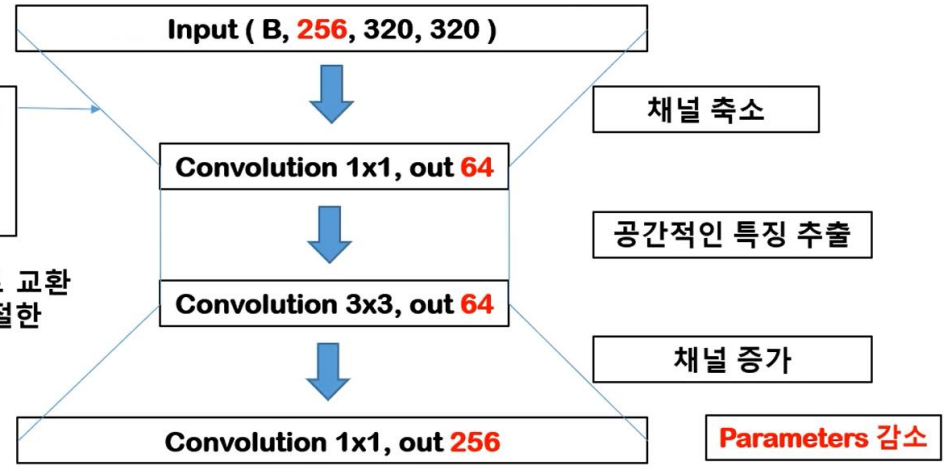


Figure 5. A deeper residual function $\mathcal{F}$ for ImageNet. Left: a building block (on 56×56 feature maps) as in Fig. 3 for ResNet-34. Right: a "bottleneck" building block for ResNet-50/101/152.

차원을 무작위로 줄이면 속도는 증가하지만, 정보 손실 발생

>> 속도와 정확도는 서로 교환 관계 이기 때문에 적절한 합의점이 필요함.



SMARCLE

Before Launching in to the MobileNet V3

Quick review：MobileNet V2

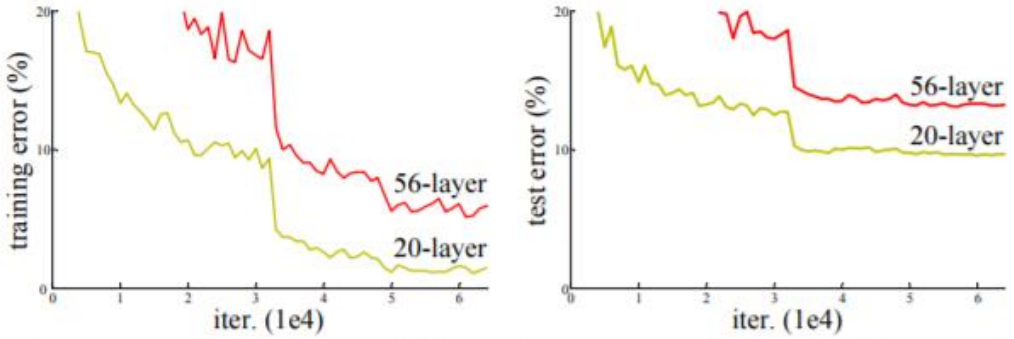


Figure 1. Training error (left) and test error (right) on CIFAR-10 with 20-layer and 56-layer "plain" networks. The deeper network has higher training error, and thus test error. Similar phenomena on ImageNet is presented in Fig. 4.

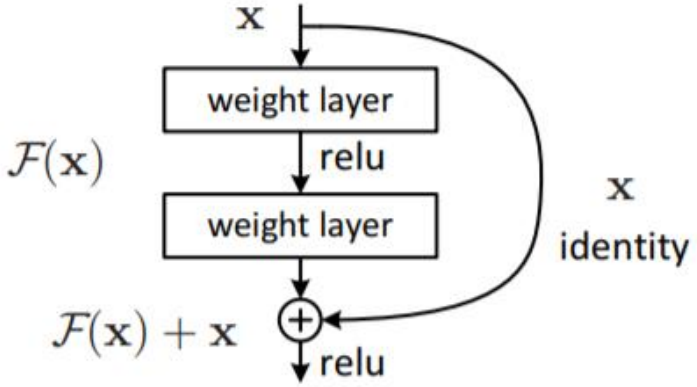
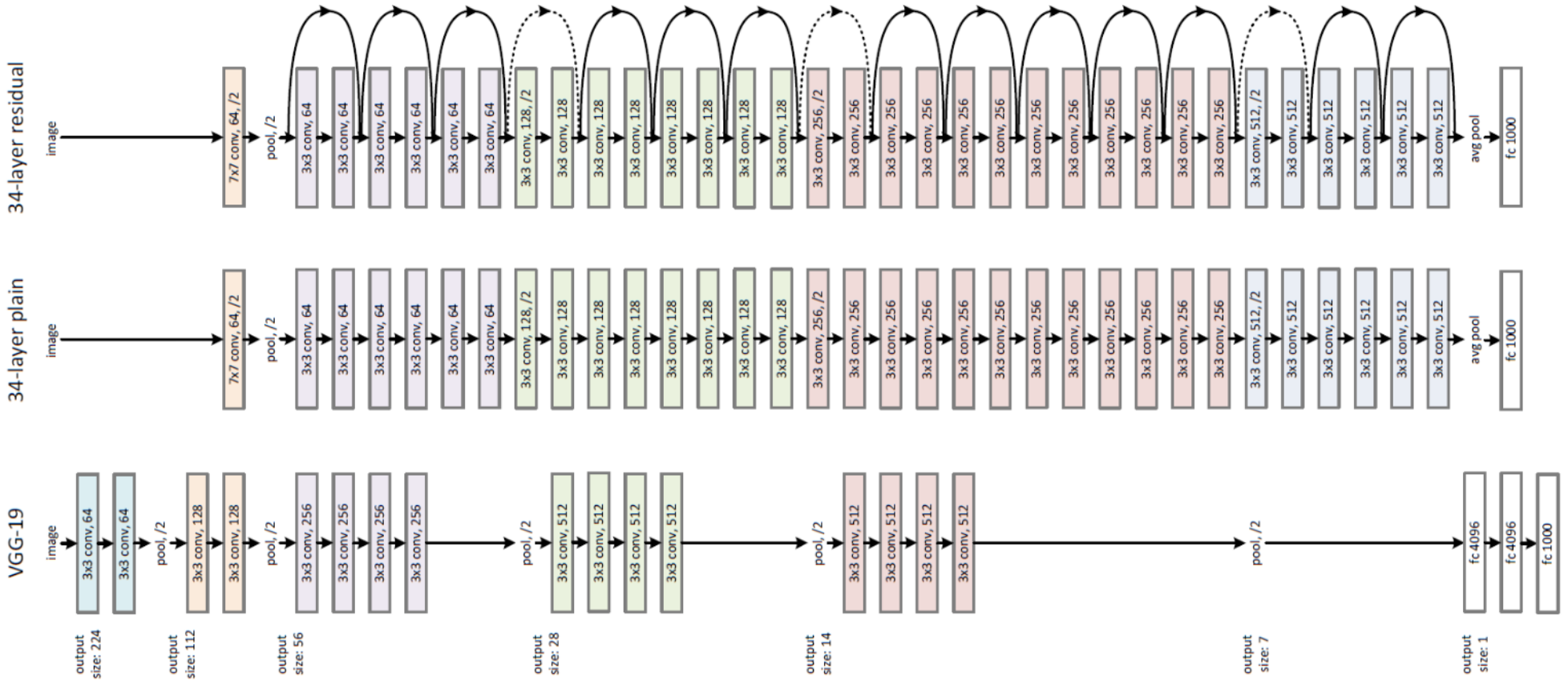


