MobileNets

Seunghun Cho

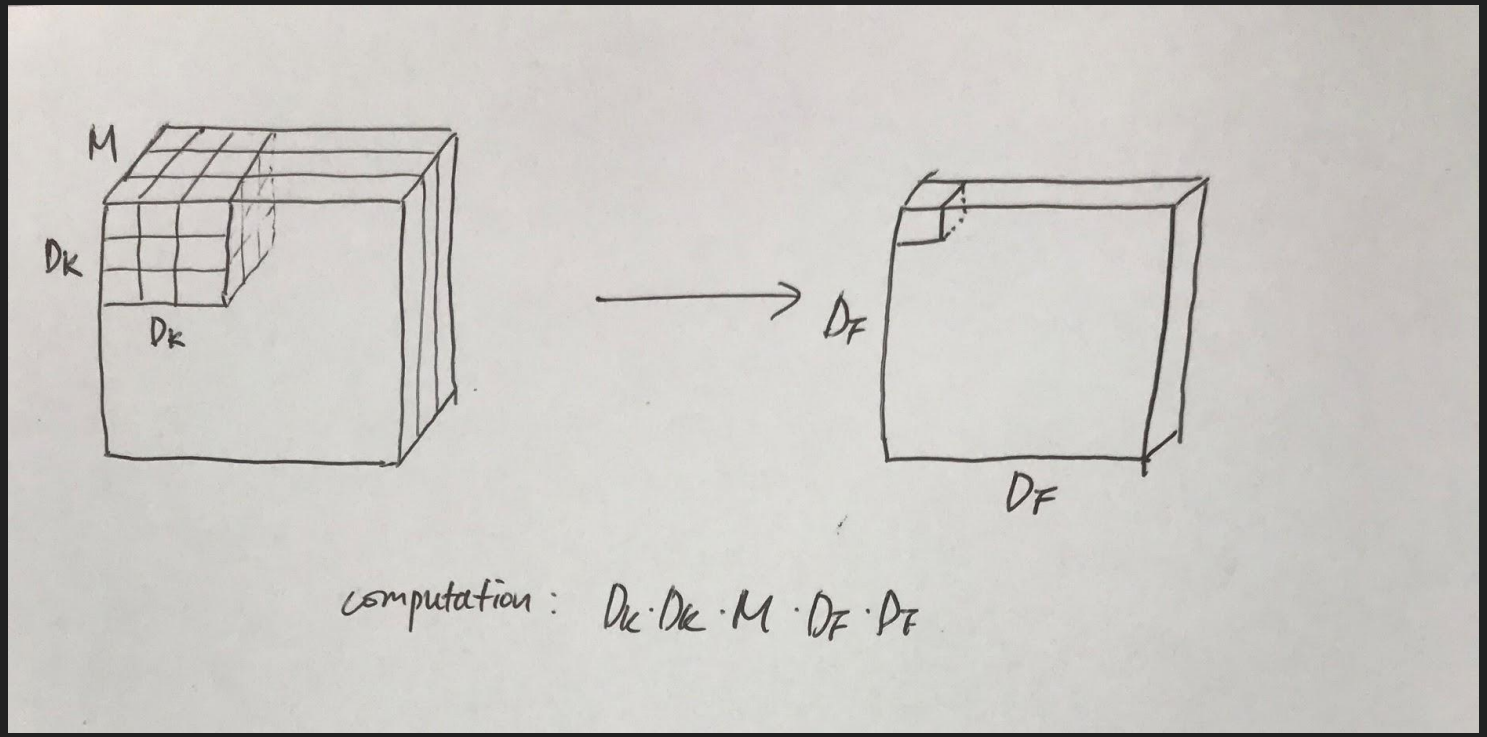
목적

Efficient한 Convolutional neural network 만들기

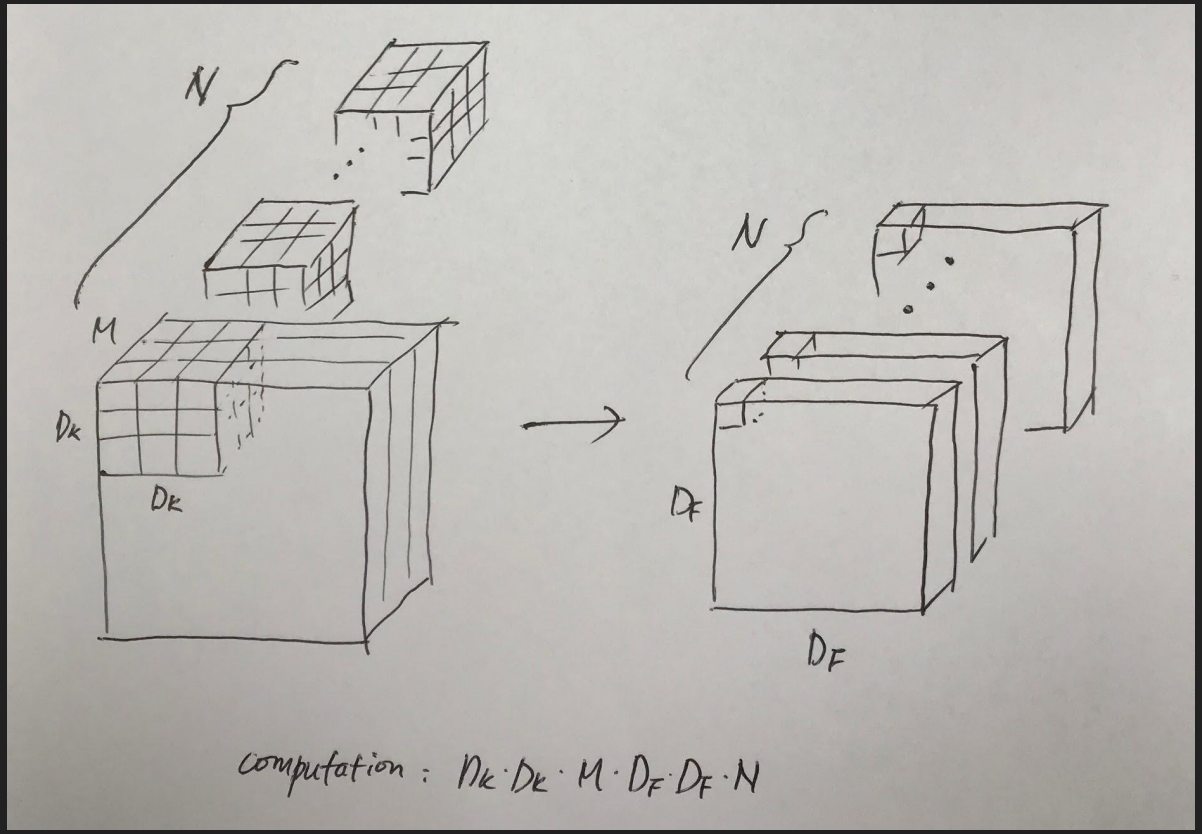
Prior Works

- Flattened networks
- Factorized networks
- Xception
- Squeezenet

