# A Model for Mobile Vision Applications

Hongzhi Liu

May 19, 2018

## 1 The Background of MobileNets Models

In recent years, convolutional neural networks (CNNs) have become ubiquitous in computer vision. There has been rising interest in building small and efficient neural networks. In other word, the general trend has been to make deeper and more complicated networks in order to achieve higher accuracy but those advances are not necessarily making networks more efficient with respect to size and speed. In many real world applications such as robotics, self-driving car and augmented reality, the recognition tasks need to be carried out in a timely fashion on a computationally limited platform. However, many methods which on small networks in the recent literature [1–3] focus only on size but do not consider speed.

In the thesis of Professor Howard, he presents a class of efficient models called MobileNets for mobile and embedded vision applications [4] which are based on a streamlined architecture that uses depthwise separable convolutions to build light weight deep neural networks. Furthermore, Professor Howard describes a set of two hyper-parameters in order to build very small, low latency models that can be easily matched to the design requirements for mobile and embedded vision applications. The effectiveness of MobileNets have been demonstrated across a wide range of applications and use cases as shown in Figure 1, including object detection, finegrain classification, face attributes and large scale geo-localization.

## 2 MobileNet Architecture

The MobileNet model, mentioned in the paper, based on depthwise separable convolutions which is a form of factorized convolutions that factorize a standard convolution into a depthwise convolution and a $1 \times 1$ convolution called a pointwise convolution.

A standard convolution both filters and combines inputs into a new set of outputs in one step. The depthwise separable convolution splits this into two layers, a separate layer for filtering and a separate layer for combining. This factorization has the effect of drastically reducing computation and model size. Figure 2 shows how a standard convolution 2(a) is factorized into a depthwise convolution 2(b) and a $1 \times 1$ pointwise convolution 2(c).

A standard convolutional layer takes as input a $D_F \times D_F \times M$ feature map $F$ and produces a $D_G \times D_G \times N$ feature map $G$ where $D_F$ is the spatial width and height of a square input feature map, $M$ is the number of input channels, $D_G$ is the spatial width and height of a square output feature map and $N$ is the number of output channel. Standard convolutions have the computational cost of as Equation 1:

$$D_K \times D_K \times M \times N \times D_F \times D_F \tag{1}$$

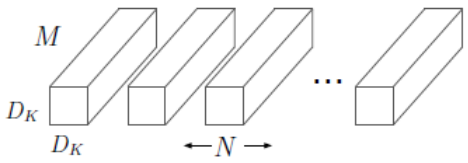
where the computational cost depends multiplicatively on the number of input channels $M$, the number of output channels $N$ the kernel size $D_K \times D_K$ and the feature map size $D_F \times D_F$. MobileNet models address each of these terms and their interactions. First it uses depthwise separable convolutions to break the interaction between the number of output channels and the size of the kernel.

Depthwise separable convolution has a computational cost of as Equation 2:

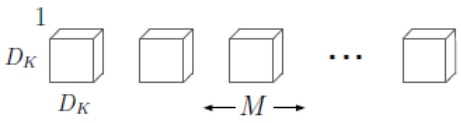
$$D_K \times D_K \times M \times D_F \times D_F + M \times N \times D_F \times D_F \tag{2}$$

By expressing convolution as a two step process of filtering and combining they get a reduction in computa-

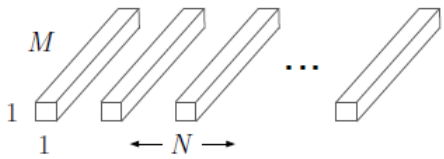

Figure 1: MobileNet models can be applied to various recognition tasks for efficient on device intelligence.




(a) Standard Convolution Filters




(b) Depthwise Convolutional Filters




(c) $1 \times 1$ Convolutional Filters called Pointwise Convolution in the context of Depthwise Separable Convolution

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.

tion as Equation 3:

$$\frac{D_K \times D_K \times M \times D_F \times D_F + M \times N \times D_F \times D_F}{D_K \times D_K \times M \times N \times D_F \times D_F} = \frac{1}{N} + \frac{1}{D_K^2} \quad (3)$$

In order to construct these smaller and less computationally expensive models they introduce a very simple parameter called width multiplier. The computational cost of a depthwise separable convolution with width multiplier is as Equation 4:

$$D_K \times D_K \times \alpha M \times D_F \times D_F + \alpha M \times \alpha N \times D_F \times D_F \quad (4)$$

where $0 < \alpha \leq 1$ with typical settings of 1, 0.75, 0.5 and 0.25. $\alpha = 1$ is the baseline MobileNet and $\alpha < 1$ are reduced MobileNets.

The second hyper-parameter to reduce the computational cost of a neural network is a resolution multiplier $\rho$. They express the computational cost for the core layers of our network as depthwise separable convolutions with width multiplier $\alpha$and resolution multiplier $\rho$ Equation 5:

$$D_K \times D_K \times \alpha M \times \rho D_F \times \rho D_F + \alpha M \times \alpha N \times \rho D_F \times \rho D_F \quad (5)$$

where $0 < \rho \leq 1$ which is typically set implicitly so that the input resolution of the network is 224, 192, 160 or 128. $\rho = 1$ is the baseline MobileNet and $\rho < 1$ are reduced computation MobileNets. Resolution multiplier has the effect of reducing computational cost by $\rho^2$.

## 3 Evaluation of the Method

Professor Howard trains PlaNet using the MobileNet architecture on the same data. While the full PlaNet model based on the Inception V3 architecture [5] has 52 million parameters and 5.74 billion mult-adds. The MobileNet model has only 13 million parameters with the usual 3 million for the body and 10 million for the final layer and 0.58 Million mult-adds. As shown in Tab. 1, 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.

Table 1: Performance of PlaNet [6] using the MobileNet architecture.Percentages are the fraction of the Im2GPS [7] 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 | PlaNet | 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% |

## References


[1] Forrest N. Iandola, Matthew W. Moskewicz, Khalid Ashraf, Song Han, William J. Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <1mb model size. *CoRR*, abs/1602.07360, 2016.

[2] Jonghoon Jin, Aysegul Dundar, and Eugenio Culurciello. Flattened convolutional neural networks for feedforward acceleration. *CoRR*, abs/1412.5474, 2014.

[3] Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi. *XNOR-Net: ImageNet Classification Using Binary Convolutional Neural Networks*. Springer International Publishing, 2016.

[4] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *CoRR*, abs/1704.04861, 2017.

[5] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, 2016.

[6] Tobias Weyand, Ilya Kostrikov, and James Philbin. Planet - photo geolocation with convolutional neural networks. pages 37–55, 2016.

[7] Jaeyoung Choi and Gerald Friedland. *Multimodal Location Estimation of Videos and Images*. Springer International Publishing, 2015.