Figure 2. Residual learning: a building block.

Before Launching in to the MobileNet V3

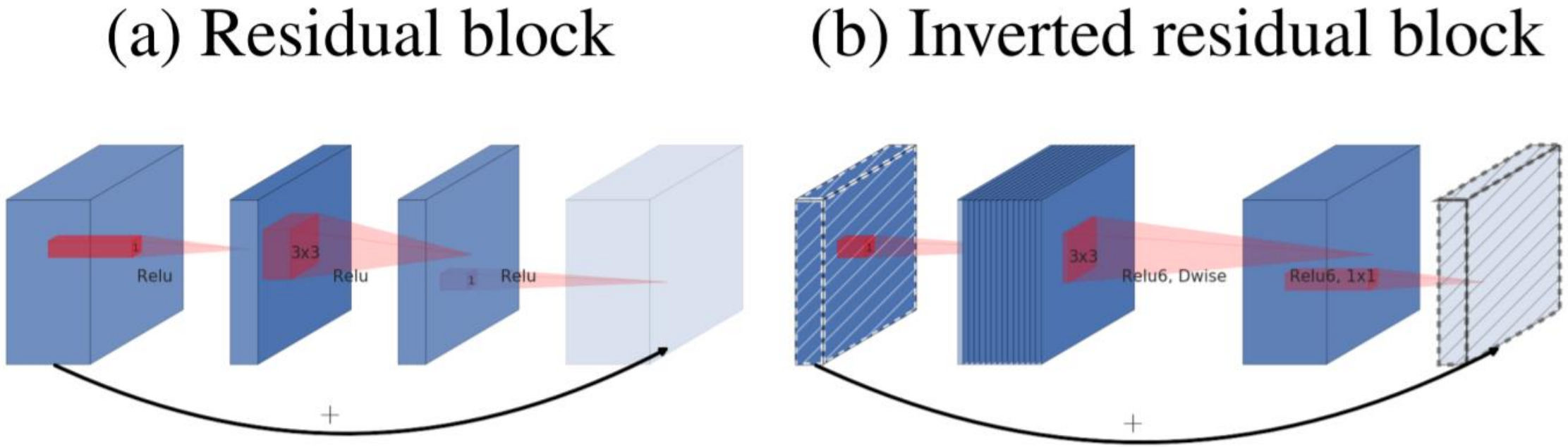
Quick review：MobileNet V2



SMARCLE

Before Launching in to the MobileNet V3

Quick review：MobileNet V2

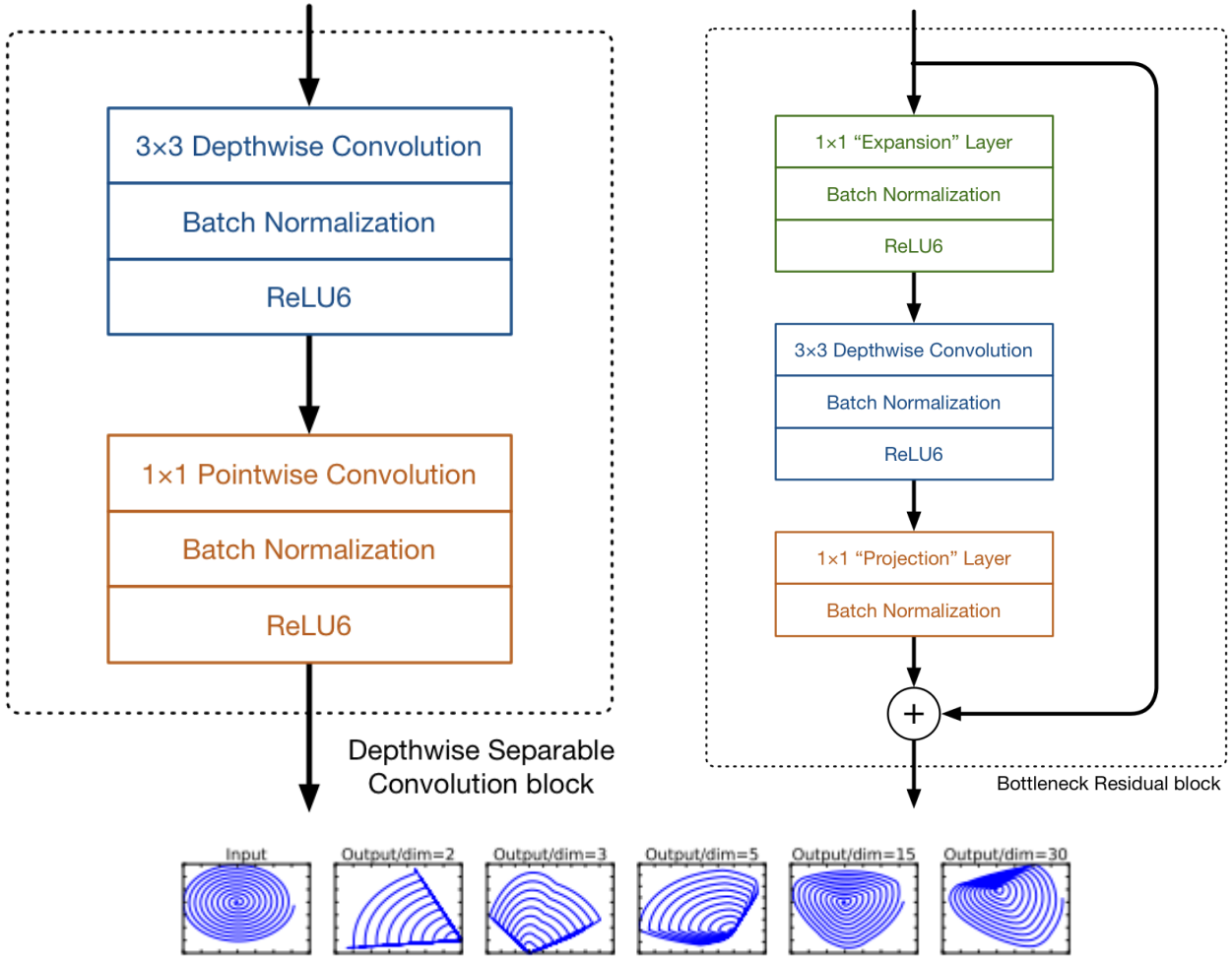


Thick-Thin-Thick

Thin-Thick-Thin

Before Launching in to the MobileNet V3

Quick review：MobileNet V2



SMARCLE

Searching for MobileNetV3

Andrew Howard[1] **Mark Sandler**[1] **Grace Chu**[1] **Liang-Chieh Chen**[1] **Bo Chen**[1] **Mingxing Tan**[2]
Weijun Wang[1] **Yukun Zhu**[1] **Ruoming Pang**[2] **Vijay Vasudevan**[2] **Quoc V. Le**[2] **Hartwig Adam**[1]
[1]Google AI, [2]Google Brain
{howarda, sandler, cxy, lcchen, bochen, tanmingxing, weijunw, yukun, rpang, vrv, qvl, hadam}@google.com

ICCV 2019
Seoul, Korea

SMARCLE

MobileNet V3

Abstract

We present the next generation of MobileNets based on a combination of complementary search techniques as well as a novel architecture design. MobileNetV3 is tuned to mobile phone CPUs through a combination of hardware-aware network architecture search (NAS) complemented by the NetAdapt algorithm and then subsequently improved through novel architecture advances. This paper starts the exploration of how automated search algorithms and network design can work together to harness complementary approaches improving the overall state of the art. Through this process we create two new MobileNet models for release: MobileNetV3-Large and