Efficient Supervised Sparse Analysis and Synthesis Operators
Supplementary Materials

Pablo Sprechmann
Duke University
pablo.sprechmann@duke.edu

Roee Litman
Tel Aviv University
roeelitman@post.tau.ac.il

Tal Ben Yakar
Tel Aviv University
talby10@gmail.com

Alex Bronstein
Tel Aviv University
bron@eng.tau.ac.il

Guillermo Sapiro
Duke University
guillermo.sapiro@duke.edu

input : Sub-gradient $\delta\mathbf{y}$ of ℓ with respect to network output; intermediate layer outputs $\{\mathbf{z}^k, \mathbf{b}^k\}$
output: Sub-gradients of ℓ with respect to the parameters, $\delta\mathbf{A}$, $\delta\mathbf{H}$, $\delta\mathbf{G}$, $\delta\mathbf{F}$, $\delta\mathbf{U}$, $\delta\mathbf{V}$, and $\delta\mathbf{t}$
Initialize $\delta\mathbf{U}^{K+1} = \delta\mathbf{y}\mathbf{x}^{\mathrm{T}}$, $\delta\mathbf{V}^{K+1} = \delta\mathbf{y}(2\mathbf{z}^{K+1} - \mathbf{b}^{K+1})^{\mathrm{T}}$, $\delta\mathbf{A} = \mathbf{0}$,
$\delta\mathbf{F}^{K+1} = \delta\mathbf{G}^{K+1} = \delta\mathbf{H}^{K+1} = \mathbf{0}$, $\delta\mathbf{t}^{K+1} = \mathbf{0}$, $\delta\mathbf{b}^{K+1} = -\mathbf{V}^{\mathrm{T}}\delta\mathbf{y}$, $\delta\mathbf{b}^{K+2} = 0$, and
$\delta\mathbf{z}^{K+1} = 2\mathbf{V}^{\mathrm{T}}\delta\mathbf{y}$.
for $k = K, K-1, \dots, 1$ **do**

$$\begin{aligned}
\delta\mathbf{s} &= \delta\mathbf{z}^{k+1} + \mathbf{H}^{\mathrm{T}}\delta\mathbf{b}^{k+1} \\
\delta\mathbf{F}^k &= \delta\mathbf{F}^{k+1} + \delta\mathbf{b}^{k+1}(\mathbf{b}^k)^{\mathrm{T}} \\
\delta\mathbf{G}^k &= \delta\mathbf{G}^{k+1} + \delta\mathbf{b}^{k+1}(\mathbf{b}^{k-1})^{\mathrm{T}} \\
\delta\mathbf{H}^k &= \delta\mathbf{H}^{k+1} + \delta\mathbf{b}^{k+1}(\mathbf{z}^{k+1} - \mathbf{z}^k)^{\mathrm{T}} \\
\delta\mathbf{t}^k &= \delta\mathbf{t}^{k+1} + \delta\mathbf{s} \odot \frac{\partial}{\partial\mathbf{t}}\boldsymbol{\sigma}_{\mathbf{t}}(\mathbf{b}^k) \\
\delta\mathbf{z}^k &= -\mathbf{H}^{\mathrm{T}}\delta\mathbf{b}^{k+1} \\
\delta\mathbf{b}^k &= \mathbf{G}^{\mathrm{T}}\delta\mathbf{b}^{k+1} + \mathbf{F}^{\mathrm{T}}\delta\mathbf{b}^{k+2} + \delta\mathbf{s} \odot \frac{\partial}{\partial\mathbf{b}^k}\boldsymbol{\sigma}_{\mathbf{t}}(\mathbf{b}^k)
\end{aligned}$$

end
$\delta\mathbf{A} = \delta\mathbf{b}^1\mathbf{x}^{\mathrm{T}}$
Output $\delta\mathbf{A}$, $\delta\mathbf{F}^1$, $\delta\mathbf{G}^1$, $\delta\mathbf{H}^1$, $\delta\mathbf{U}^1$, $\delta\mathbf{V}^1$, and $\delta\mathbf{t}^1$.