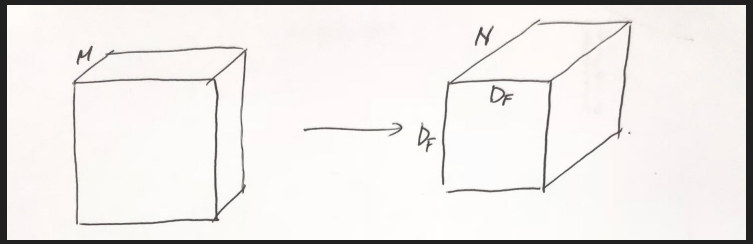
Recap: Standard Convolutional Neural Network (1/2)



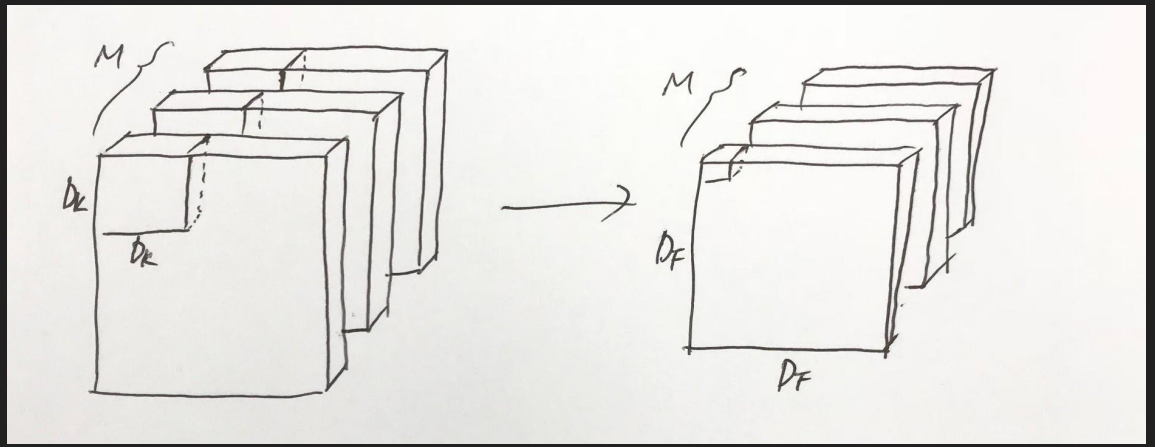
Recap: Standard Convolutional Neural Network (2/2)



MobileNets (1/6)



Step 1

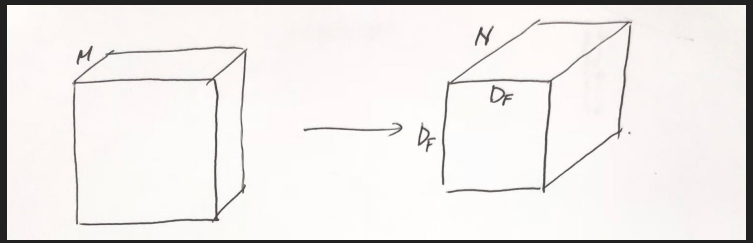


Computation:

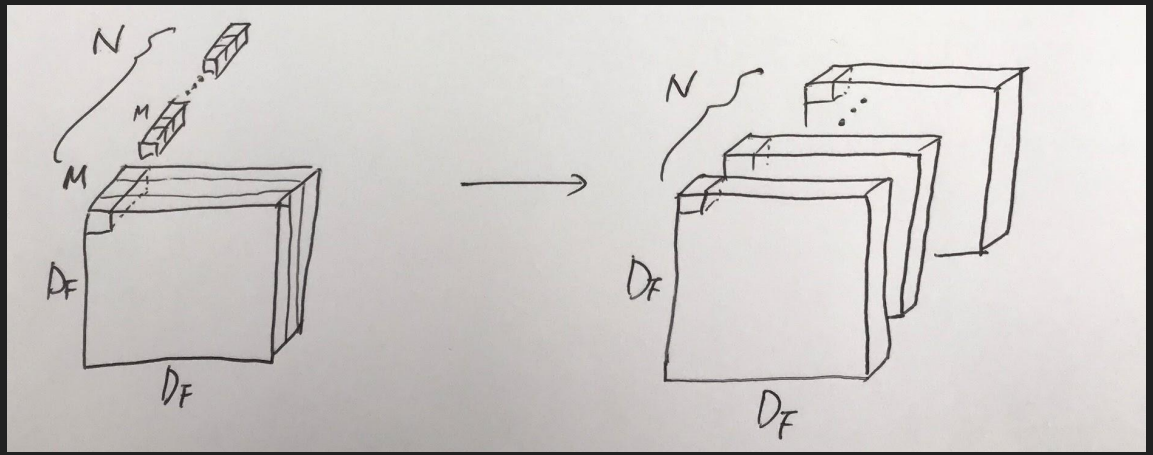
Dk * Dk * Df * Df * M

:= Depthwise Convolution

MobileNets (2/6)



Step 2

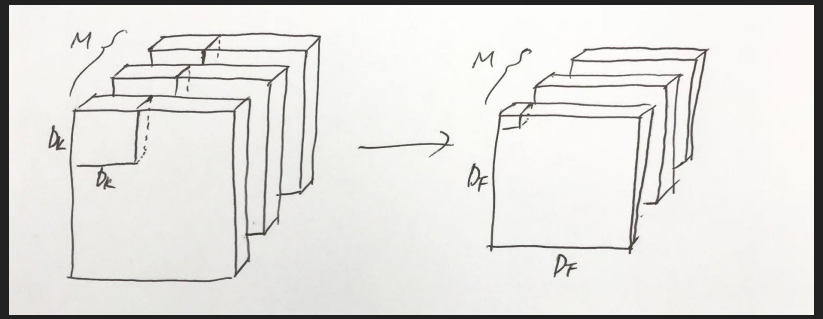


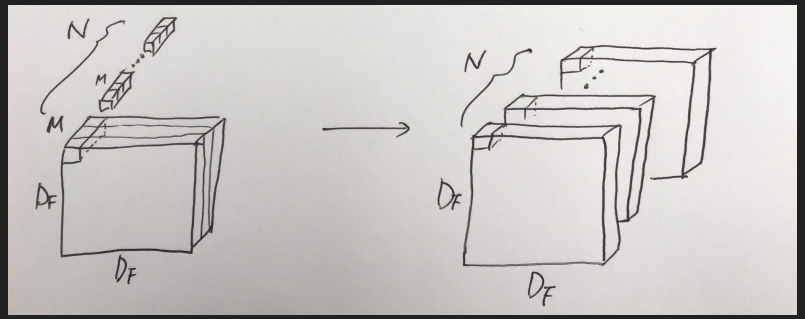
Computation:

1 * 1 * M * Df * Df * N

:= Pointwise Convolution

MobileNets (3/6)





Total Computation:

Dk * Dk * Df * Df * M + 1 * 1 * M * Df * Df * N

MobileNets (4/6): Computation Cost

$$\frac{\text{Depthwise + Pointwise Convolution}}{\text{Standard Convolution}} = \frac{Dk * Dk * Df * Df * M + 1 * 1 * M * Df * Df * N}{Dk * Dk * Df * Df * M * N}$$

$$= \frac{1}{N} + \frac{1}{Dk * Dk}$$

MobileNets (5/6): Architecture

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	$1 \times 1 \times 1000$

Full convolution

MobileNets (6/6): Performance

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

Computation cost 88.30% 감소

Usage - Tensorflow / Keras

Example

```
inputs = Input(shape=(30, 30, 3))

x = DepthwiseConvolution2D(3, (3, 3))(inputs)
x = Conv2D(6, (1, 1))(x)

model = Model(inputs=inputs, outputs=x)

model.summary()
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_2 (InputLayer)         (None, 30, 30, 3)         0
_________________________________________________________________
depthwise_conv2d_2 (Depthwis (None, 28, 28, 3)         30
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 28, 28, 6)         24
=================================================================
Total params: 54
Trainable params: 54
Non-trainable params: 0
_________________________________________________________________
```

https://www.tensorflow.org/api_docs/python/tf/nn/depthwise_conv2d
https://keras.io/layers/convolutional/#separableconv2d