MobileNetV3-Small which are targeted for high and low resource use cases. These models are then adapted and applied to the tasks of object detection and semantic segmentation. For the task of semantic segmentation (or any dense pixel prediction), we propose a new efficient segmentation decoder Lite Reduced Atrous Spatial Pyramid Pooling (LR-ASPP). We achieve new state of the art results for mobile classification, detection and segmentation. MobileNetV3-Large is 3.2% more accurate on ImageNet classification while reducing latency by 20% compared to MobileNetV2. MobileNetV3-Small is 6.6% more accurate compared to a MobileNetV2 model with comparable latency. MobileNetV3-Large detection is over 25% faster at roughly the same accuracy as MobileNetV2 on COCO detection. MobileNetV3-Large LR-ASPP is 34% faster than MobileNetV2 R-ASPP at similar accuracy for Cityscapes segmentation.

Key Idea

Platform-aware NAS(Network Architecture Search),
NetAdapt,
Squeeze-and-Excite,
h-swish Activation Function

SMARCLE

Squeeze-and-Excite

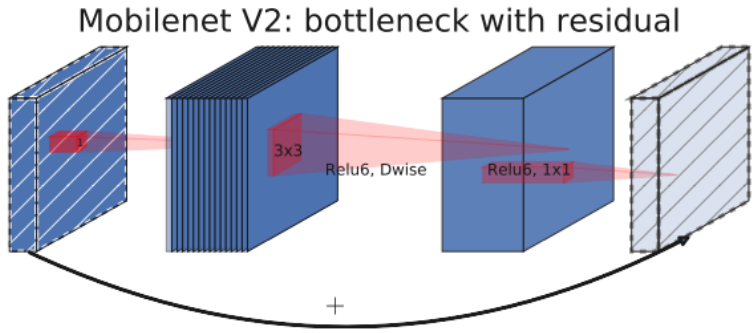


Figure 3. MobileNetV2 [39] layer (Inverted Residual and Linear Bottleneck). Each block consists of narrow input and output (bottleneck), which don't have nonlinearity, followed by expansion to a much higher-dimensional space and projection to the output. The residual connects bottleneck (rather than expansion).

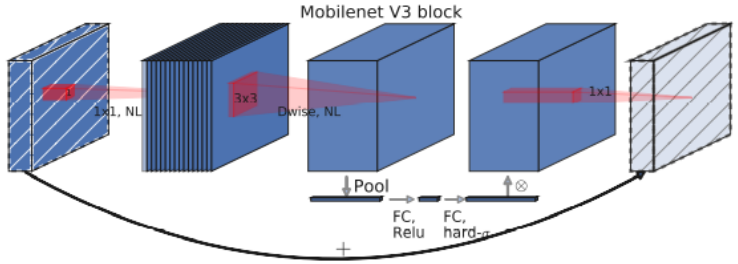
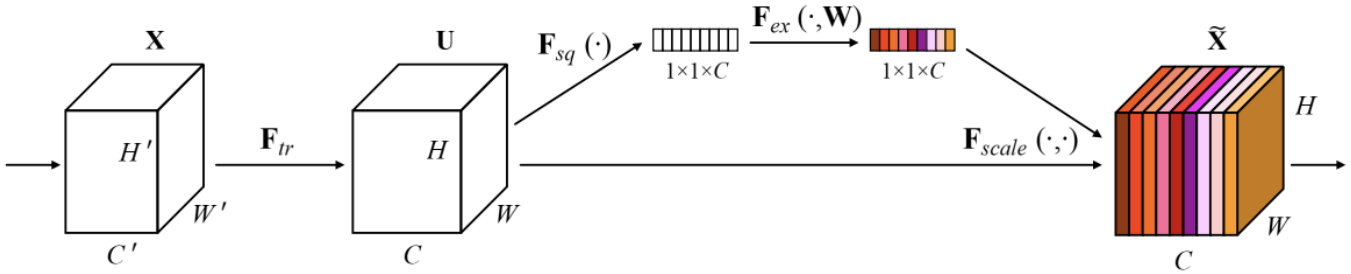


