Project Final Report

Benchmarking Inference Neural Networks on mobile devices

Student: **Sobirdzhon Bobiev**
Supervisor: **Adil Mehmood Khan**

In this project we wanted to analyse performance of individual layers in convolutional neural networks. The affect of separable convolutions and different kernel sizes on the inference speed were in our interest. In the following two sections we will explain each of them in details.

Separable Convolutions

The different *flavours* of convolutional layers was the center of our focus. Here are the four types of convolutions that we incorporated in our models:

1. *Normal convolution* - a normal 2D convolution
2. *Spatially separable* - conceptually, it tries to factorize the kernel of size (k, k, d) into (k, 1, d) followed by (1, k, 1) kernel where d is the depth and k is the width and height of the square kernel.
3. *Depthwise separable* - a depthwise convolution followed by pointwise convolution. This is the type of convolution that was popularized by MobileNet.
4. *Depthwise and spatially separable* - this combines the ideas of the two above convolution types. This consists of three parts:
    1) Depthwise convolution with kernel size (k, 1)
    2) Depthwise convolution with kernel size (1, k)
    3) Pointwise convolution

We want to mention that the 2nd and 4th type are not commonly used.

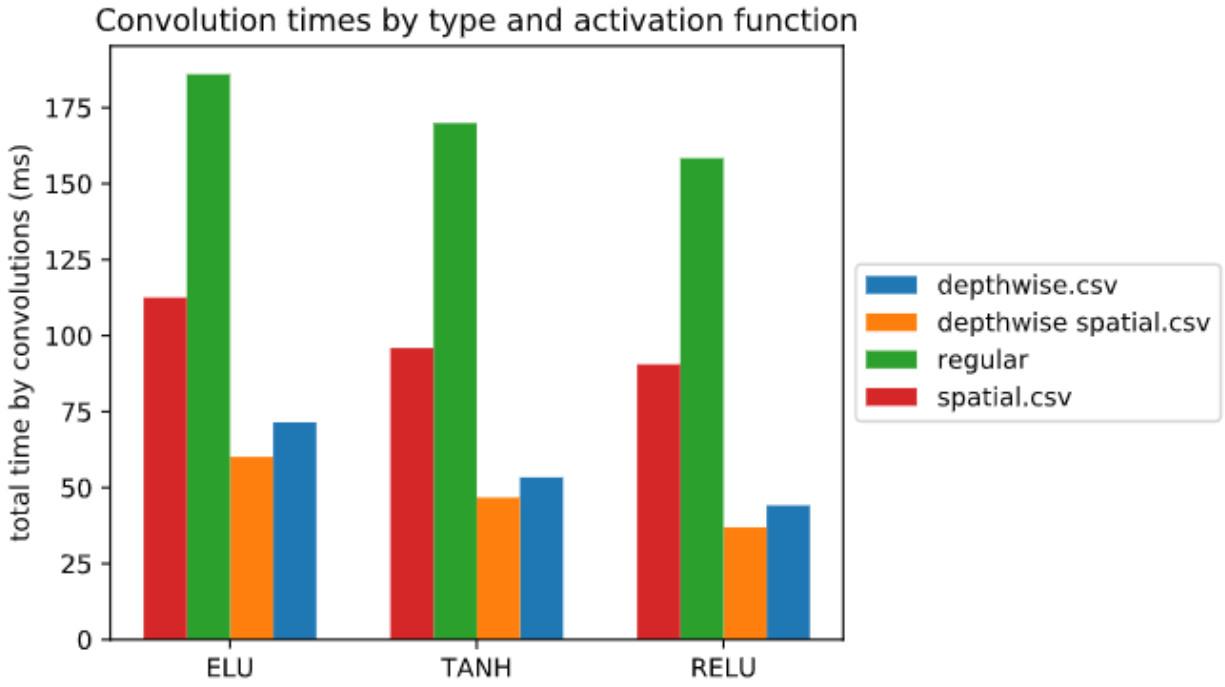
Together with three activation functions, ELU, RELU, TANH, there are 3x4=12 possible models in our set.

