

# MobileNet

## 0. Abstract

### 1. Depthwise separable convolutions

(깊이별 분리 합성곱)

## 1. Introduction

1. latency  $\downarrow$ , acc  $\uparrow \rightarrow$  2개의 hyper-parameter  
(resource)  
but, trade-off  $\exists$
- Width Multiplier ( $\alpha$ ) 넓이 배수  
Resolution Multiplier ( $\ell$ ) 해상도 배수

## 2. Prior Work

1. dataset의 디자인 fine-tuning net 양도 or 같은 네트워크  
(resource)  
 $\rightarrow$  "speed" 높이 필요

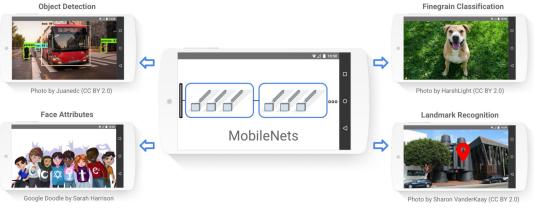


Figure 1. MobileNet models can be applied to various recognition tasks for efficient on-device intelligence.

## 3. MobileNet Architecture

### 3.1 Depthwise Separable Convolution

#### 1. Depthwise separable convolutions (factorized conv)

$\rightarrow$  Standard Conv  $\xrightarrow{\text{separate}}$  depthwise conv + pointwise conv

깊이별 분리 합성곱  $\rightarrow$  1x1 CONV  
(필터링) (축소)

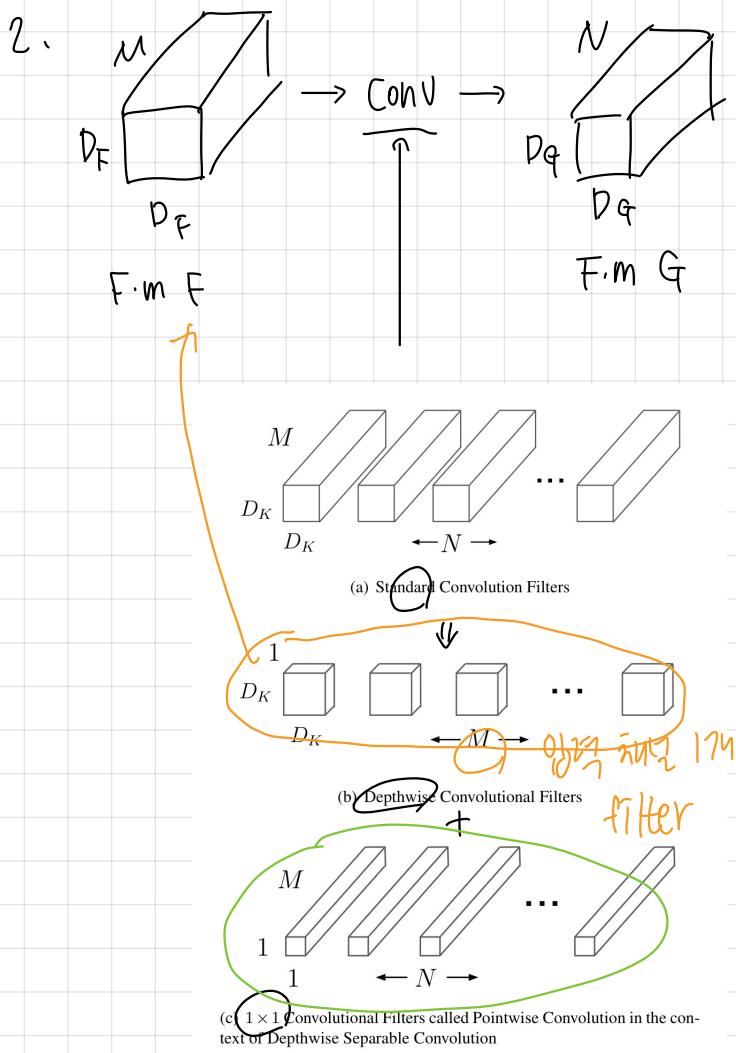


Figure 2. The standard convolutional filters in (a) are replaced by two layers: depthwise convolution in (b) and pointwise convolution in (c) to build a depthwise separable filter.

