Image dequantization for hyperspectral lossy compression with convolutional neural networks

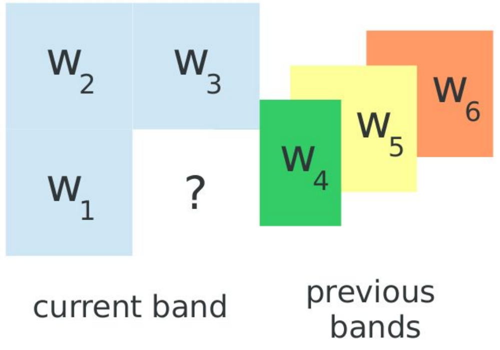
Diego Valsesia
Enrico Magli

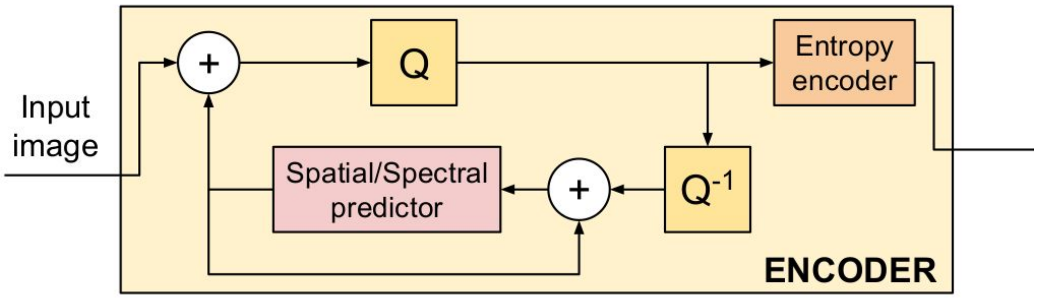
European Workshop on On-Board Data Processing (OBDP2019)
February 26th 2019

POLITECNICO DI TORINO

CCSDS 123.0-B-2

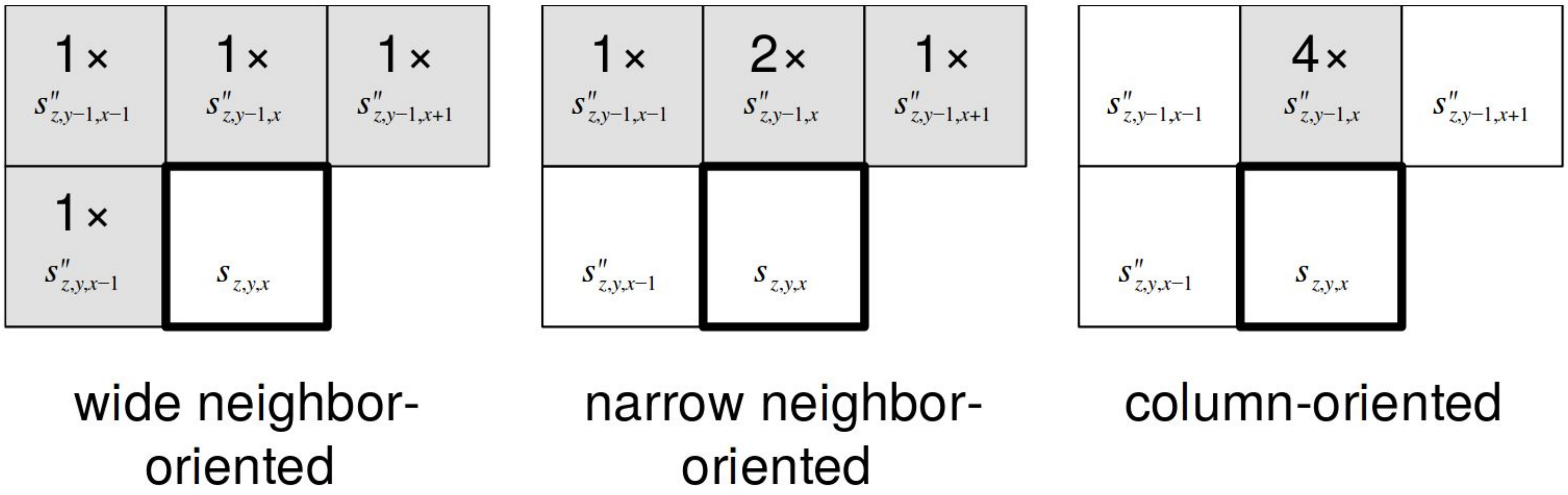
- New (February 2019) CCSDS standard for **onboard lossy compression** of hyperspectral images
- Extends previous lossless standard
- **Predictive coding** approach:
  - Predict pixel value from a causal neighborhood
  - Quantize prediction residual
  - Entropy encode quantized residual