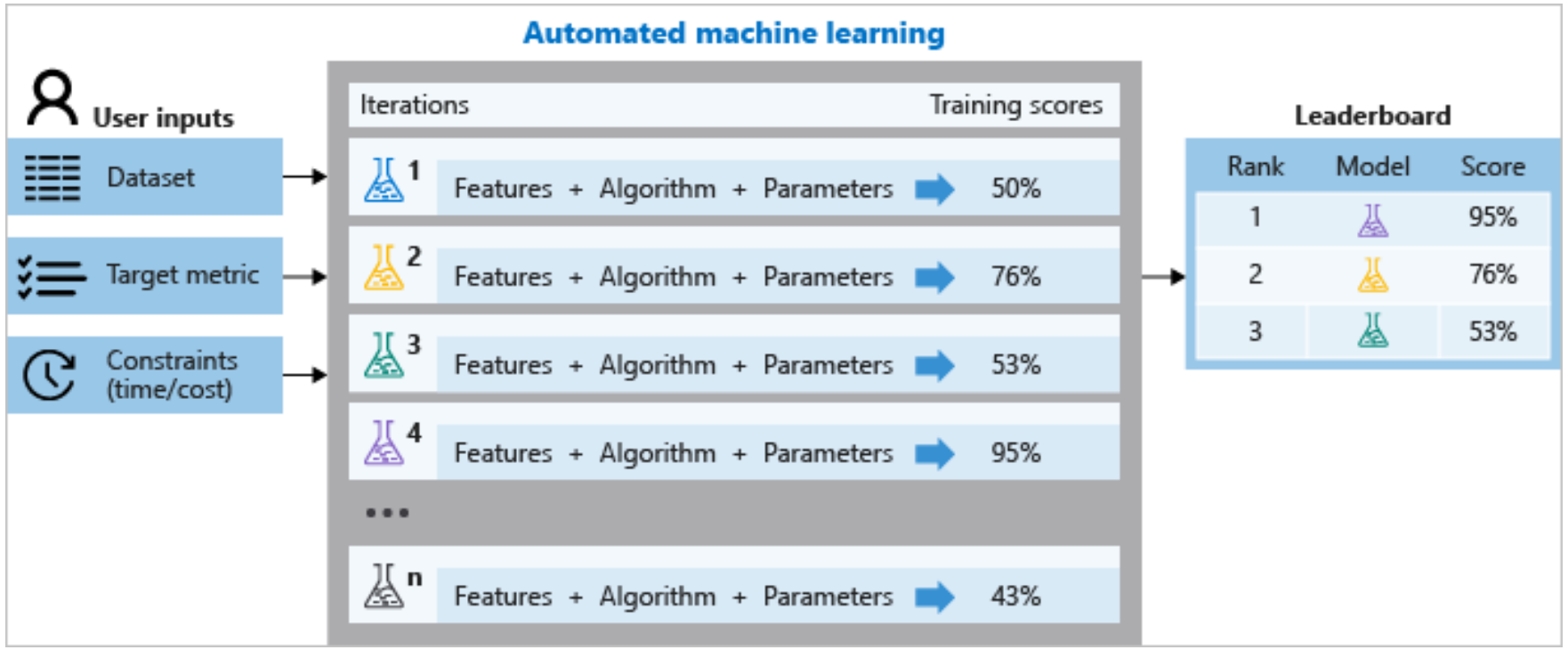
Figure 4. MobileNetV2 + Squeeze-and-Excite [20]. In contrast with [20] we apply the squeeze and excite in the residual layer. We use different nonlinearity depending on the layer, see section 5.2 for details.



SMARCLE

AutoML; NAS(Network Architecture Search)

MnasNet: Platform-Aware Neural Architecture Search for Mobile
Platform-Aware NAS for Block-wise Search



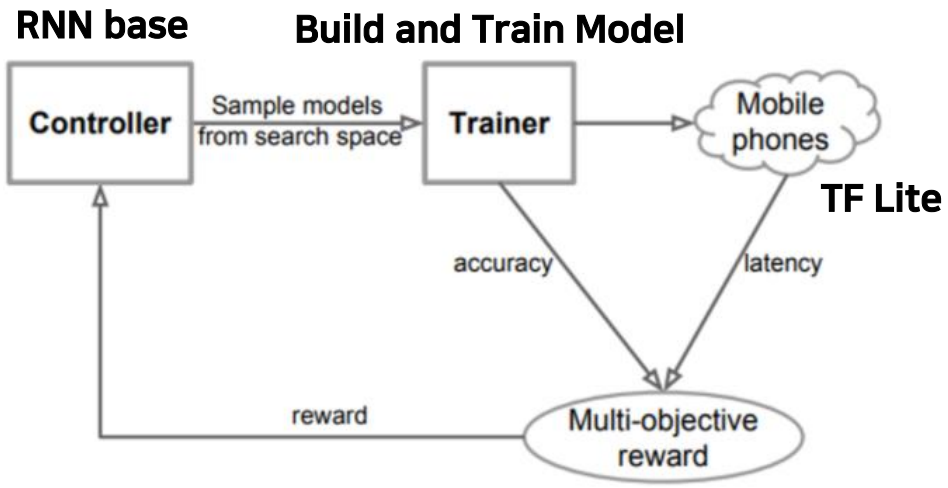
ICLR 2017, NEURAL ARCHITECTURE SEARCH WITH REINFORCEMENT LEARNING

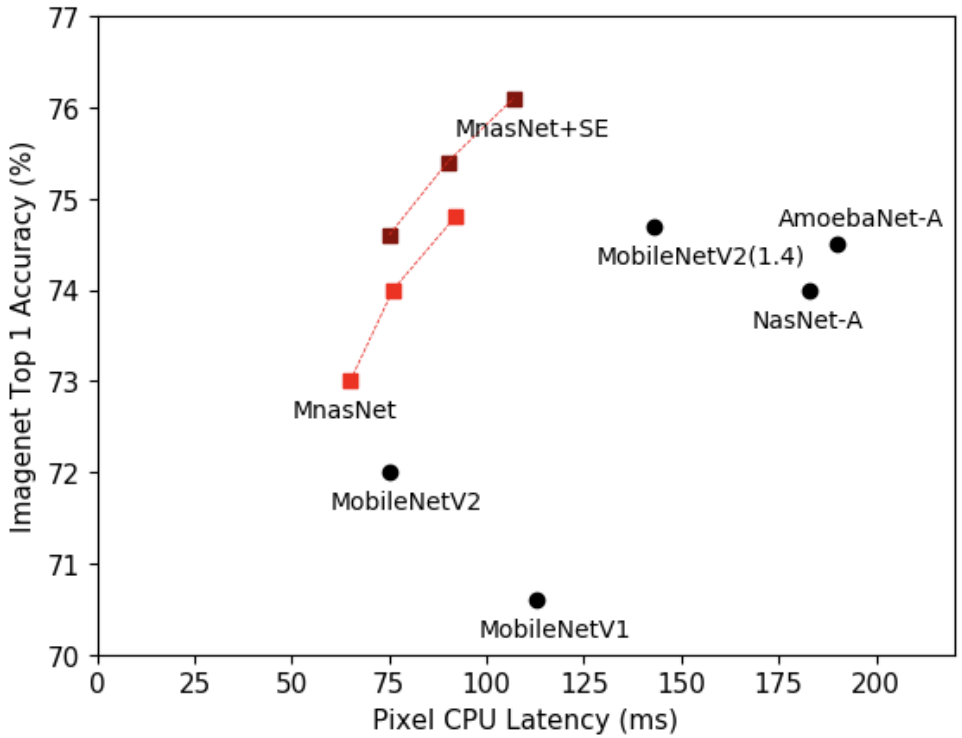
SMARCLE

AutoML; NAS(Network Architecture Search)