의미 f.m

$$(a) \frac{G_{k,l}}{\text{의미 f.m}} = \sum_{i,j,m} K_{i,j,m,n} \cdot F_{k+i-1, l+j-1, m}$$

conv Computational cost

$$D_K \cdot D_K \cdot M \cdot N \cdot D_F \cdot D_F$$

⇒ "depth wise separable"  
"깊이별 필터"

$$(b) \frac{\hat{G}_{k,l,m}}{\text{Chin } M \text{ 의 }} = \sum_{i,j} \frac{\hat{K}_{i,j,m}}{\text{depth-wise filter}} \cdot F_{k+i-1, l+j-1, m}$$

Chin M 의 filter

의미



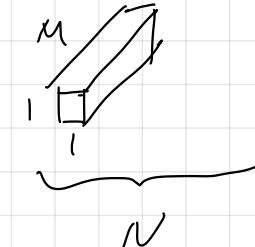
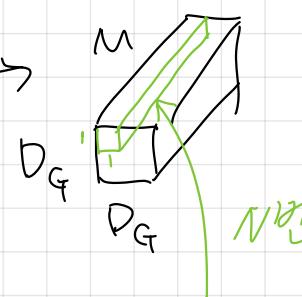
$$= D_K \times D_K \times M$$

$$\frac{D_F - D_K + 2P}{S} + 1$$

$$D_K \cdot D_K \cdot M \cdot D_F \cdot D_F$$

$$(c) D_K \cdot D_K \cdot M \cdot D_F \cdot D_F + M \cdot N \cdot D_F \cdot D_F$$

$$\frac{D_K \cdot D_K \cdot M \cdot D_F \cdot D_F + M \cdot N \cdot D_F \cdot D_F}{D_K \cdot D_K \cdot M \cdot N \cdot D_F \cdot D_F} = \frac{1}{N} + \frac{1}{D_K^2}$$

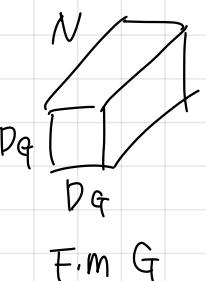


: 연산 후 각 다른 N개의  $|X| \times |X|$  의 filter

적용 후 합쳐지는 쓰임

(1개 필터당  $|X| \times |X| \rightarrow |X| \times N$ )

Concat



## 3.2 Network Structure and Training

1.

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           | $14 \times 14 \times 512$  |
|                 | $1 \times 1 \times 512 \times 512$   | $14 \times 14 \times 512$  |
| 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 $7 \times 7$                    | $7 \times 7 \times 1024$   |
| FC / s1         | $1024 \times 1000$                   | $1 \times 1 \times 1024$   |
| Softmax / s1    | Classifier                           | $1 \times 1 \times 1000$   |

→ 몇 층만 Pointwise conv  
7 층짜리 Depth-wise separable conv

→ HxW  $\approx 1^2$

2.

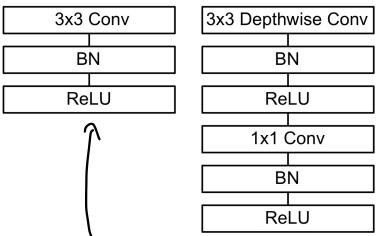


Figure 3. Left: Standard convolutional layer with batchnorm and ReLU. Right: Depthwise Separable convolutions with Depthwise and Pointwise layers followed by batchnorm and ReLU.

3.

Table 2. Resource Per Layer Type

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

↳ 1x1 conv가 차지하는 시간은 시간. Params

4. depthwise 계산에는 모듈러스가 적용으로 weight-decay 및 regularization 안 하는게 효율

### 3.3 Width Multiplier: Thinner Models

1. 더 짧은 계산 비용

2.  $\alpha$  = width multiplier

↳ 각 layer의 net는 줄임으로 **thin하게**

↳  $\frac{O}{\alpha}$  Ch  $M \rightarrow \alpha M$  ( $\alpha \in [0, 1]$ , generally  $1, 0.75, 0.5, 0.25$ )  
 $\frac{O}{\alpha}$  Ch  $N \rightarrow \alpha N$   
 base model < 1 : reduced

↳  $D_K \cdot D_K \cdot \alpha M \cdot D_F \cdot D_F + \alpha M \cdot \alpha N \cdot D_F \cdot D_F$

↳  $\alpha$ 의 차이로 계산 비용이 줄어들게 됨

### 3.4 Resolution Multiplier: Reduced Representation

1.  $\rho$  = resolution multiplier

↳ 단계별 줄임수 ( $\ell \in [0, 1]$ , 224/192/160/128 단계별  $\ell=1$  (base),  $\rho < 1$ : reduced)

2. 각각  $\alpha, \rho$  모두 계산 비용

↳  $D_K \cdot D_K \cdot \alpha M \cdot \rho D_F \cdot \rho D_F + \alpha M \cdot \alpha N \cdot \rho D_F \cdot \rho D_F$

↳  $\rho^2$  으로 계산 ↓

3. 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  $D_K = 3, M = 512, N = 512, D_F = 14$ . → 예제 f.w: (4x4x512 (standard: 3x3x512))

| Layer/Modification             | Million<br>Mult-Adds | Million<br>Parameters |
|--------------------------------|----------------------|-----------------------|
| (standard) fully - Convolution | 462                  | 2.36                  |
| Depthwise Separable Conv       | 57.3                 | 0.27                  |
| $\alpha = 0.75$                | 29.6                 | 0.15                  |
| $\rho = 0.714$                 | 15.1                 | 0.05                  |