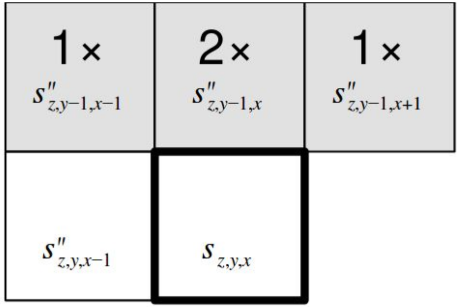
CCSDS 123.0-B-2

- Many prediction modes depending on how to use the neighboring pixels
- In particular:
  - Wide neighbor-oriented local sums: best compression ratios
  - Narrow neighbor-oriented local sums: avoids pixel on the left
  - Column-oriented local sums: only pixel on top is used (e.g., for striped images)



Why narrow sums? The throughput problem

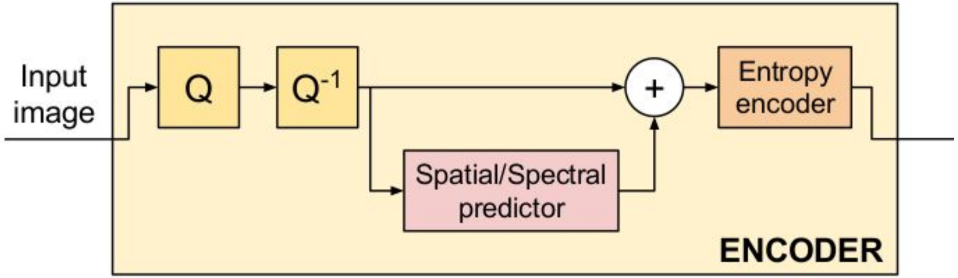
- Narrow sums avoid the pixel on the left
- The **quantizer is inside** the prediction loop
- Before predicting the new pixel, all the pixels in the neighborhood must be **reconstructed** from the quantized residuals
  - **Data dependency!**
- The **pixel on the left** is coded immediately before



- **Waiting** for it causes a **bottleneck** in throughput (no parallelization is possible)
- This issue is not present in **lossless** mode (no reconstruction needed):

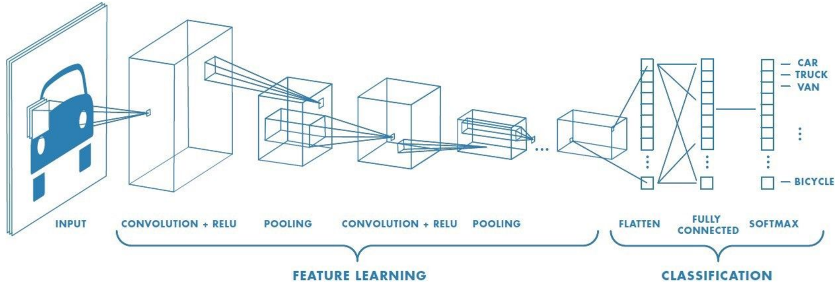
~4x faster than lossy

So, do we really need the in-loop quantizer?