MnasNet: Platform-Aware Neural Architecture Search for Mobile

Platform-Aware NAS for Block-wise Search





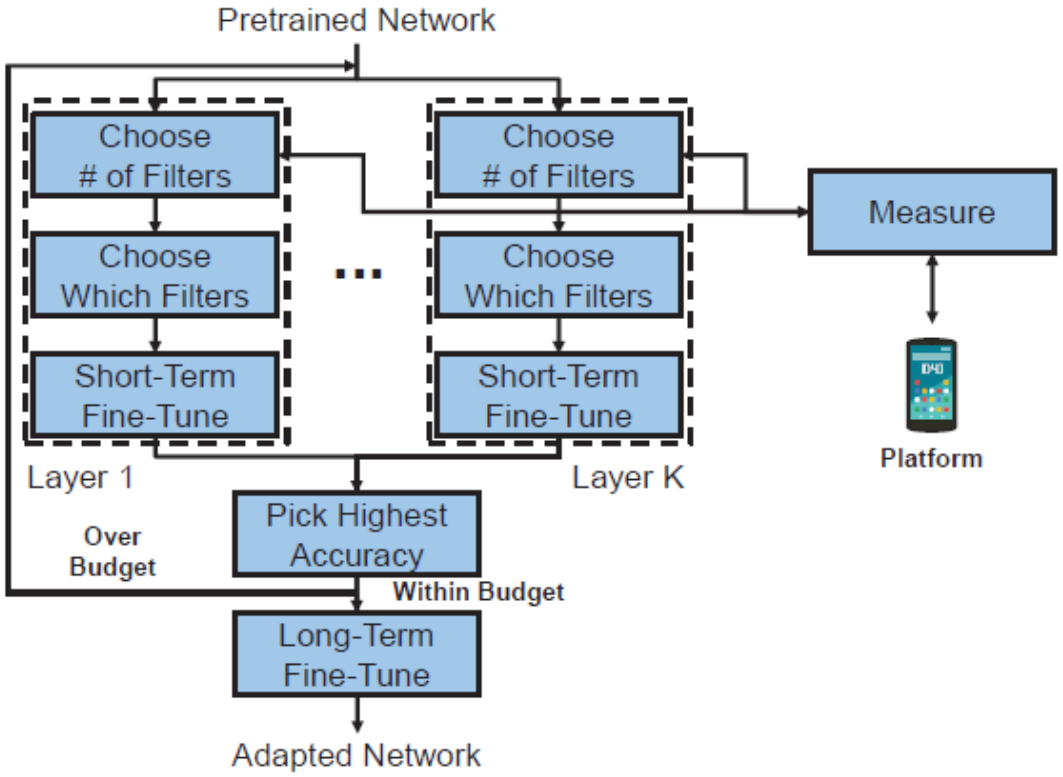
Based on MobileNet V2

Block 단위 최적화

SMARCLE

NetAdapt: Platform-Aware Neural Network Adaptation for Mobile Applications

NetAdapt for Layer-wise Search



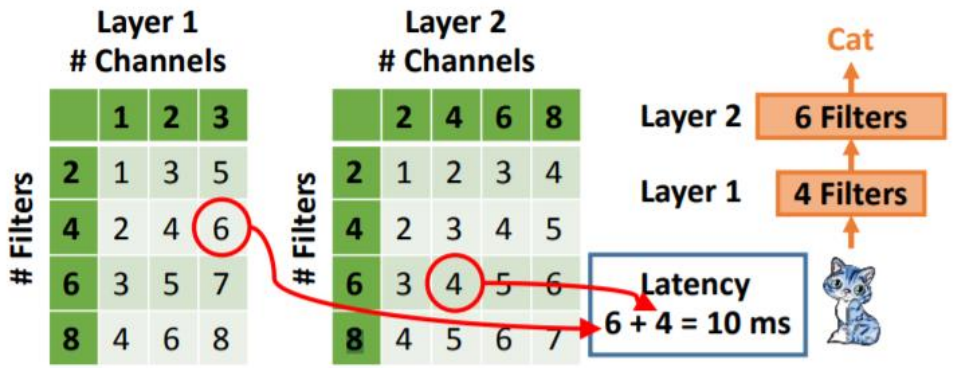


Fig. 3. This figure illustrates how layer-wise look-up tables are used for fast resource consumption estimation.

