# MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications

Google Inc.

reporter：Manting Xie

adviser：Yizhen Lao

# contents

- MobileNets are based on a streamlined architecture that uses depth-wise separable convolutions to build light weight deep neural networks.
- MobileNet Comparison to Popular Models
- MobileNet architecture and two hyper-parameters width multiplier and resolution multiplier
- different applications and use cases

# PART ONE

## background

Convolutional neural networks have become ubiquitous in computer vision ever since AlexNet [19] popularized deep convolutional neural networks by winning the ImageNet Challenge: ILSVRC 2012

## trend

The general trend has been to make deeper and more complicated networks in order to achieve higher accuracy

## problem

these advances to improve accuracy are not necessarily making networks more efficient with respect to size and speed

# PART TWO

MobileNets are a class of network architectures which allows a model developer to specifically choose a small network that matches the resource restrictions (latency, size) for their application. MobileNets primarily focus on optimizing for latency but also yield small networks

01

02

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 8 compares full MobileNet to the original GoogleNet and VGG16. MobileNet is nearly as accurate as VGG16 while being 32 times smaller and 27 times less compute intensive. It is more accurate than GoogleNet while being smaller and more than 2.5 times less computation

03

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 |

Reduced MobileNet is 4% better than AlexNet while being 45× smaller and 9.4× less compute than AlexNet. It is also 4% better than Squeezenet at about the same size and 22× less computation

# PART THREE

# Depthwise Separable Convolution

1. The core layers that MobileNet is built on which are depthwise separable filters

The MobileNet model is based on depthwise separable convolutions which is a form of factorized convolutions which factorize a standard convolution into a depthwise convolution and a 1×1 convolution called a pointwise convolution. For MobileNets the depthwise convolution applies a single filter to each input channel. The pointwise convolution then applies a 1×1 convolution to combine the outputs the depthwise convolution.

This factorization has the effect of drastically reducing computation and model size.

# MobileNet Architecture

2 The MobileNet structure is built on depthwise separable convolutions as mentioned in the previous section except for the first layer which is a full convolution

The MobileNet architecture is defined in Table 1. All layers are followed by a batchnorm [13] and ReLU nonlinearity with the exception of the final fully connected layer which has no nonlinearity and feeds into a softmax layer for classification.

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

# Width Multiplier: Thinner Models

3

Although the base MobileNet architecture is already small and low latency, many times a specific use case or application may require the model to be smaller and faster. In order to construct these smaller and less computationally expensive models we introduce a very simple parameter α called **width multiplier**

**The role** of the width multiplier is to thin a network uniformly at each layer. Width multiplier has the effect of reducing computational cost and the number of parameters quadratically by roughly . Width multiplier can be applied to any model structure to define a new smaller model with a reasonable accuracy, latency and size trade off

# Resolution Multiplier: Reduced Representation

4 The second hyper-parameter to reduce the computational cost of a neural network is a **resolution multiplier ρ**.

We apply this to the input image and the internal representation of every layer is subsequently reduced by the same multiplier. In practice we implicitly set ρ by setting the input resolu-tion.

# PART FOUR

# MobileNets applied to a number of different applications


We train MobileNet for fine grained recognition on the Stanford Dogs dataset. We extend the approach of he unreasonable effectiveness of noisy data for fine-grained recognition and collect an even larger but noisy training set from the web. We use the noisy web data to pretrain a fine grained dog recognition model and then fine tune the model on the Stanford Dogs training set.

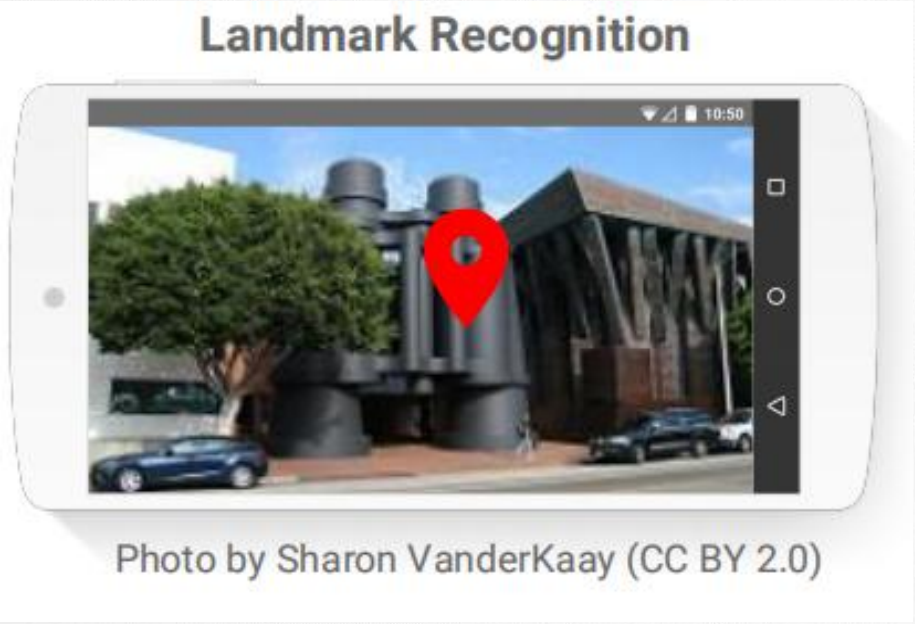
Table 10. MobileNet for Stanford Dogs

| 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 |

Results on Stanford Dogs test set are in Table 10.

MobileNet can almost achieve the state of the art results from [18] at greatly reduced computation and size.

# Large Scale Geolocalizaton




PlaNet casts the task of determining where on earth a photo was taken as a classification problem. The approach divides the earth into a grid of geographic cells that serve as the target classes and trains a convolutional neural network on millions of geo-tagged photos. PlaNet has been shown to successfully localize a large variety of photos and to outperform Im2GPS.

| 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% |

The full PlaNet model based on the Inception V3 architecture [31] has 52 million parameters and 5.74 billion mult-adds. The MobileNet model has only 13 million parameters.
the MobileNet version delivers only slightly decreased performance compared to PlaNet despite being much more compact. Moreover, it still outperforms Im2GPS by a large margin.

# T H A N K S

2021 . 12 . 14