MobileNet

MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications

이 논문에서의 핵심 기술은 3가지가 있습니다.

- Depthwise separable convolution
- Width multiplier
- Resolution multiplier

그 중, Depthwise separable convolution를 가장 핵심적으로 다루고 있습니다.

INTRODUCTION

Convolution Neural Network는 일반적으로 더 높은 정확도를 얻기 위해 더 깊고 복잡한 네트워크를 만드는 추세가 있습니다. 하지만, 정확성을 높이기 위해서 더 깊고 복잡한 네트워크를 만드는 것은 반드시 좋은 모델이 되는 것은 아닙니다.

컴퓨터 비전에 대한 상업적인 요구는 다음과 같습니다. (+추가적인 내용)

1. Data-centers(clouds)

- Rarely safety-critical 안전성 요구
- Low power is nice to have 저전력
- Real-time is preferable 실시간

1. Gadgets= Smartphones, Self,driving cars, Drones, etc.

- Usually safety-critical(except smartphone)
- Low power is must-have
- Real-time is required

우리가 실제 만든 모델을 사용하는 작업을 생각해보면 로봇, 자율주행차, 증강현실 등 이러한 작업은 제한된 플랫폼에서 실시간으로 작업을 수행해주어야 합니다. 그렇기 때문에, 다양한 플랫폼에서 쉽게 사용할 수 있고 빠른 작업 처리를 할 수 있는 모델이 필요합니다.

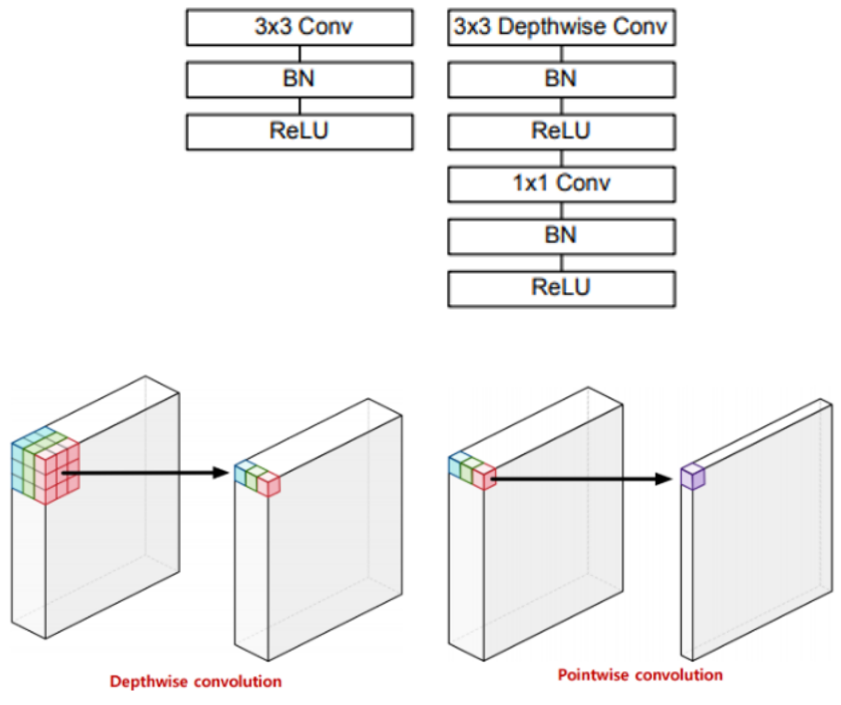
그래서 우리는 일부 구조 변경과, 2개의 하이퍼 파라미터를 기존 모델에서 추가하여 아키텍처를 더 작고 효율적으로 만든 MobileNet을 제안합니다.

MobileNet Architecture

- **Depthwise Separable Convolution**

MobileNet 모델은 depthwise separable convolutions을 기반으로 합니다.

기존 Convolution layer 3*3을 진행하는 방식 대신, Depthwise Conv 3*3을 진행한 뒤, 1*1 Conv를 진행하는 방식이다.



여기서 Pointwise convolution이 1*1 Conv와 동일하다. MobuleNet은 이와 같은 두 번의 걸친 layer계산을 통해 연산량과 파라미터 수를 줄였습니다.

