The Math Behind MobileNets Efficient Computation

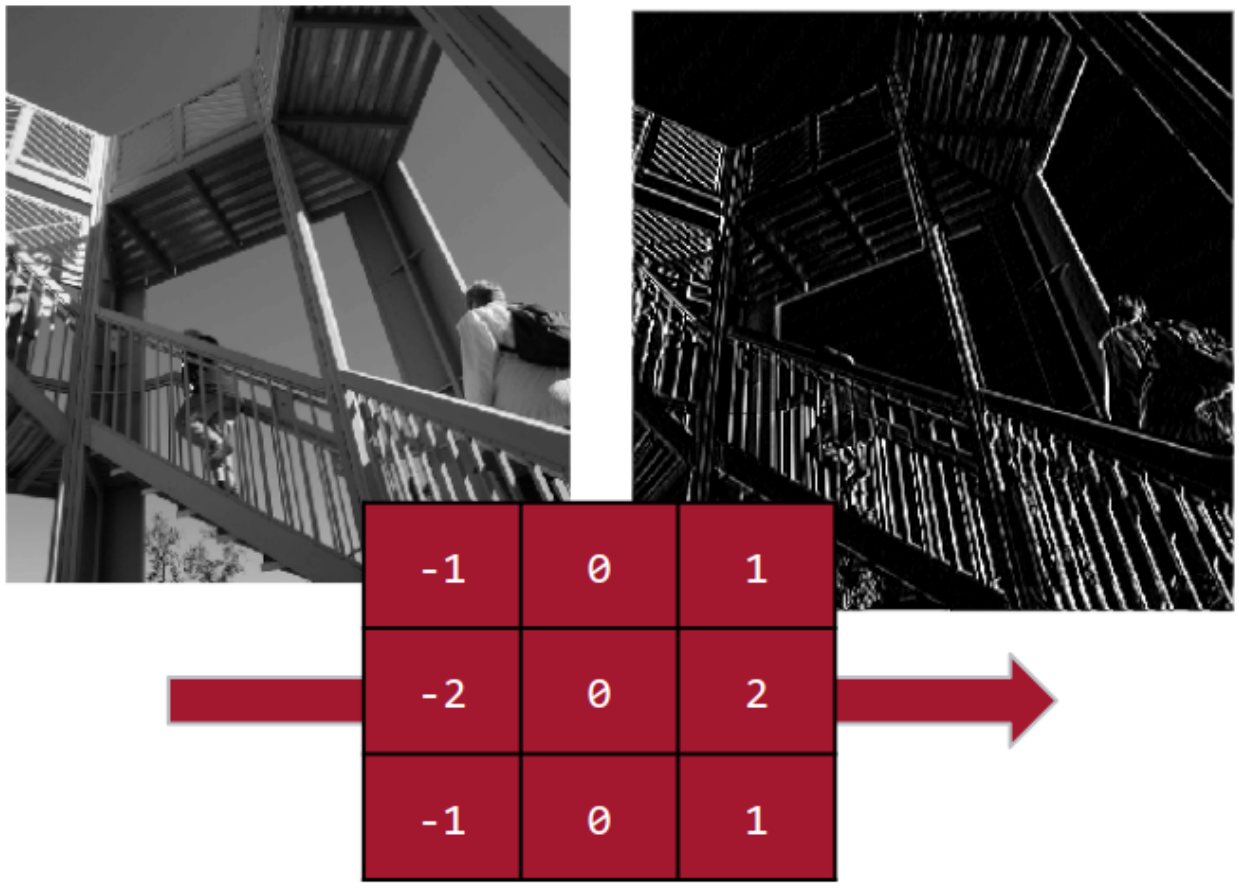
Now that you've seen that MobileNets drastically reduce the size of computer vision neural networks, let's dive a little deeper into how their key innovation works: **Depthwise Separable Convolutions**

.