- Alternative solution:
  - **Prequantize** all the raw pixels
  - Run a **lossless** predictor
  - Entropy code the residuals
- Lossless predictor solves the throughput problem
- Known to be **suboptimal** in terms of rate-distortion performance (especially at low bitrates)
- Can we recover this suboptimality with some clever processing **on-ground**?

Convolutional Neural Networks

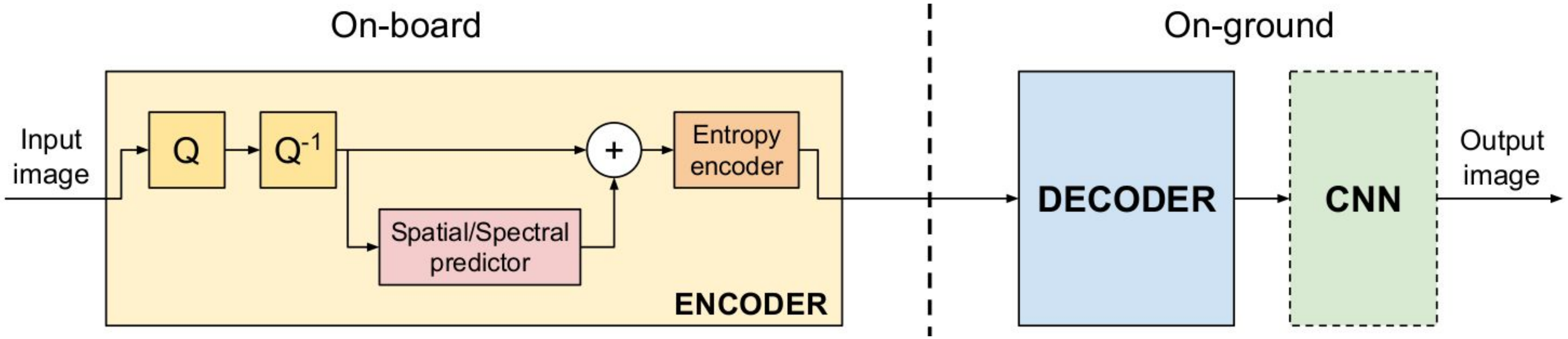


- **Convolutional Neural Networks** (CNNs) are made of layers with a filter bank and a nonlinear activation function

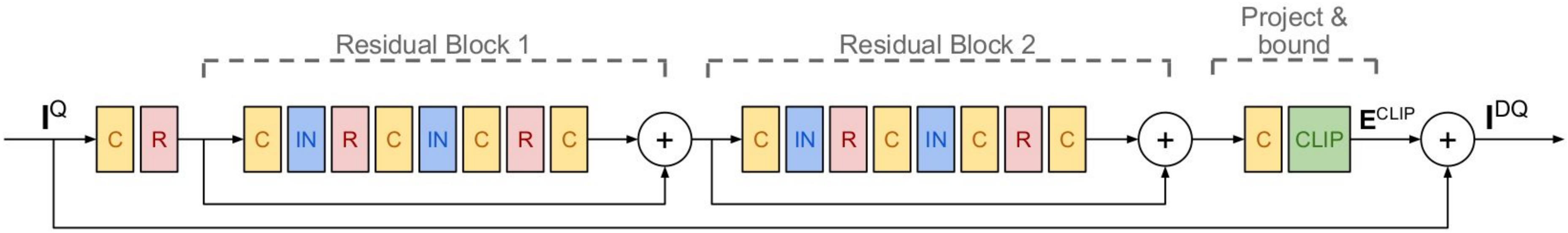
- CNNs are **state of the art** for imaging problems such as: classification, object detection, segmentation, …
- **Inverse problems** such as denosising, superresolution, restoration can also be solved by CNNs better than model-based approaches
- Why so powerful? CNNs create **nonlinear hierarchies of local features** that create a much more sophisticated prior for natural images than anything hand-crafted

