MobileNets

COMPUTER VISION - CLASSIFICATION PART 2

16기 분석_조하늘

0. Abstract

- MobileNet은 모바일이나 embedded vision applications등 다양한 부분에서 적용되는 네트워크.
특히 연산량과 파라미터 수를 줄이면서 모델의 효율과 성능을 높임.
즉, Depthwise separable convolutions로 이뤄져 경량딥러닝을 가능케 해 -> small deep neural network 기여

Q) 왜 작고 효율적인 신경망(small deep neural network)은 중요한가?

- 추가적으로 두개의 하이퍼 파라미터인 width multiplier과 resolution multiplier
- 마지막으로, MobileNet 이 활용되는 다양한 분야들을 소개하고, 다른 모델들과도 비교해 그 효과 알아보기

1. MobileNet 배경

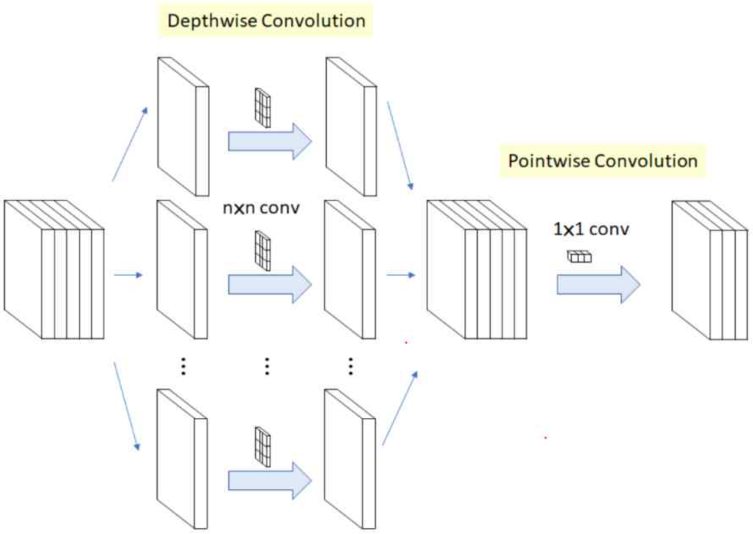
-기존 CNN, 연산량, 속도 문제 지님 :
 convolution 연산시 width, height, channel을 동시에 고려해 연산하고,
accuracy를 높이느라 연산량이 크고 속도가 느려지는 단점

- 이 문제 해결을 위해 이 논문에선 MobileNet을 제안.

* Depthwise Separable Convolution!*

2. MobileNet Architecture

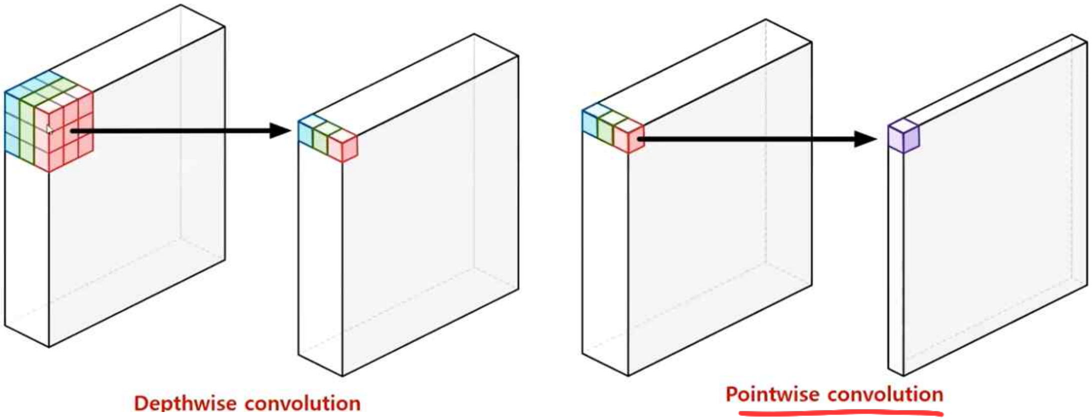
- Depthwise separable convolution = depthwise convolution + pointwise convolution (1*1 conv)로 구성