The table below presents the amount calculations done and number of parameters in each of the convolution types:

Conv type	FLOPs (# of multiplications)	Number of parameters
Normal convolution	$f \cdot f \cdot d_{in} \cdot H_{out} \cdot W_{out} \cdot d_{out}$	$f \cdot f \cdot d_{in} \cdot d_{out}$
Spatially separable	$f \cdot d_{in} \cdot H_{out} \cdot (W_{out} + f - 1) \cdot d_{out}$ + $f \cdot H_{out} \cdot W_{out} \cdot d_{out}$	$f \cdot d_{in} \cdot d_{out} + f \cdot d_{out}$
Depthwise separable	$f \cdot f \cdot d_{in} H_{out} W_{out} + d_{in} H_{out} W_{out} d_{out}$	$f \cdot f \cdot d_{in} + d_{in} \cdot d_{out}$

Depthwise and spatially separable	$f \cdot H_{out} \cdot (W_{out} + 2) + f \cdot H_{out} \cdot W_{out} + d_{in} H_{out} W_{out} d_{out}$	$2 \cdot f \cdot d_{in} + d_{in} \cdot d_{out}$

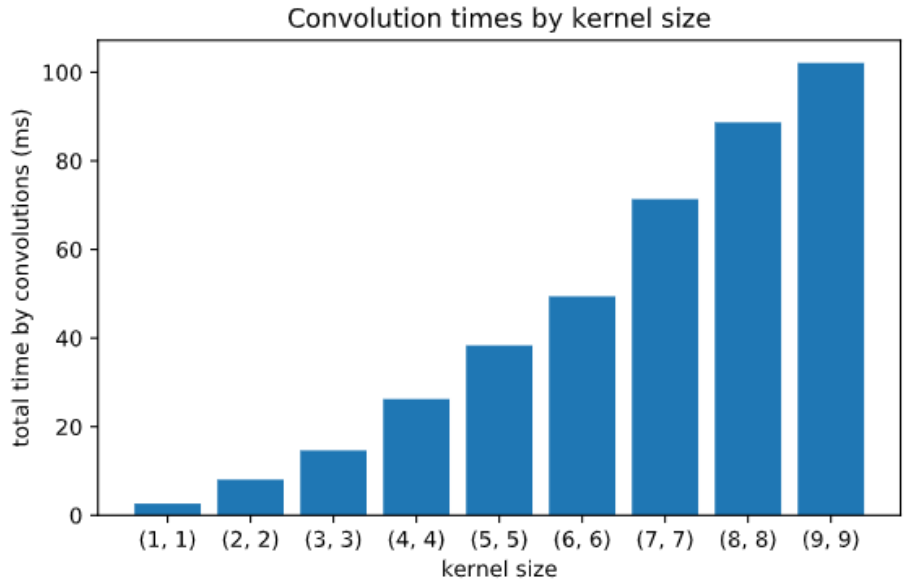
The plot below presents benchmarking results for all these 12 models which vary in activation functions and convolutions types. We can see that using separable convolutions we can gain speed from 1.5 up to 3 times than using normal convolutions.



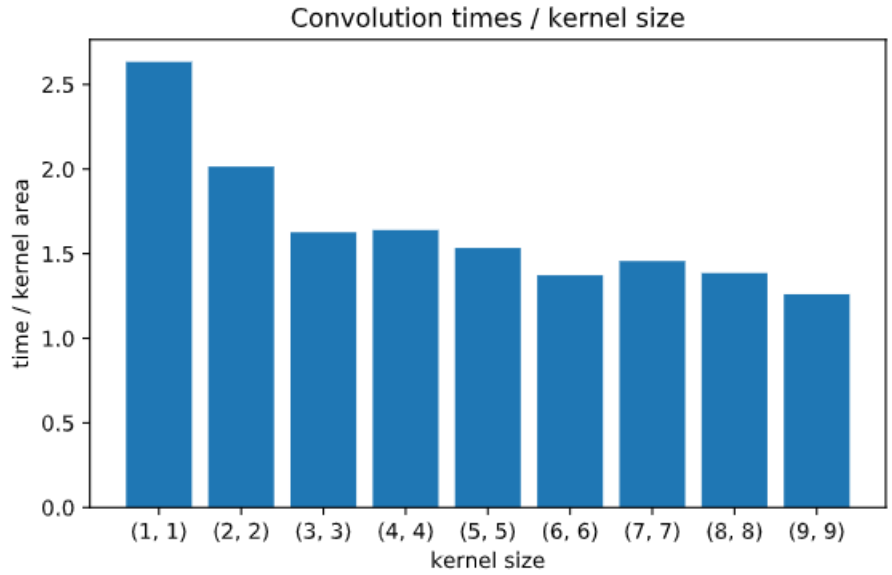
Different kernel sizes

Another set of models that we created are same except the kernel sizes in each convolution layers. We considered kernel sizes from 1x1 up to 9x9 (squares). Our benchmarking results

show that speed linearly correlates with the kernel area.



When we normalize by kernel area we get the following plot



Conclusion

Optimizing neural network architectures for mobile platforms is possible in many ways. Since convolutional layers are dominant in many models we tried to experiment with 4 variations of them and concluded that using separable convolutions we can gain speed up to 3 times than using normal convolutions.

Our experiments with kernel sizes show that computation time in convolution layers roughly depend linearly on the kernel area, (thus depends on the number of operations), for large enough kernel sizes (>=3), which is an expected result.