Image dequantization with CNNs

- Goal: a CNN to **remove the artifacts** introduced by the lossy compression process for hyperspectral images
- To be applied at the **ground segment**, where computational resource are abundant
- CNN post-processing boosts rate-distortion performance

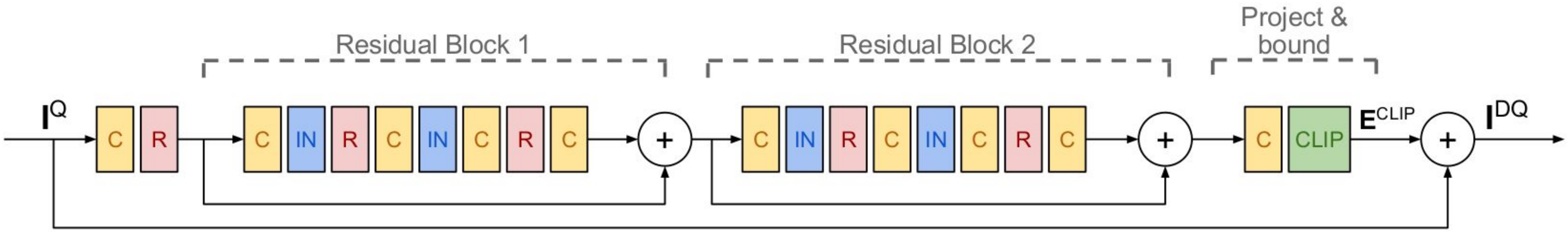


A residual CNN hyperspectral dequantizer



- CNN **estimates the distortion** introduced by lossy compression (global input-output residual): easier than directly cleaning up the image
- Inner **residual** blocks allow a deeper, more powerful network
- **Instance normalization** layers provide "contrast normalization" to keep outputs well normalized and learn faster
- Input size: $N_L \times N_C \times 8$ (8 bands at a time are considered)
  - Sliding a window over spectral dimension allows to process images with more bands

A residual CNN hyperspectral dequantizer



- CLIP operation ensures **consistent reconstruction**
  - The restored pixel values will fall in the same original quantization bins

$$E^{\text{CLIP}}_{x,y,z} = \begin{cases} -\Delta & \text{if } E_{x,y,z} \leq -\Delta \\ \Delta & \text{if } E_{x,y,z} \geq \Delta \\ E_{x,y,z} & \text{otherwise} \end{cases}$$