Layer 단위 최적화

SMARCLE

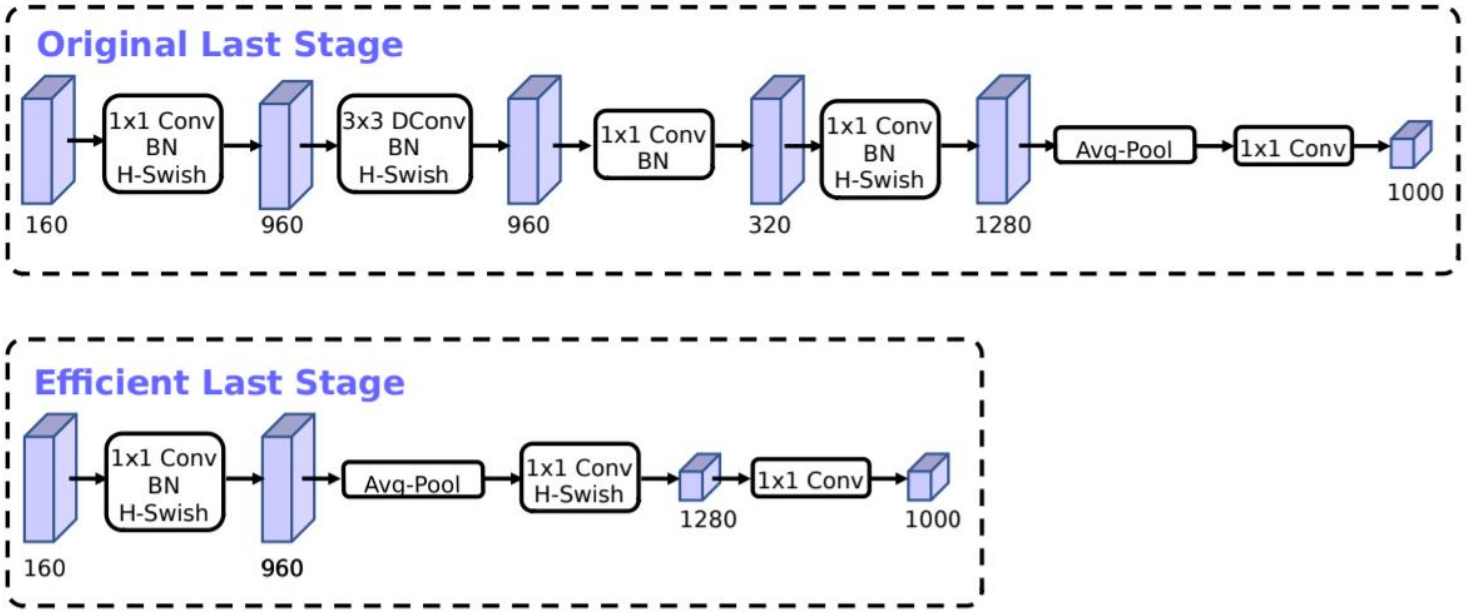


Figure 5. Comparison of original last stage and efficient last stage. This more efficient last stage is able to drop three expensive layers at the end of the network at no loss of accuracy.

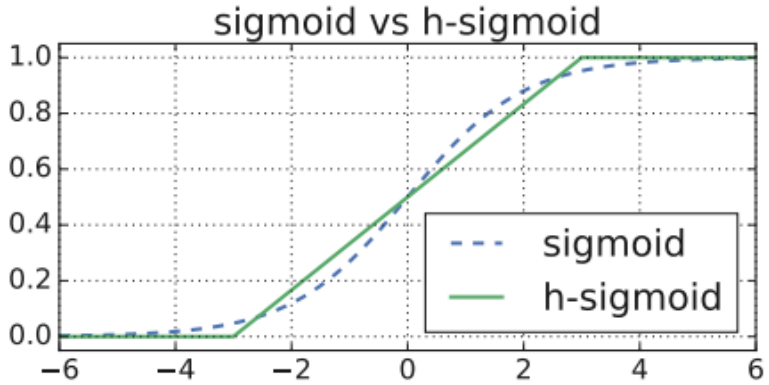
In [36, 13, 16] a nonlinearity called *swish* was introduced that when used as a drop-in replacement for ReLU, that significantly improves the accuracy of neural networks. The nonlinearity is defined as

$$\text{swish}\, x = x \cdot \sigma(x)$$

While this nonlinearity improves accuracy, it comes with non-zero cost in embedded environments as the sigmoid function is much more expensive to compute on mobile devices. We deal with this problem in two ways.

1. We replace sigmoid function with its piece-wise linear hard analog: $\frac{\text{ReLU6}(x+3)}{6}$ similar to [11, 44]. The minor difference is we use ReLU6 rather than a custom clipping constant. Similarly, the hard version of swish becomes

$$\text{h-swish}[x] = x\frac{\text{ReLU6}(x+3)}{6}$$



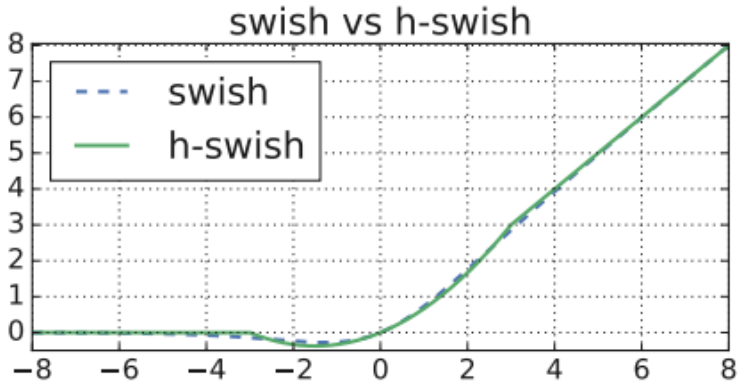


Figure 6. Sigmoid and swish nonlinearities and ther "hard" coun terparts.

5.4. MobileNetV3 Definitions

MobileNetV3 is defined as two models: MobileNetV3-Large and MobileNetV3-Small. These models are targeted at high and low resource use cases respectively. The models are created through applying platform-aware NAS and NetAdapt for network search and incorporating the network improvements defined in this section. See table 1 and 2 for full specification of our networks.

MobileNet V3

Input	Operator	exp size	#out	SE	NL	s
$224^2 \times 3$	conv2d	-	16	-	HS	2
$112^2 \times 16$	bneck, 3x3	16	16	-	RE	1
$112^2 \times 16$	bneck, 3x3	64	24	-	RE	2
$56^2 \times 24$	bneck, 3x3	72	24	-	RE	1
$56^2 \times 24$	bneck, 5x5	72	40	✓	RE	2
$28^2 \times 40$	bneck, 5x5	120	40	✓	RE	1
$28^2 \times 40$	bneck, 5x5	120	40	✓	RE	1
$28^2 \times 40$	bneck, 3x3	240	80	-	HS	2
$14^2 \times 80$	bneck, 3x3	200	80	-	HS	1
$14^2 \times 80$	bneck, 3x3	184	80	-	HS	1
$14^2 \times 80$	bneck, 3x3	184	80	-	HS	1
$14^2 \times 80$	bneck, 3x3	480	112	✓	HS	1
$14^2 \times 112$	bneck, 3x3	672	112	✓	HS	1
$14^2 \times 112$	bneck, 5x5	672	160	✓	HS	2
$7^2 \times 160$	bneck, 5x5	960	160	✓	HS	1
$7^2 \times 160$	bneck, 5x5	960	160	✓	HS	1
$7^2 \times 160$	conv2d, 1x1	-	960	-	HS	1
$7^2 \times 960$	pool, 7x7	-	-	-	-	1
$1^2 \times 960$	conv2d 1x1, NBN	-	1280	-	HS	1
$1^2 \times 1280$	conv2d 1x1, NBN	-	k	-	-	1

Table 1. Specification for MobileNetV3-Large. SE denotes whether there is a Squeeze-And-Excite in that block. NL denotes the type of nonlinearity used. Here, HS denotes h-swish and RE denotes ReLU. NBN denotes no batch normalization. s denotes stride.