기존의 3*3 conv 같은 경우의 계산량은 $D_k * D_k * M * N * D_F * D_F$와 같은데 'Dk'는 커널의 크기, 'Df'는 피쳐맵의 크기, M과 N은 INPUT, OUTPUT 채널 수입니다. 하지 만 이와 같은 연산량은 다음과 같이 변합니다.

DK · DK · M · N · DF · DF이는 Depthwise Conv 3*3 레이어의 계산량이고, DK · DK · M · DF · DF 이는 1*1 Conv의 계산량이 됩니다. 이를 합하면 총 DK · DK · M · DF · DF + M · N · DF · DF 계산량을 얻게 됩니다.

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

이는, 기존 3*3 Conv보다 배만큼의 효율성을 얻어 낼 수 있고 이후 실험 결과에서 표준 8배 ~9배 정도의 적은 계산을 사용합니다.

input : 3 x 28 x 28 filter : 3 x 3output : 32 x 28 x 28 Dk = 3, M = 3, N = 32

Standard Convolution = Dk x Dk x M x N = 3 x 3 x 3 x 32 = 864개

Depthwise Separable Convolution = Dk x Dk x 1 x M + 1 x 1 x M x N = 3 x 3 x 1 x 3 + 1 x 1 x 3 x 32 = 123개

- **Network Structure and Training**

MobileNet에서는 첫 번째 레이어는 표준의 컨볼루션 레이어를 사용하고, 나머지는 depthwise separable convolutions를 사용합니다. 최종 fc_layer를 제외하고 모든 레이어 뒤에는 batch_normalization과 ReLU활성화 함수가 사용됩니다. Down_sampling은 standard convolution layer와 depthwise separable convolution layer에도 전부 사용됩니다. 마지막에서는 averaging pooling을 통해 7*7을 1*1로 전환해줍니다. 모든 층의 개수를 세면 MobileNet에는 28개의 계층이 있습니다. MobileNet은 다음 사진과 같은 구조를 따릅니다.

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	$1 \times 1 \times 1000$

- **Width Multiplier: Thinner Models**
- **Resolution Multiplier: Reduced Representation**

MobileNet에서는 두 가지 하이퍼 파라미터를 더 추가하였는데 하나는 channel의 개수를 조정하는 Width Multiplier와 이미지의 해상도를 조절하는 Resolution Multiplier이다.

Channel의 개수를 조절하는 파라미터의 전형적인 값은 1, 0.75, 0.5 and 0.25으로 세팅을 하였고, 이미지의 해상도를 조정하는 파라미터의 전형적인 값은 224, 192, 160 or 128로 한다. 그렇게 하면 레이어의 연산량은 DK · DK · αM · ρDF · ρDF + αM · αN · ρDF · ρDF이 된다. α는 Width Multiplier, ρ는 Resolution Multiplier를 의미한다. 이미지의 해상도가 기본 1*1이라 하고 ρ를 0.75라 한다면 ρ^2배만큼의 연산량으로 줄일 수 있게 된다. 두 개의 하이퍼 파라미터를 사용하여 연산량을 다시금 줄일 수 있게 된다.

Experiments

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

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

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

다음과 같은 그림의 결과에서 MobileNet을 사용하였을 때, 정확도는 약간 떨어지지만Million Mult-유사하고 Adds(한 이미지를 인식하는데 필요한 곱셈-합 연산 횟수)와 Million Parameters(사용한 파라미터 수)가 현저히 적어진 것을 볼 수 있다. Channel과 Resolution을 줄이는 작업을 했을 때 또한 모델의 속도가 빨라진 것을 확인할 수 있다.

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

MobileNet은 GoogleNet과 VGG 16과 정확도가 유사하며 모델은 훨씬 가벼운 것을 볼 수 있습니다.

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%

또한, 다른 유명한 모델과 비교하였을 때 일부 특정 종류의 이미지마다 성능이 좋은 경우도 존재하였습니다.

그리고 MobileNet은 객체 탐지, 얼굴 인식과 같은 작업 또한 준수하고 빠른 작업이 가능했습니다.

Conclusion

depthwise separable convolutions을 기반으로 한 MobileNet을 제안하였다. 그리고 이를 통해 효율적인 모델이 고안되었으며 width multiplier와 resolution multiplier을 사용하여 더 작고 빠른 MobileNets을 구성하였다. 다른 다양한 모델과도 비교를 해보았을 때 우리는 다양한 작업에서 이를 사용할 수 다고 결론을 내렸다.