MobileNet, MobileNetV2

Author

Andrew Howard

Google Research
google.com의 이메일 확인됨
Computer Vision Machine L...

Mark Sandler

Google
google.com의 이메일 확인됨 - 홈페이지

제목	인용	연도
Mobilenets: Efficient convolutional neural networks for mobile vision applications AG Howard, M Zhu, B Chen, D Kalenichenko, W Wang, T Weyand, ... arXiv preprint arXiv:1704.04861	12866	2017
Mobilenetv2: Inverted residuals and linear bottlenecks M Sandler, A Howard, M Zhu, A Zhmoginov, LC Chen Proceedings of the IEEE Conference on Computer Vision and Pattern ...	39443	2018
Searching for MobileNetV3 A Howard, M Sandler, G Chu, LC Chen, B Chen, M Tan, W Wang, Y Zhu, ... arXiv preprint arXiv:1905.02244	11632	2019

모바일, 임베디드 같은 제한된 환경에서 효율적으로 동작하는 모델

MobileNet V1 - idea

Depthwise Separable Convolutions

- Depthwise Convolution + Pointwise Convolution
- 계산량과 모델 크기 감소

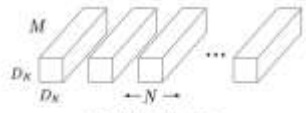
Width Multiplier

- 입력 및 출력 채널 수를 조정

Resolution Multiplier

- 이미지 해상도 감소

MobileNet V1 – Depthwise Separable Convolutions



Standard Convolution

$$D_k * D_k * M * N * D_F * D_F$$

Depthwise Separable Convolution

$$D_k * D_k * M * D_F * D_F + M * N * D_F + D_F$$

표준 합성곱 대비 계산량 $\frac{1}{N} + \frac{1}{D_k^2}$ 감소

MobileNet V1 – Depthwise Separable Convolutions

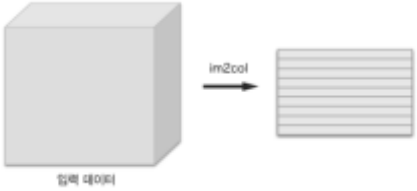
Table 1. MobileNet Body Architecture

Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
$5\times$ Conv dw / s1	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
$5\times$ Conv / s1	$1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$
Conv dw / s2	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024$ dw	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool 7×7	$7 \times 7 \times 1024$
FC / s1	1024×1000	$1 \times 1 \times 1024$
Softmax / s1	Classifier	$1 \times 1 \times 1000$

MobileNet V1 – Depthwise Separable Convolutions(GEMM 보충)

Table 2. Resource Per Layer Type

Type	Mult-Adds	Parameters
Conv 1 × 1	94.86%	74.59%
Conv DW 3 × 3	3.06%	1.06%
Conv 3 × 3	1.19%	0.02%
Fully Connected	0.18%	24.33%



모든 연산을 1*1 Conv에 집중

최적화된 General Matrix Multiply(GEMM) 활용 가능

- 메모리 내에서의 초기 재정렬이 필요 없어 효율적이다.

MobileNet V1 – Parameter

Width Multiplier

채널 수를 조절하는 파라미터

$D_k * D_k * \alpha M * D_F * D_F + \alpha M * \alpha N * D_F + D_F$

계산 비용과 파라미터가 α^2에 비례해 감소 ($\alpha < 1$)

Resolution Multiplier

공간적인 데이터 크기를 조절하는 파라미터

$D_k * D_k * M * \rho D_F * \rho D_F + M * N * \rho D_F + \rho D_F$

계산 비용과 파라미터가 ρ^2만큼 감소 ($\rho < 1$)

Result - MobileNet

Table 4. Depthwise Separable vs Full Convolution MobileNet

Model	ImageNet Accuracy	Million Mult-Adds	Million Parameters
Conv MobileNet	71.7%	4866	29.3
MobileNet	70.6%	569	4.2

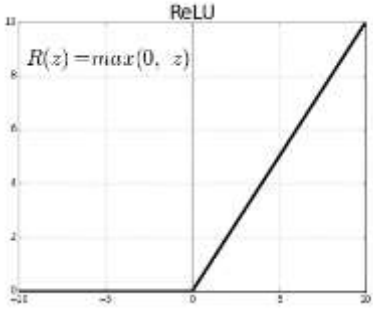
Table 6. MobileNet Width Multiplier

Width Multiplier	ImageNet Accuracy	Million Mult-Adds	Million Parameters
1.0 MobileNet-224	70.6%	569	4.2
0.75 MobileNet-224	68.4%	325	2.6
0.5 MobileNet-224	63.7%	149	1.3
0.25 MobileNet-224	50.6%	41	0.5

Table 7. MobileNet Resolution

Resolution	ImageNet Accuracy	Million Mult-Adds	Million Parameters
1.0 MobileNet-224	70.6%	569	4.2
1.0 MobileNet-192	69.1%	418	4.2
1.0 MobileNet-160	67.2%	290	4.2
1.0 MobileNet-128	64.4%	186	4.2

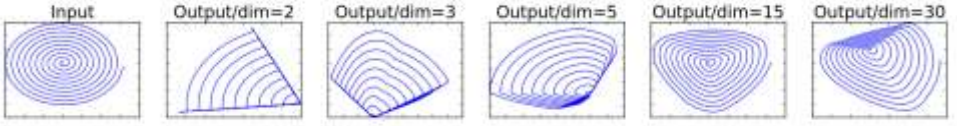
MobileNet V1 – 문제점



MobileNet V1은 Width Multiplier를 이용해 채널을 줄인다.

