**MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications**

*Andrew G. Howard, et al.*

* Source: <https://arxiv.org/pdf/1704.04861.pdf>
* MobileNet is designed specifically for vision applications on mobile and embedded devices.
* Suitable for robotics, self-driving cars, augmented reality, etc.
* The main feature of the model is “Depthwise Separatable Convolutions”
* Depthwise Separable Convolution (will call it DSC here onwards in this document) is basically created by splitting the standard conv. operation into two steps: depthwise convolution and pointwise convolution.

*Intuition:*

*A standard conv. layer has many 3D filters. Each filter works on all the input channels. So, there is filtering and combining results from various input channels.*

*In DSC, depthwise conv. layer filters data from each channel individually (i.e. there is one filter per input channel), and then many pointwise filters combine the results from all the channels.*

* The paper also gives two parameters that allow the MobileNet to be shrinked as per your use case and available resources.
* Depthwise convolution:

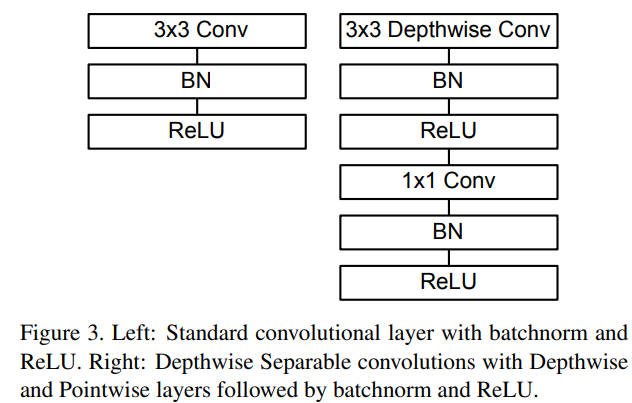
In standard conv. layer, each filter is 3D, and it spans all the channels of the input. However, in depthwise conv, each filter is 2D, and it applies only to a specific channel. So, there is only one filter per input channel.

So, the output of such conv. operation will have the same no. of channels as in the input.

* Pointwise convolution:

This is same as the standard convolution, but the spatial size of the filters is fixed to 1\*1. The third dimension of each filter is same as the no. of channels in the input, just as in a standard conv. layer.

* So, a standard conv. layer involves filtering and combining results from various channels. In DSC, filtering is done by a depthwise conv. layer and combining results from channels is done by a pointwise conv. layer.



* DSC has far less no. of computation compared to standard conv. layer

e.g.

*Input shape = (Im, In, Nc)*

*Filter size = (Fm, Fn, Nc)*

*No. of filters = K*

*Output shape = (Om, On, K)*

No. of computations in a standard conv. layer:

*(Fm \* Fn \* Nc) \* (Om \* On) \* K*

No. of computations in Depthwise conv. layer:

*(Fm \* Fn \* 1) \* (Om \* On) \* Nc*

No. of computations in Pointwise conv. layer:

*(1 \* 1 \* Nc) \* (Om \* On) \* K*

Total no. of computations in DSC:

*(Fm \* Fn) \* (Om \* On) \* Nc + Nc \* (Om \* On) \* K*

Typically, we have

* MobileNet uses 3 × 3 depthwise separable convolutions, so the overall computations is about 9 times less than a network of same depth using standard conv. layers.
* First layer in MobileNet is standard conv. layer
* All layers, except the output layer, are followed by batch norm and ReLU.
* The last layer is linear (i.e. without any activation function) and is passed through softmax.
* Global average pooling layer is used before the last FC layer.
* Counting depthwise and pointwise convolutions as separate layers, MobileNet has 28 layers
* Training: tensorflow; RMSprop; less regularization and data augmentation as the model is small
* Either no weight decay or very small decay in depthwise conv. layers
* Width multiplier ():

The role of the width multiplier is to thin a network uniformly at each layer.

For a given layer and width multiplier , the number of input channels becomes and the number of output channels becomes . Because of this, the no. of computations also decreases in DSC by roughly

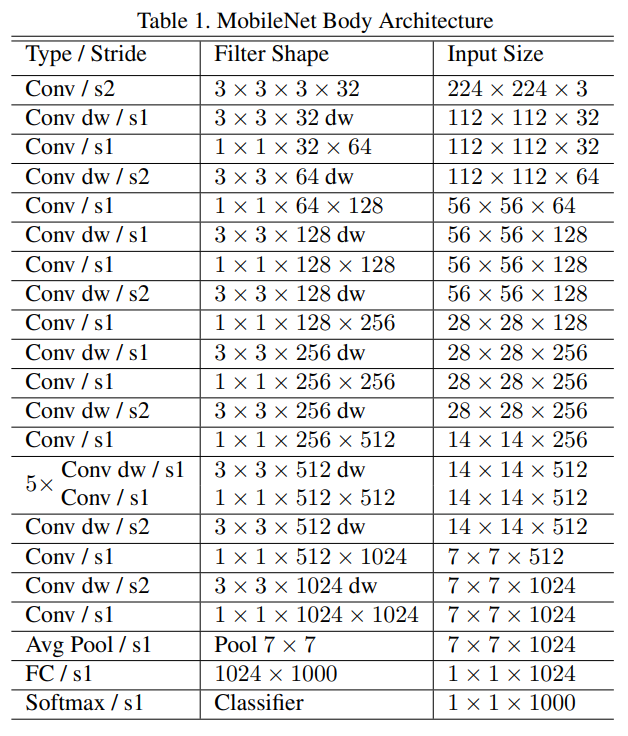
, with typical settings of 1, 0.75, 0.5 and 0.25.

means the baseline MobileNet model.