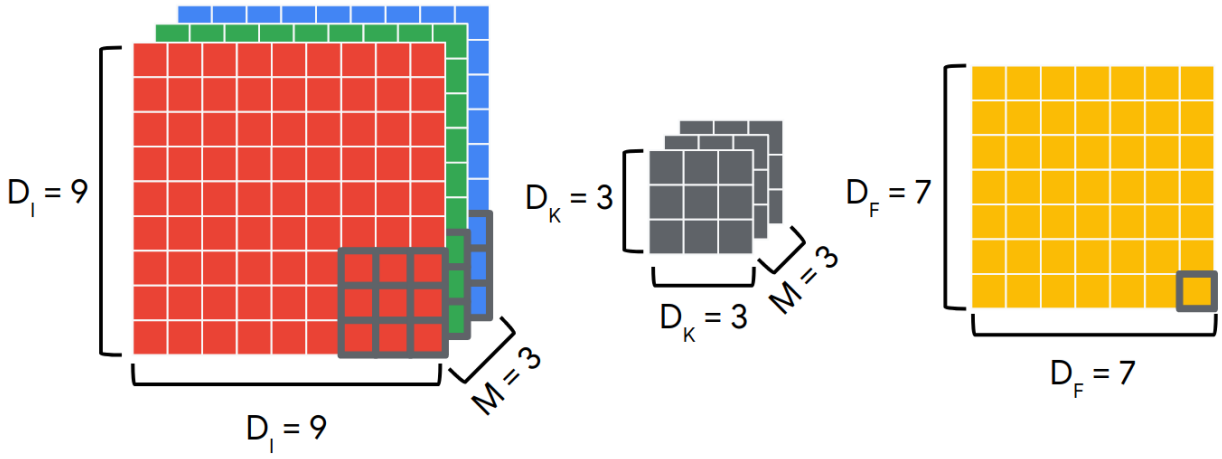
Standard Convolutions

In order to better understand how depth wise separable convolutions are different from standard convolutions let's first re-examine standard convolutions. In particular, we are going to quantify the number of multiplication operations and parameters in a standard convolution.

In the last module, when we considered filters in detail we only looked at filters applied to grayscale images like the vertical line filter shown below:



These images can also be called **single-channel** images as they only have one set of pixel values. In contrast, most color images are **three-channel RGB images**. Where there is a value representing how much red, green, and blue is in the color. As such the filters (also often referred to as **kernels**) used are not simply a matrix but instead a **tensor** as it needs to multiply all three channels at the same time. This tensor operation is shown below. In this example, we have a 3x3x3 kernel being convolved with a 9x9x3 image to produce a 7x7x1 output. The highlighted squares are the inputs and outputs of the last convolution operation.



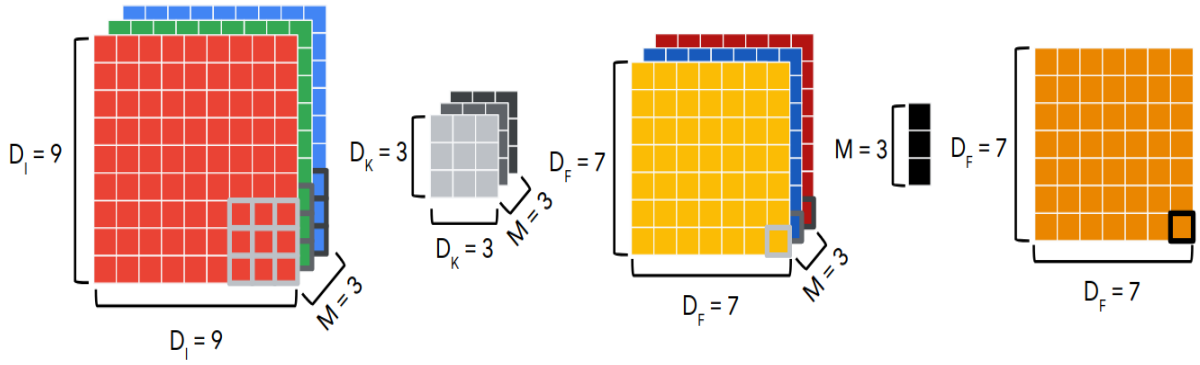
Note that in general, each convolution requires Dk * Dk * M multiplications due to the size of the kernel and the number of channels in the image. Then to produce a single output we need to do these operations Df * Df times to produce the full output. Also in general we don't just use a single kernel, we use N kernels. Multiple kernels are referred to as a filter. A filter is a concatenation of multiple different kernels where each kernel is assigned to a particular channel of the input. As such the total number of multiplications will be:

$$D_K * D_K * M * D_F * D_F * N = D_K^2 * M * D_F^2 * N$$

Depthwise Separable Convolutions

Depthwise Separable Convolutions instead proceed in a two-step process. First, each channel is treated independently as if they were separate single-channel images and filters are applied to them creating multiple outputs. This is referred to as a **Depthwise Convolution**. Next, a **Pointwise Convolution** is applied to those outputs using a 1x1xC filter to compute the final output. This process can be seen below where we again take our 9x9x3 image and apply three separate 3x3 filters to it depthwise to produce a 7x7x3 output.

We then take that output and apply a 1x1x3 pointwise filter to it to produce our final 7x7x1 output.



Note that in general now each depthwise convolution requires M filters of Dk * Dk multiplications and to produce a single output we need to do these operations Df * Df times. Therefore we need Dk * Dk * M * Df * Df multiplications for that stage.

For the pointwise convolution, we now use a 1x1xM filter Df * Df times. In general, just like regular convolutions, we don't use a single filter, we will use multiple filters. During Depthwise Separable Convolutions those multiple filters occur in the pointwise step. Therefore if we had N pointwise filters we will then need M * Df * Df * N total multiplications for this stage.

Summing the number of multiplications we will need in both stages we find that in total we need:

$$D_K * D_K * M * D_F * D_F + M * D_F * D_F * N = M * D_F^2 * (D_K^2 + N)$$

Comparing the two kinds of Convolutions

We can compare the two kinds of convolutions through a ratio of the number of multiplications required for each. Placing standard convolutions on the denominator we get:

$$\frac{\text{Depthwise Separable}}{\text{Standard}} = \frac{M * D_F^2 * (D_K^2 + N)}{D_K^2 * M * D_F^2 * N}$$

$$\frac{\text{Depthwise Separable}}{\text{Standard}} = \frac{D_K^2 + N}{D_K^2 * N}$$

$$\frac{\text{Depthwise Separable}}{\text{Standard}} = \frac{1}{N} + \frac{1}{D_K^2}$$

This means that the more filters we use and the larger the kernels are, the more multiplications we can save. If we use our example from above where D_K=3 and we conservatively use only N=10 filters we will find that the ratio becomes 0.2111 meaning that by using Depthwise Separable Convolutions we save almost 5x the number of multiplication operations! This is far more efficient and can greatly improve latency.

Also, note that in the case of standard convolution we have Dk^2 * M * N learnable parameters in our various filters/kernels. In contrast, in the Depthwise Separable case, we have Dk^2 * M + M * N. Again if we take the ratio of the two we find that:

$$\frac{\text{Depthwise Separable}}{\text{Standard}} = \frac{1}{N} + \frac{1}{D_K^2}$$

This means that we also have a much smaller memory requirement as we have far fewer parameters to store!

There is a tradeoff however, in improving our latency and memory needs we have reduced the number of parameters that we can use to learn with. Thus our models are more limited in their expressiveness. This is usually sufficient for TinyML applications but is something to consider when using Depth Wise Separable Convolutions in general!

Finally, if you'd like to read more detail about MobileNets you can check out the paper describing them here.