LINN: Lifting Inspired Invertible Neural Network for Image Denoising

Junjie Huang and Pier Luigi Dragotti
Electrical and Electronic Engineering Department, Imperial College London
Background — Image Denoising

- The objective is to recover a clean image from the observed noisy image:
Background — Image Denoising

Model-based methods

• Based on well-defined image priors or noise statistics
• White-box systems with good interpretability and strong generalization ability

Learning-based methods

• Learning from noisy-clean image pairs
• Black-box system and restricted generalization ability
Motivation and Idea

- **Motivation**: whether it is possible to learn a non-linear wavelet transform for image denoising and other image restoration tasks?

- **Idea**: propose an image denoising invertible neural network based on the principle of transform-based denoising
  - ✓ A lifting inspired invertible neural network
  - ✓ Sparsity-driven denoising network
DnINN: Image Denoising Invertible Neural Network

Noisy image $\Leftrightarrow$ Denoising Network $\Leftrightarrow$ Denoised image

$\text{DnINN: Image Denoising Invertible Neural Network}$
Lifting Scheme

Splitting and merging operator
- Split \( x \) into odd part \( x_o \) and even part \( x_e \)
- Combine \( x_o \) and \( x_e \) into \( x \)

Predictor and updater
- A predictor is used to predict \( x_o \) from \( x_e \)
- The updater adjusts \( x_e \) based on the prediction error of \( x_o \)
Lifting inspired Invertible Neural Network

- Forward pass

Diagram: Image to representation, showing a flow from input image to representation through Predictor and Updater modules, with separate channels for Detail part and Coarse part.
Lifting inspired Invertible Neural Network

- **Forward pass**

  - Image to representation

- **Backward pass**

  - Representation to image

When no operation is applied on the representation, perfect reconstruction can be achieved using the backward pass.
LINN — Splitting/Merging Operator

- **Forward pass**
  - The splitting operator is the **Undecimated Haar Wavelet Transform**

- **Backward pass**
  - The merging operator is the **Inverse Undecimated Haar Wavelet Transform**
LINN — Predictor/Updater Networks

- **Forward pass**
- **Backward pass**
LINN — Predictor/Updater Networks

• Forward pass
  – There are $I$ pairs of P-Net and U-Net to sequentially update the detail and the coarse part

\[
\begin{align*}
  Z_d^{(i)} &= Z_d^{(i-1)} - P_i \left( Z_c^{(i-1)} \right) \\
  Z_c^{(i)} &= Z_c^{(i-1)} + U_i \left( Z_d^{(i)} \right)
\end{align*}
\]

• Backward pass
  – The same $I$ pairs of P-Net and U-Net are used for reconstruction

\[
\begin{align*}
  Z_c^{(i-1)} &= Z_c^{(i)} - U_i \left( Z_d^{(i)} \right) \\
  Z_d^{(i-1)} &= Z_d^{(i)} + P_i \left( Z_c^{(i-1)} \right)
\end{align*}
\]
LINN — Predictor/Updater Networks

- Convolutional networks with soft-thresholding non-linearity

\[ S_\lambda (x) \]

\( \lambda \) soft-threshold
Denoising Network

• Non-invertible component
• The denoising network enforces the detail part to be sparse
• A well-understood denoising network can lead to enhanced interpretability
Denoising Network

- $l_1$-norm minimization problem:

$$g = \arg\min_g \frac{1}{2\sigma^2} \| z_d^l - g \|^2_2 + \lambda \| g \|_1$$

- Closed-form solution:

$$g = S_{\sigma^2\lambda}(z_d^l)$$

Noise adaptive soft-threshold
Denoising Network

• Over-parameterized $l_1$-norm minimization problem:

$$g = \arg\min_g \frac{1}{2\sigma^2} \|z_d^I - D \ast g\|_2^2 + \lambda \|g\|_1$$

  – Learned Iterative Shrinkage Thresholding Algorithm (ISTA):

$$g_{t+1} = S_{\lambda/\mu} \left( \left( I - \frac{1}{\mu} D^T \ast D \right) \ast g_t + \frac{1}{\mu} D^T \ast z_d^I \right)$$
Simulation Results

- Training loss:
  - Mean squared error between restored image and clean image
- Optimizer:
  - ADAM with learning rate $1 \times 10^{-3}$
- Training data:
  - BSD dataset: 400 images of size $180 \times 180$
| Methods        | Model Size     | $\sigma_N = 15$ | $\sigma_N = 25$ | $\sigma_N = 50$ |
|---------------|---------------|-----------------|-----------------|-----------------|
| BM3D [9]      | -             | 31.07           | 28.57           | 25.63           |
| WNNM [10]     | -             | 31.37           | 28.83           | 25.87           |
| EPLL [24]     | -             | 31.21           | 28.68           | 25.67           |
| TNRD [12]     | $26.6 \times 10^3$ | 31.42           | 28.92           | 25.97           |
| DnCNN [13]    | $556.0 \times 10^3$ | 31.70           | 29.19           | 26.20           |
| DnINN$_{ST}$  | $134.7 \times 10^3$ | 31.58           | 29.08           | 26.14           |
| DnINN$_{LISTA}$ | $135.2 \times 10^3$ | 31.59           | 29.09           | 26.14           |
| DnINN$_{ST}$ (2-scale) | $269.3 \times 10^3$ | 31.62           | 29.14           | 26.19           |
| DnINN$_{LISTA}$ (2-scale) | $270.3 \times 10^3$ | 31.63           | 29.14           | 26.20           |

**TABLE 1**

The model size and PSNR (dB) results of different methods on BSD68 dataset on noise level $\sigma_N = 15, 25, 50$. 
Simulation Results

Detail channels output from the lifting scheme neural network

LISTA-Net

Detail channels after denoising
Simulation Results

![Graph showing PSNR (dB) vs Testing Noise Level for DnINN-50 and DnCNN-50 models, with a green vertical line indicating the training noise level.](image-url)
Conclusions

• We proposed a image denoising invertible neural network (DnINN) method based on the principles of transform-based denoising
  – LINN implements the non-linear transform with perfect reconstruction capability
  – Simple denoising networks can remove the noise in the transform coefficients

• Simulation results show that DnINN method achieves comparable results as the DnCNN method while using $\frac{1}{4}$ learnable parameters