ReLU는 비선형 함수이기 때문에 저차원 공간에서 정보 손실 발생

MobileNet V1 – 문제점



저차원일 때 특징을 표현할 수 있는 값이 적기 때문에 ReLU에 의해 값이 0이 되면 원래 차원으로 복원됐을 때 정확한 표현이 불가능

고차원은 특정 값이 0이 되더라도 다른 차원의 값으로 충분히 원래의 상태를 표현 가능

MobileNet V2 – idea

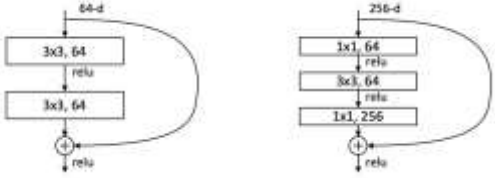
Linear Bottlenecks

- 출력 단계에서 비선형을 제거하여 정보를 손실 없이 유지

Inverted Residual

- 입출력은 좁은 차원, 중간에만 확장된 차원을 사용

MobileNet V2 – Bottlenecks

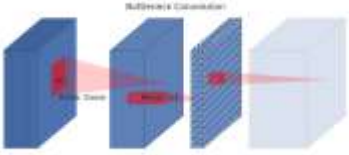


1*1 Conv는 차원을 감소시키고 복원하는 역할

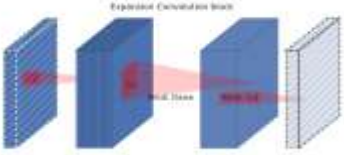
중간 층은 차원이 작아진 bottleneck을 형성

채널 압축과 복원을 통해 중요한 feature만 추출하고, 다시 큰 feature로 만들 수 있다

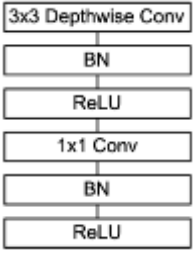
MobileNet V2 – Linear Bottlenecks



Separable with linear bottleneck



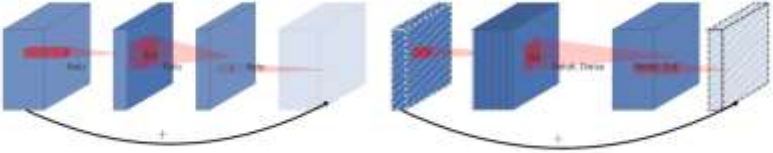
Bottleneck with expansion layer



기존 MoblieNet의 비선형 문제를 해결하기 위해 Linear Bottlenecks 사용

마지막 1*1 Conv의 ReLU를 제거

MobileNet V2 – Inverted Residuals



Residual block

Inverted residual block

기존 Residual Block은 입력과 출력이 모두 고차원이고 중간 연산을 줄이는 방식

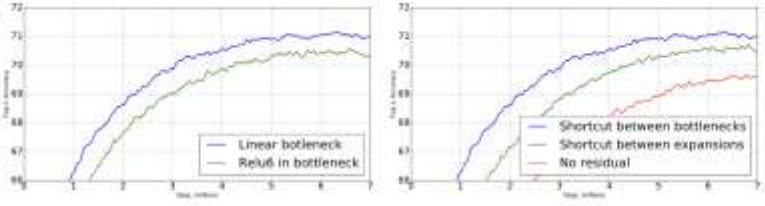
MobliNetV2는 중간 연산이 확장되는 방식

- 중간 연산 확장을 통해 표현력 강화
- ReLU로 인한 정보 손실을 최소화

Result – MobileNetV2

Input	Operator	t	c	n	s
$224^2 \times 3$	conv2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$14^2 \times 64$	bottleneck	6	96	3	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	conv2d 1x1	-	1280	1	1
$7^2 \times 1280$	avgpool 7x7	-	-	1	-
$1 \times 1 \times 1280$	conv2d 1x1	-	k	-	

Input	Operator	Output
$h \times w \times k$	1x1 conv2d , ReLU6	$h \times w \times (tk)$
$h \times w \times tk$	3x3 dwise s=s, ReLU6	$\frac{h}{s} \times \frac{w}{s} \times (tk)$
$\frac{h}{s} \times \frac{w}{s} \times tk$	linear 1x1 conv2d	$\frac{h}{s} \times \frac{w}{s} \times k'$



(a) Impact of non-linearity in the bottleneck layer.

(b) Impact of variations in residual blocks.

Result – MobileNetV2

Network	Top 1	Params	MAdds	CPU
MobileNetV1	70.6	4.2M	575M	113ms
ShuffleNet (1.5)	71.5	**3.4M**	292M	-
ShuffleNet (x2)	73.7	5.4M	524M	-
NasNet-A	74.0	5.3M	564M	183ms
MobileNetV2	**72.0**	**3.4M**	**300M**	**75ms**
MobileNetV2 (1.4)	**74.7**	6.9M	585M	**143ms**

ImageNet Classification

Result – MobileNetV2

Network	mAP	Params	MAdd	CPU
SSD300[34]	23.2	36.1M	35.2B	-
SSD512[34]	26.8	36.1M	99.5B	-
YOLOv2[35]	21.6	50.7M	17.5B	-
MNet V1 + SSDLite	22.2	5.1M	1.3B	270ms
MNet V2 + SSDLite	22.1	**4.3M**	**0.8B**	200ms

Object Dectection