Input	Operator	exp size	#out	SE	NL	s
$224^2 \times 3$	conv2d, 3x3	-	16	-	HS	2
$112^2 \times 16$	bneck, 3x3	16	16	✓	RE	2
$56^2 \times 16$	bneck, 3x3	72	24	-	RE	2
$28^2 \times 24$	bneck, 3x3	88	24	-	RE	1
$28^2 \times 24$	bneck, 5x5	96	40	✓	HS	2
$14^2 \times 40$	bneck, 5x5	240	40	✓	HS	1
$14^2 \times 40$	bneck, 5x5	240	40	✓	HS	1
$14^2 \times 40$	bneck, 5x5	120	48	✓	HS	1
$14^2 \times 48$	bneck, 5x5	144	48	✓	HS	1
$14^2 \times 48$	bneck, 5x5	288	96	✓	HS	2
$7^2 \times 96$	bneck, 5x5	576	96	✓	HS	1
$7^2 \times 96$	bneck, 5x5	576	96	✓	HS	1
$7^2 \times 96$	conv2d, 1x1	-	576	✓	HS	1
$7^2 \times 576$	pool, 7x7	-	-	-	-	1
$1^2 \times 576$	conv2d 1x1, NBN	-	1024	-	HS	1
$1^2 \times 1024$	conv2d 1x1, NBN	-	k	-	-	1

Table 2. Specification for MobileNetV3-Small. See table 1 for notation.

SMARCLE

ms

Network	Top-1	MAdds	Params	P-1	P-2	P-3
V3-Large 1.0	**75.2**	**219**	5.4M	**51**	**61**	**44**
V3-Large 0.75	73.3	155	4.0M	39	46	40
MnasNet-A1	75.2	315	3.9M	71	86	61
Proxyless[5]	74.6	320	4.0M	72	84	60
V2 1.0	72.0	300	3.4M	64	76	56
V3-Small 1.0	**67.4**	**56**	2.5M	**15.8**	**19.4**	**14.4**
V3-Small 0.75	65.4	44	2.0M	12.8	15.6	11.7
Mnas-small [43]	64.9	65.1	1.9M	20.3	24.2	17.2
V2 0.35	60.8	59.2	1.6M	16.6	19.6	13.9

Table 3. Floating point performance on the Pixel family of phones (P-n denotes a Pixel-n phone). All latencies are in ms and are measured using a single large core with a batch size of one. Top-1 accuracy is on ImageNet.

ms

Network	Top-1	P-1	P-2	P-3
V3-Large 1.0	**73.8**	44	42.5	31.7
V2 1.0	70.9	52	48.3	37.0
V3-Small	**64.9**	15.5	14.9	10.7
V2 0.35	57.2	16.7	15.6	11.9

Table 4. Quantized performance. All latencies are in ms. The inference latency is measured using a single large core on the respective Pixel 1/2/3 device.

MobileNet V3

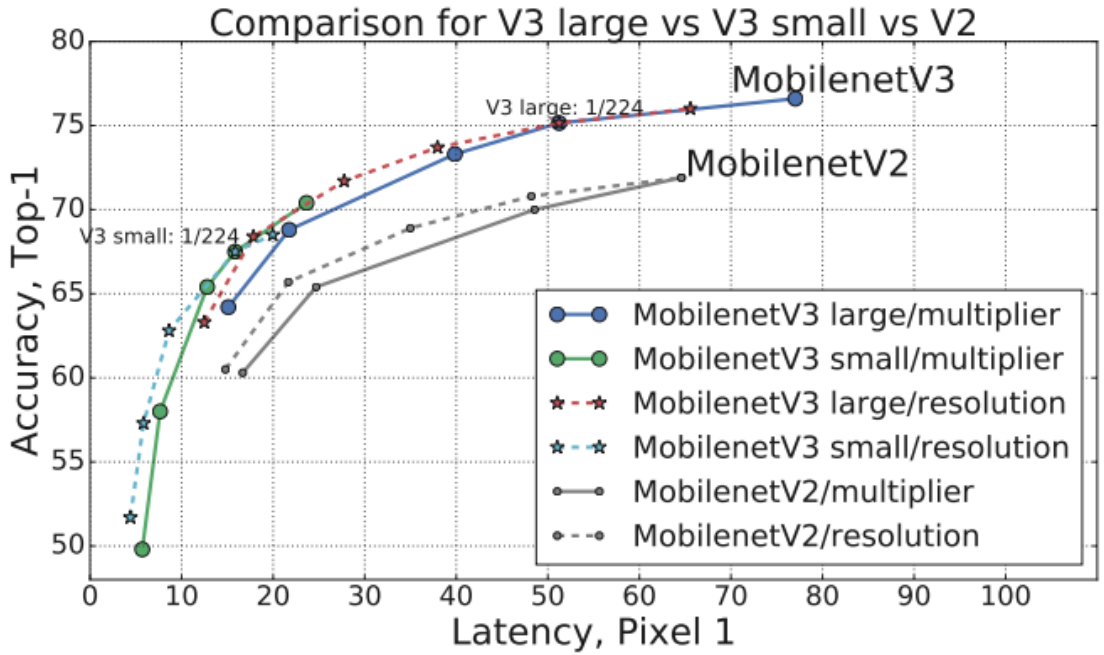


Figure 7. Performance of MobileNetV3 as a function of different multipliers and resolutions. In our experiments we have used multipliers 0.35, 0.5, 0.75, 1.0 and 1.25, with a fixed resolution of 224, and resolutions 96, 128, 160, 192, 224 and 256 with a fixed depth multiplier of 1.0. Best viewed in color. Top-1 accuracy is on ImageNet and latency is in ms.

	Top-1	P-1	P-1 (no-opt)
V3-Large 1.0	75.2	51.4	57.5
ReLU	74.5 (-.7%)	50.5 (-1%)	50.5
h-swish @16	75.4 (+.2%)	53.5 (+4%)	68.9
h-swish @112	75.0 (-.3%)	51 (-0.5%)	54.4

Table 5. Effect of non-linearities on MobileNetV3-Large. In h-swish @N, N denotes the number of channels, in the first layer that has h-swish enabled. The third column shows the runtime without optimized h-swish. Top-1 accuracy is on ImageNet and latency is in ms.