Original Depthwise Separable Convolution

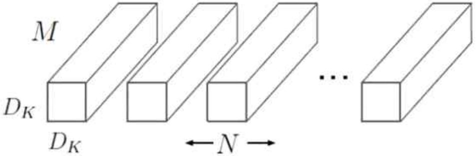
2. MobileNet Architecture

- Depthwise separable convolution = depthwise convolution + pointwise convolution (1*1 conv)로 구성

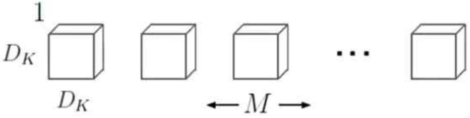


2. MobileNet Architecture

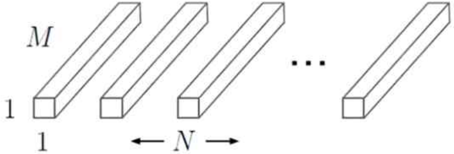
-기존 Convolution vs Depthwise Separable Convolution (filter비교)



(a) Standard Convolution Filters



(b) Depthwise Convolutional Filters



(c) 1×1 Convolutional Filters called Pointwise Convolution in the context of Depthwise Separable Convolution

dk = 필터의 가로와 세로 의미

m= input 채널 수

n = output채널 수 (필터 수)

2. MobileNet Architecture

- 기존 Convolution vs Depthwise Separable Convolution

- Standard convolutions have the computational cost of
  - $D_K \times D_K \times M \times N \times D_F \times D_F$
- Depthwise separable convolutions cost
  - $D_K \times D_K \times M \times D_F \times D_F + M \times N \times D_F \times D_F$
- Reduction in computations
  - $1 / N + 1 / D_K^2$
  - If we use 3x3 depthwise separable convolutions, we get between 8 to 9 times less computations

D_K : width/height of filters
D_F : width/height of feature maps
M : number of input channels
N : number of output channels(number of filters)

2. MobileNet Architecture

- Model Structure

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× Conv dw / s1	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
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	$1 \times 1 \times 1000$

비중

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%

3. Width Multiplier & Resolution Multiplier
(Hyperparameters)

- Width Multiplier – Thinner Models
  - For a given layer and width multiplier α, the number of input channels M becomes αM and the number of output channels N becomes αN – where α with typical settings of 1, 0.75, 0.6 and 0.25 $\alpha < 1$
- Resolution Multiplier – Reduced Representation
  - The second hyper-parameter to reduce the computational cost of a neural network is a resolution multiplier ρ
  - 0<ρ≤1, which is typically set of implicitly so that input resolution of network is 224, 192, 160 or 128(ρ = 1, 0.857, 0.714, 0.571)
- Computational cost:

$$D_K \times D_K \times \alpha M \times \rho D_F \times \rho D_F + \alpha M \times \alpha N \times \rho D_F \times \rho D_F$$

3. Width Multiplier & Resolution Multiplier
(Hyperparameters)

Table 3. Resource usage for modifications to standard convolution. Note that each row is a cumulative effect adding on top of the previous row. This example is for an internal MobileNet layer with $D_K = 3$, $M = 512$, $N = 512$, $D_F = 14$.

Layer/Modification	Million Mult-Adds	Million Parameters
Convolution	462	2.36
Depthwise Separable Conv	52.3	0.27
$\alpha = 0.75$	29.6	0.15
$\rho = 0.714$	15.1	0.15

4. Experiments & Results

Table 8. MobileNet Comparison to Popular Models

Model	ImageNet Accuracy	Million Mult-Adds	Million Parameters
1.0 MobileNet-224	70.6%	569	4.2
GoogleNet	69.8%	1550	6.8
VGG 16	71.5%	15300	138

Table 9. Smaller MobileNet Comparison to Popular Models

Model	ImageNet Accuracy	Million Mult-Adds	Million Parameters
0.50 MobileNet-160	60.2%	76	1.32
Squeezenet	57.5%	1700	1.25
AlexNet	57.2%	720	60

4. Experiments & Results

Table 10. MobileNet for Stanford Dogs

Model	Top-1 Accuracy	Million Mult-Adds	Million Parameters
Inception V3 [18]	84%	5000	23.2
1.0 MobileNet-224	83.3%	569	3.3
0.75 MobileNet-224	81.9%	325	1.9
1.0 MobileNet-192	81.9%	418	3.3
0.75 MobileNet-192	80.5%	239	1.9

Table 11. Performance of PlaNet using the MobileNet architecture. Percentages are the fraction of the Im2GPS test dataset that were localized within a certain distance from the ground truth. The numbers for the original PlaNet model are based on an updated version that has an improved architecture and training dataset.