Algorithm 2: Backpropagation process for the computation of the sub-gradients of $\ell(\mathbf{y})$. $\delta*$ denotes the gradient of ℓ with respect to $*$ as customary in neural network literature. $\odot$ denotes element-wise product.

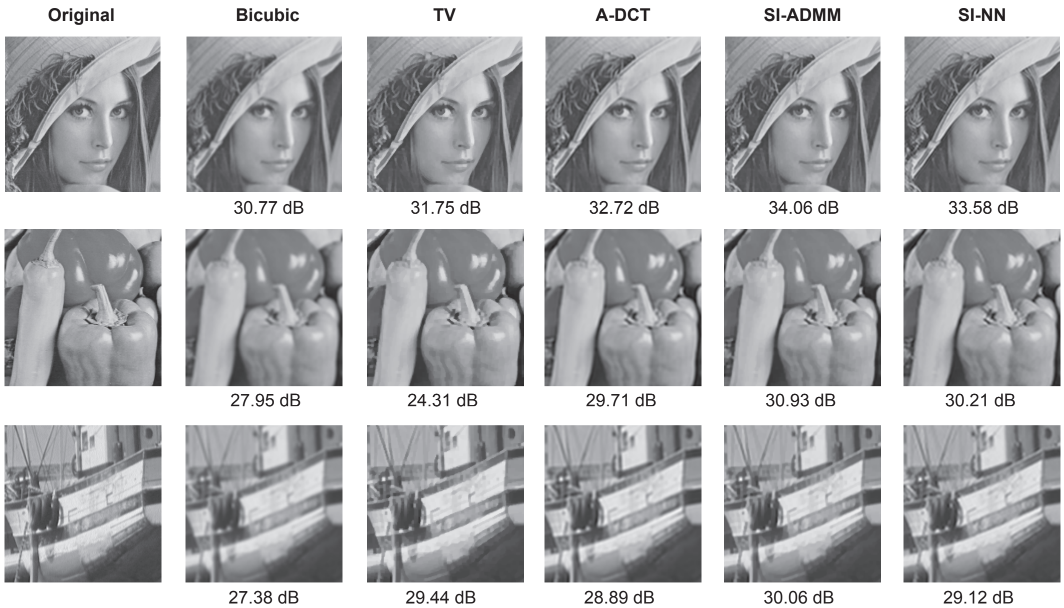

Figure 3: Outputs of different image super-resolution methods (left-to-right): original image, bicubic interpolation, shift-invariant analysis models with TV and DCT priors, supervised shift-invariant analysis model, and its fast neural network approximation. PSNR is reported in dB below each image.

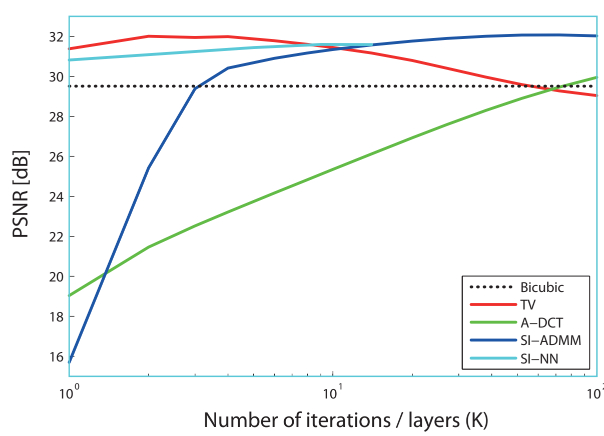

Figure 4: Performance of different analysis super-resolution models as the function of number of ADMM iterations, compared to the performance of the convolutional neural network approximation as the function of number of layers. Compared are shift-invariant analysis models with TV and DCT priors (TV and A-DCT), supervised shift-invariant analysis model (SI-ADMM), and its fast neural network approximation (SI-NN). Bicubic interpolation (Bicubic) is given as a reference. Note the typical non-monotone behavior of the total variation model.