## 4. Experiments

### 4.1 Model choices / 4.2 Model Shrinking Hyperparameters

1. 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 5. Narrow vs Shallow MobileNet ( $\alpha$ )

| Model             | ImageNet Accuracy | Million Mult-Adds | Million Parameters |
|-------------------|-------------------|-------------------|--------------------|
| 0.75 MobileNet    | 68.4%             | 325               | 2.6                |
| Shallow MobileNet | 65.3%             | 307               | 2.9                |

Table 6. MobileNet Width Multiplier ( $\alpha$ )

| 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 ( $\rho$ )

| 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                |

ex) 7% img 224  $\rightarrow \rho = 1$

192  $\rightarrow \rho \approx 0.86 \rightarrow \rho \uparrow \text{Mult-Add} \downarrow \text{acc} \downarrow$   
 $\vdots$   
 $\text{trade-off}$

2.  $\alpha \in [1, 0.75, 0.5, 0.25]$  ] [62n Q[1]]  
 $H \times W \in [224, 192, 160, 128]$

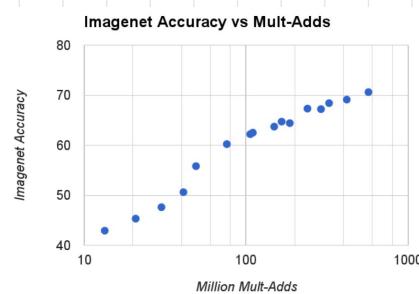


Figure 4. This figure shows the trade off between computation (Mult-Adds) and accuracy on the ImageNet benchmark. Note the log linear dependence between accuracy and computation.

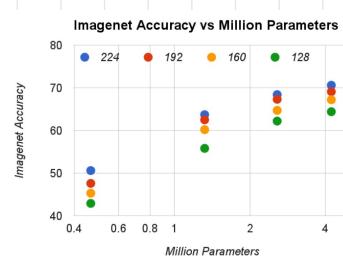


Figure 5. This figure shows the trade off between the number of parameters and accuracy on the ImageNet benchmark. The colors encode input resolutions. The number of parameters do not vary based on the input resolution.

3

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 |
|-------------------|-------------------|-------------------|--------------------|
| 0.5 MobileNet-160 | 60.2%             | 76                | 1.32               |
| SqueezeNet        | 57.5%             | 1700              | 1.25               |
| AlexNet           | 57.2%             | 720               | 60                 |

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%            |

4.3 ~ 4.7 %

Table 14. MobileNet Distilled from FaceNet

| Model              | 1e-4 Accuracy | Million Mult-Adds | Million Parameters |
|--------------------|---------------|-------------------|--------------------|
| FaceNet [25]       | 83%           | 1600              | 7.5                |
| 1.0 MobileNet-160  | 79.4%         | 286               | 4.9                |
| 1.0 MobileNet-128  | 78.3%         | 185               | 5.5                |
| 0.75 MobileNet-128 | 75.2%         | 166               | 3.4                |
| 0.75 MobileNet-128 | 72.5%         | 108               | 3.8                |

Table 12. Face attribute classification using the MobileNet architecture. Each row corresponds to a different hyper-parameter setting (width multiplier  $\alpha$  and image resolution).

| Width Multiplier / Resolution | Mean AP | Million Mult-Adds | Million Parameters |
|-------------------------------|---------|-------------------|--------------------|
| 1.0 MobileNet-224             | 88.7%   | 568               | 3.2                |
| 0.5 MobileNet-224             | 88.1%   | 149               | 0.8                |
| 0.25 MobileNet-224            | 87.2%   | 45                | 0.2                |
| 1.0 MobileNet-128             | 88.1%   | 185               | 3.2                |
| 0.5 MobileNet-128             | 87.7%   | 48                | 0.8                |
| 0.25 MobileNet-128            | 86.4%   | 15                | 0.2                |
| Baseline                      | 86.9%   | 1600              | 7.5                |

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                |

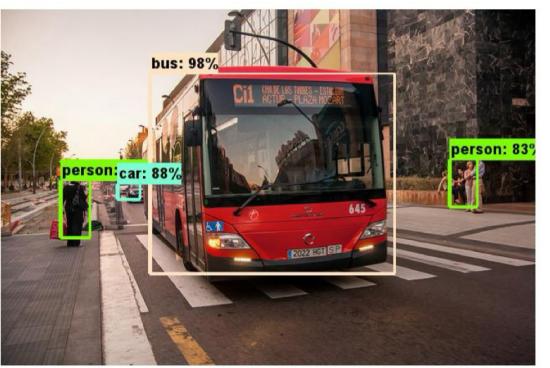


Figure 6. Example objection detection results using MobileNet SSD.

5. conclusions