Detail-Preserving Pooling in Deep Networks
– Supplemental Material –

Faraz Saeedan[1] Nicolas Weber[1,2*] Michael Goesele[1,3*] Stefan Roth[1]
[1]TU Darmstadt [2]NEC Laboratories Europe [3]Oculus Research

A. Learned Pooling Parameters

The learned pooling parameters for a VGG network with Lite-DPP$_{\text{Asym}}$ after training on CIFAR10 have been shown in Fig. 6 in the main manuscript. Here, Fig. 7 additionally shows the learned pooling parameters for a VGG network using the *symmetric* Lite-DPP$_{\text{Sym}}$, trained on the same dataset. It can be seen that the general trend of the pooling parameters resembles that of the asymmetric case. The earlier pooling layers have a smaller reward bias, which indicates an extremum-like tendency, while the deeper layers learn larger bias values and hence behave more similar to average pooling. Compared to the asymmetric case, here the reward exponent learned for the early layers is smaller, which indicates that the network seems to prefer a somewhat smaller level of detail preservation with extremum pooling compared to max pooling in the asymmetric case.

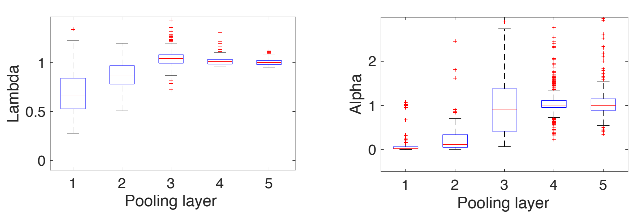

Figure 7. Illustration of the learned λ (*left*) and α (*right*) for different pooling layers in a VGG network with Lite-DPP$_{\text{Sym}}$ on CIFAR10. Higher values of λ indicate a more extremum-like behavior, while higher values of α indicate an averaging tendency.

B. Network Details

For our experiments on CIFAR10, we have used a VGG-like and a NIN-like network (aside from RestNet-110 and DenseNet-BC). The exact architecture of the former two networks is given in Table 5. Every convolution layer is followed by a batch normalization and a ReLU nonlinearity. The NIN-like network has a fully connected layer after the global average pooling, as we found that adding this layer helps avoiding the otherwise fairly common problem of local minima.

To apply different pooling methods we have simply swapped the pooling layers in the original networks of Table 5 with the desired pooling layers. To use strided convolutions instead, we have removed the pooling layers and instead set the stride in the convolution layers before the original pooling locations to two.

VGG configuration	**NIN configuration**
conv3-64 $\times 2$ **pooling/2** conv3-128 $\times 2$	conv5-192 $\times 1$ conv1-160 $\times 1$ conv1-96 $\times 1$
pooling/2	**pooling/2**
conv3-256 $\times 3$ **pooling/2** conv3-512 $\times 3$	conv5-192 $\times 1$ conv1-192 $\times 2$
pooling/2	**pooling/2**
conv3-512 $\times 3$ **pooling/2** FC-512	conv3-192 $\times 1$ conv1-192 $\times 2$ avg pooling/8
FC-10	FC-10

Table 5. The configurations of two of the networks used for the CIFAR10 experiments. Each convolutional layer is followed by batch normalization and a ReLU unit (not shown). The layers in bold have been replaced with DPP.

*This work was carried out while at TU Darmstadt.