Scale	Im2GPS [7]	PlaNet [35]	PlaNet MobileNet
Continent (2500 km)	51.9%	77.6%	79.3%
Country (750 km)	35.4%	64.0%	60.3%
Region (200 km)	32.1%	51.1%	45.2%
City (25 km)	21.9%	31.7%	31.7%
Street (1 km)	2.5%	11.0%	11.4%

PlaNet : 52M parameters, 5.74B mult-adds
MobilNet : 13M parameters, 0.58M mult-adds

4. Experiments & Results

Table 13. COCO object detection results comparison using different frameworks and network architectures. mAP is reported with COCO primary challenge metric (AP at IoU=0.50:0.05:0.95)

Framework Resolution	Model	mAP	Billion Mult-Adds	Million Parameters
SSD 300	deeplab-VGG	21.1%	34.9	33.1
	Inception V2	22.0%	3.8	13.7
	MobileNet	19.3%	1.2	6.8
Faster-RCNN 300	VGG	22.9%	64.3	138.5
	Inception V2	15.4%	118.2	13.3
	MobileNet	16.4%	25.2	6.1
Faster-RCNN 600	VGG	25.7%	149.6	138.5
	Inception V2	21.9%	129.6	13.3
	Mobilenet	19.8%	30.5	6.1

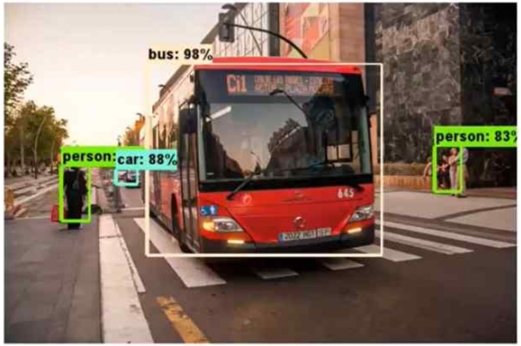


Figure 6. Example objection detection results using MobileNet SSD.

4. Experiments & Results

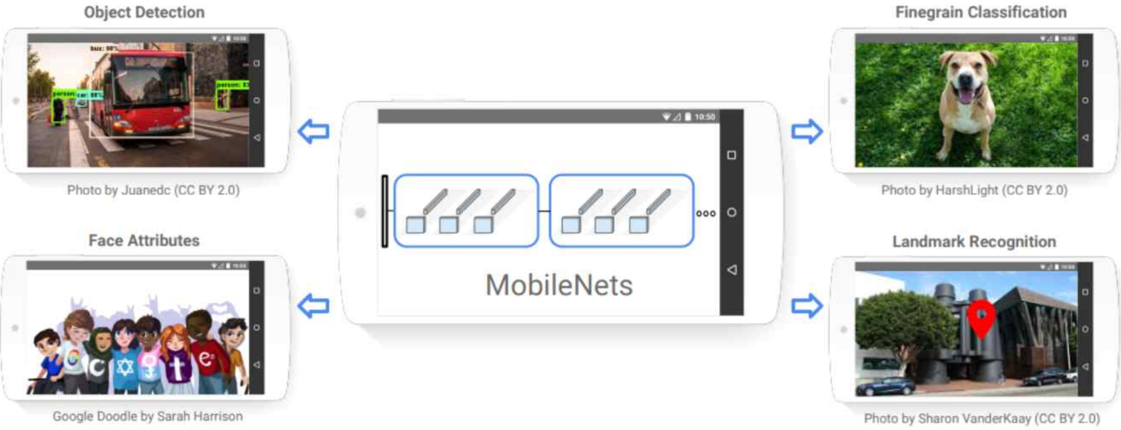


Figure 1. MobileNet models can be applied to various recognition tasks for efficient on device intelligence.

Thank you