SMARCLE

MobileNet V3

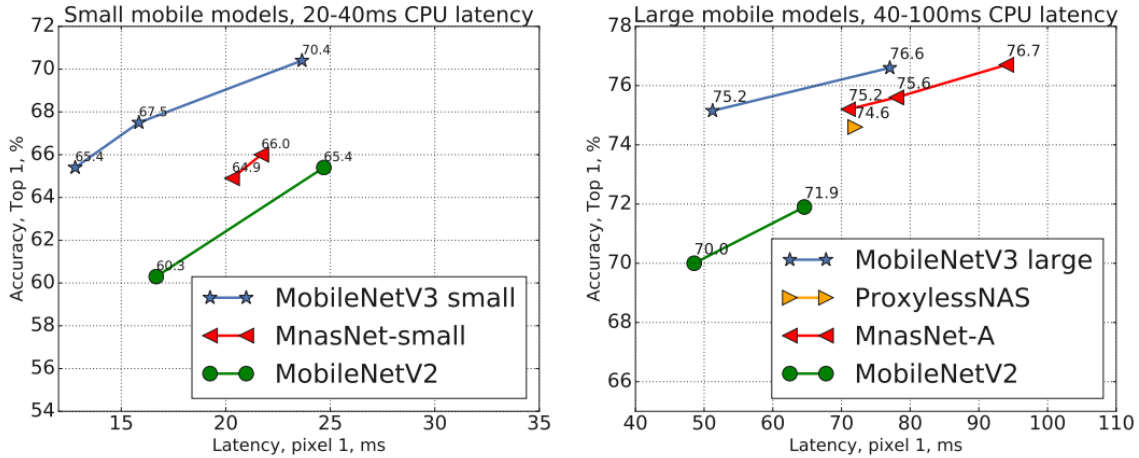


Figure 1. The trade-off between Pixel 1 latency and top-1 ImageNet accuracy. All models use the input resolution 224. V3 large and V3 small use multipliers 0.75, 1 and 1.25 to show optimal frontier. All latencies were measured on a single large core of the same device using TFLite[1]. MobileNetV3-Small and Large are our proposed next-generation mobile models.

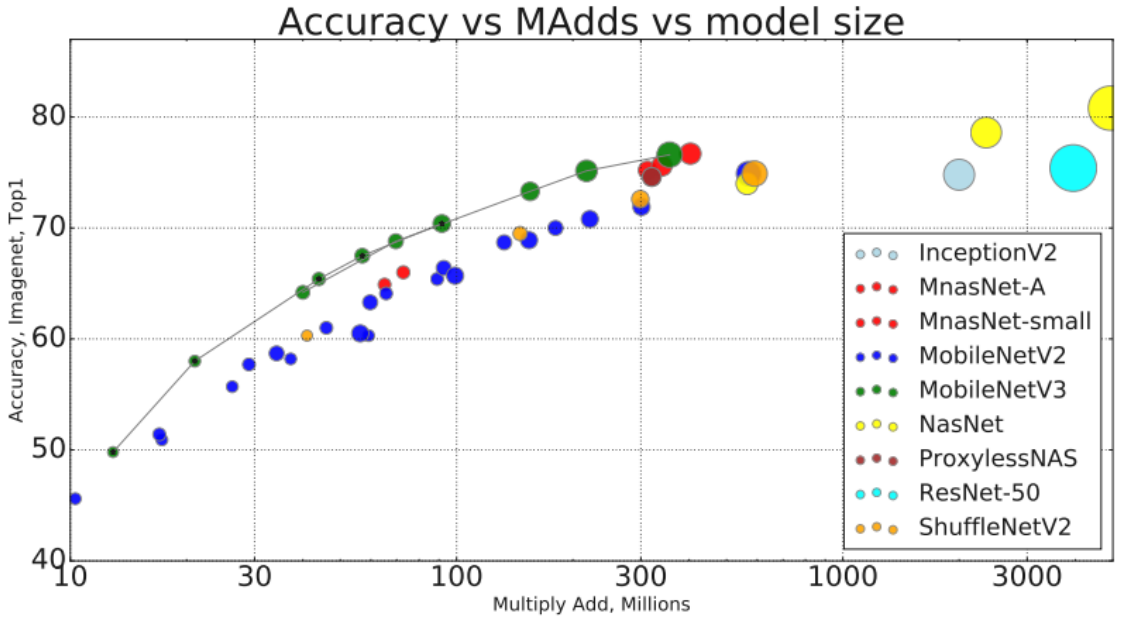


Figure 2. The trade-off between MAdds and top-1 accuracy. This allows to compare models that were targeted different hardware or software frameworks. All MobileNetV3 are for input resolution 224 and use multipliers 0.35, 0.5, 0.75, 1 and 1.25. See section 6 for other resolutions. Best viewed in color.

SMARCLE

Detection, Semantic Segmentation, so on..

Backbone	mAP	Latency (ms)	Params (M)	MAdds (B)
V1	22.2	228	5.1	1.3
V2	22.1	162	4.3	0.80
MnasNet	23.0	174	4.88	0.84
V3	22.0	137	4.97	0.62
V3†	22.0	119	3.22	0.51
V2 0.35	13.7	66	0.93	0.16
V2 0.5	16.6	79	1.54	0.27
MnasNet 0.35	15.6	68	1.02	0.18
MnasNet 0.5	18.5	85	1.68	0.29
V3-Small	16.0	52	2.49	0.21
V3-Small†	16.1	43	1.77	0.16

Table 6. Object detection results of SSDLite with different backbones on COCO test set. †: Channels in the blocks between $C4$ and $C5$ are reduced by a factor of 2.

N	Backbone	RF2	SH	F	mIOU	Params	MAdds	CPU (f)	CPU (h)
1	V2	-	×	256	72.84	2.11M	21.29B	3.90s	1.02s
2	V2	✓	×	256	72.56	1.15M	13.68B	3.03s	793ms
3	V2	✓	✓	256	72.97	1.02M	12.83B	2.98s	786ms
4	V2	✓	✓	128	72.74	0.98M	12.57B	2.89s	766ms
5	V3	-	×	256	72.64	3.60M	18.43B	3.55s	906ms
6	V3	✓	×	256	71.91	1.76M	11.24B	2.60s	668ms
7	V3	✓	✓	256	72.37	1.63M	10.33B	2.55s	659ms
8	V3	✓	✓	128	72.36	1.51M	9.74B	2.47s	657ms
9	V2 0.5	✓	✓	128	68.57	0.28M	4.00B	1.59s	415ms
10	V2 0.35	✓	✓	128	66.83	0.16M	2.54B	1.27s	354ms
11	V3-Small	✓	✓	128	68.38	0.47M	2.90B	1.21s	327ms

Table 7. Semantic segmentation results on Cityscapes *val* set. **RF2**: **R**educe the **F**ilters in the last block by a factor of **2**. V2 0.5 and V2 0.35 are MobileNetV2 with depth multiplier = 0.5 and 0.35, respectively. **SH: S**egmentation **H**ead, where × employs the R-ASPP while ✓ employs the proposed LR-ASPP. **F:** Number of **F**ilters used in the Segmentation Head. **CPU (f):** CPU time measured on a single large core of Pixel 3 (floating point) w.r.t. a **full-resolution** input (i.e., 1024 × 2048). **CPU (h):** CPU time measured w.r.t. a **half-resolution** input (i.e., 512 × 1024). Row 8, and 11 are our MobileNetV3 segmentation candidates.

SMARCLE