Supplemental Material: CoordGate

This document acts as the supplemental material for the paper *CoordGate: Efficiently Computing Spatially-Varying Convolutions in Convolutional Neural Networks.*

Additional Results

This section provides some additional results and metrics that it was not possible to include in the main text due to space limitations. All these results concern experiment 4.2 in the main text.

Training Curves

Training curves for the results displayed in Fig. 4b. Each model was trained with the Adam optimizer with default initial parameters, and the learning rate was set to half if the validation loss did not decrease for 10 epochs.

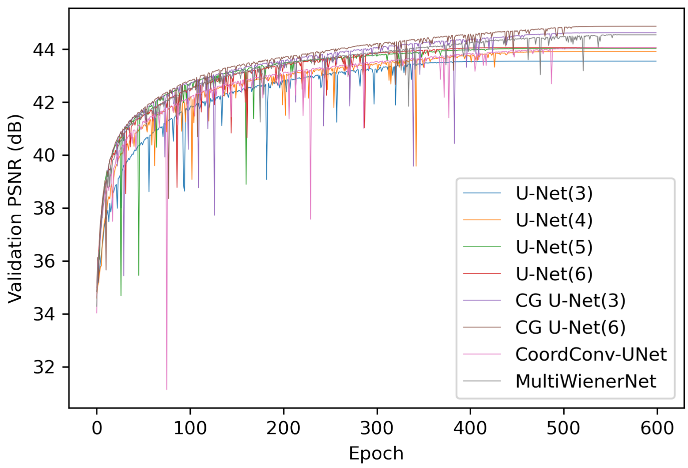

Figure 1: Curves showing the PSNR evaluated on the validation dataset during training, for each model.

SSIM Results

The structural similarity index measure (SSIM) for each trained model, evaluated on the test set. We see that these results follow the trend of the PSNR.

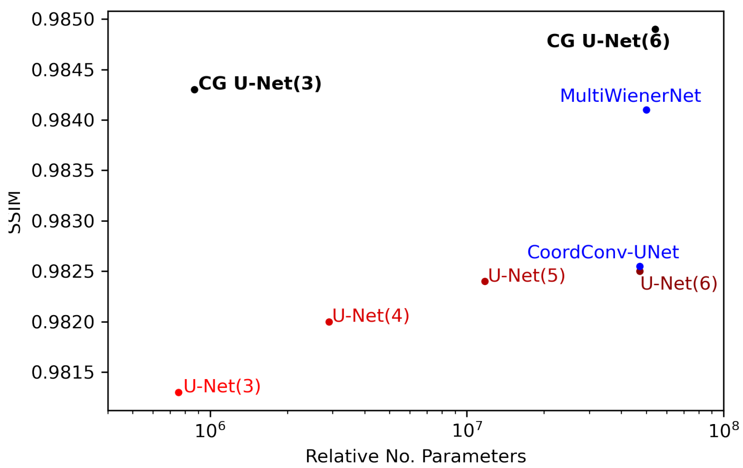

Figure 2: The SSIM of each trained models predictions, evaluated on the test dataset.

Equivalency to Locally Connected Layer

In the situation that the kernels within the convolutional layer form a complete basis, the CoordGate module is equally expressive as a locally-connected layer. To see this, let's take a simple example case of a 3x3 convolution, where a complete basis is for instance formed by 9 "pixel-basis" filters of the form:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \cdots, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{1}$$

CoordGate encodes position into a gating map, to amplify or dampen the individual filters at each position of the convolutional feature map. For instance, at two adjacent spatial positions, the gating map tensor could have the values [1,2,1,1,1,1,1,1,1] and [0,2,1,1,1,1,1,1,1]. After the Hadamard product is taken between the gating map and the feature map, summing over the channel dimension will result in combined filters:

$$\begin{bmatrix} 1 & 2 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 2 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}. \tag{2}$$

So we have broken the weight sharing inherent to convolutional layers and instead have individual filters for the receptive field of each neuron. This is exactly what a locally-connected neural network does. If we were to add an additional gating map this operation would be identical to a locally-connected layer; without it all neurons have a shared bias term.