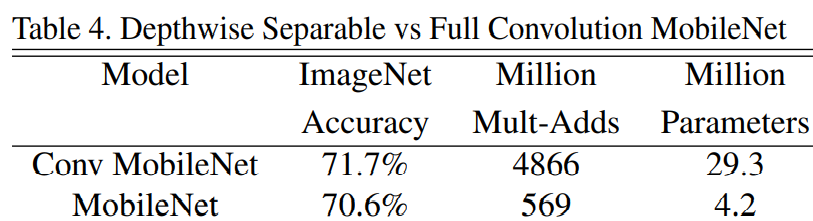
* Resolution multiplier ():

Apply this to input image resolution, and as a result the internal representation of every layer is also reduced. This also reduces the no. of computations by roughly

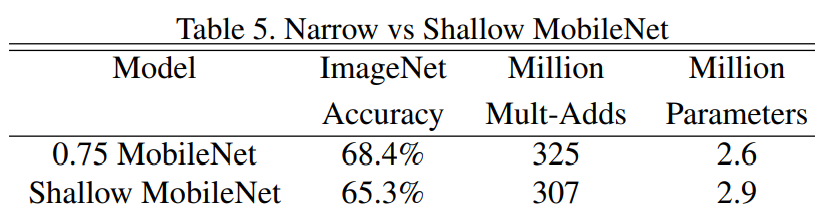
, which is set in such a way that the resolution is one of 224, 192, 160, 128

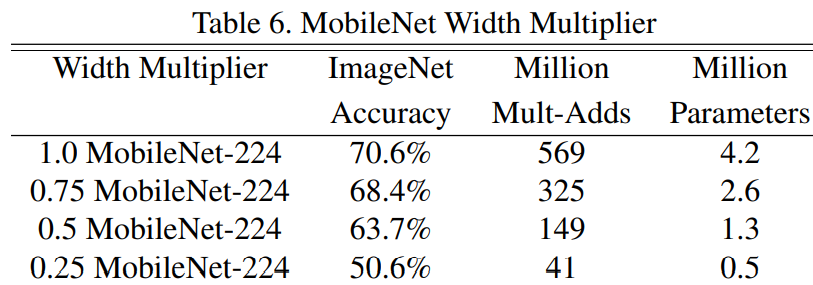
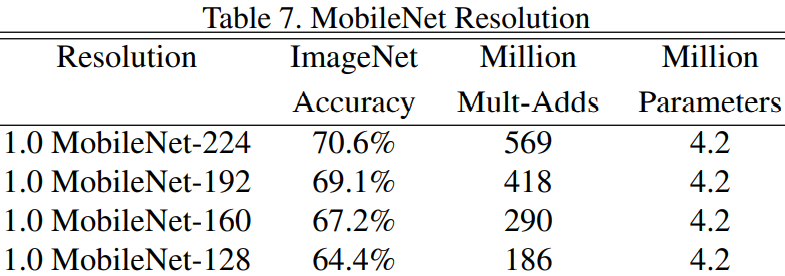


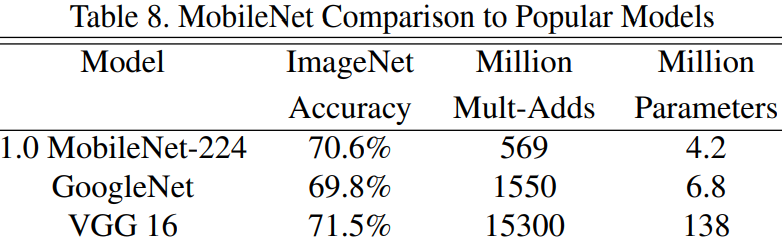
* Some comparisons:
  + Compared to full conv., using depthwise separable conv. reduces accuracy by only 1% on ImageNet but saves tremendously on mult-adds and parameters.



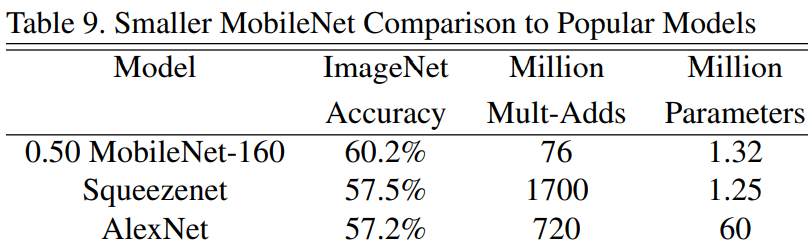
* + Thinner models beat shallower models.



* + 
  + 
  + 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.



* + 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.



* + Similarly, MobileNet was tested on Stanford Dogs, PlaNet, Object detection, Face Embeddings, etc. and was found to produce accuracy close to state-of-the-art, with much less computations and time.