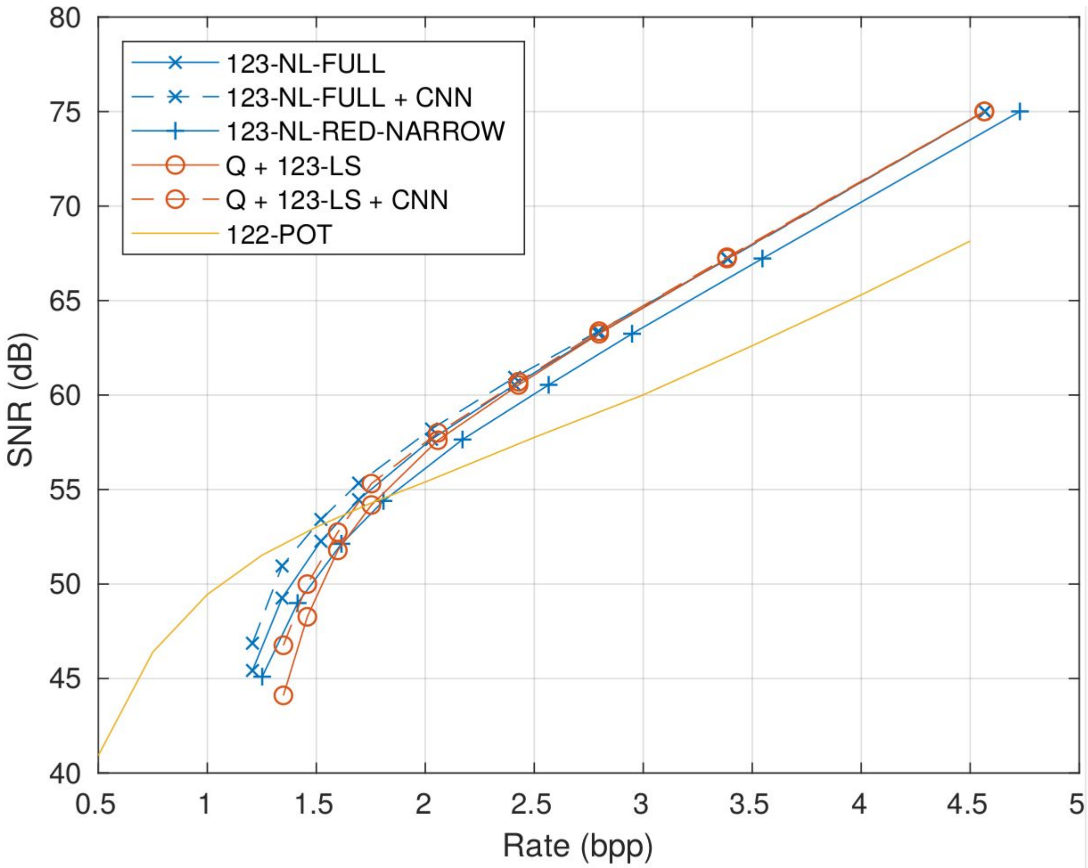
- Training finds the weight values that minimize the **MSE** between the reconstructed image and the original

CNN training details

- Optimal performance requires a model trained for:
  - a target **sensor** (e.g., the AVIRIS instrument)
  - a target **compression algorithm** (e.g., prequantization + lossless CCSDS 123.0-B-2)
  - a target **quality point** (e.g., a fixed quantization step size)
- Some features learned by the network are **general** and changing one target requires only **finetuning** the trained model (with a small amount of data)

- **Training data**: 70000 patches of size 32 x 32 x 8 randomly extracted from decoded images for the Cuprite, Jasper and Moffett scenes acquired by AVIRIS
- **Test data**: Yellowstone scenes (unseen during training, size: 512 x 680 x 224)

Results



123-NL-FULL: CCSDS-123.0-B-2 lossy, full, wide neighbor-oriented mode

123-NL-RED-NARROW: CCSDS-123.0-B-2 lossy, reduced, narrow neighbor-oriented mode

Q + 123-LS: prequantization + lossless CCSDS-123.0-B-2 full, wide neighbor-oriented

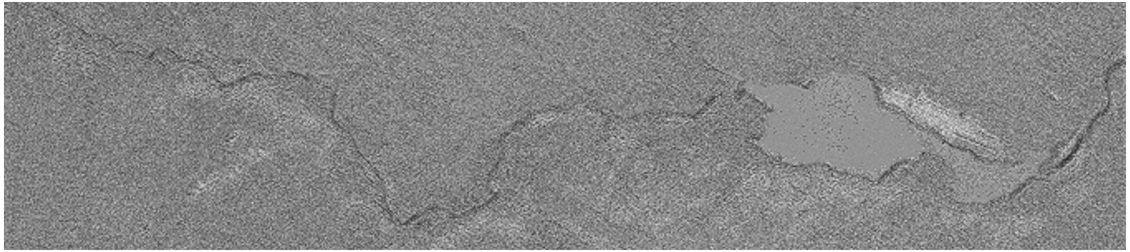
- **Small suboptimality** of Q+123-LS with respect to 123-NL-FULL
- **Entirely recovered** by the CNN at 2 bpp
- **Always better** than 123-NL-RED-NARROW even without CNN

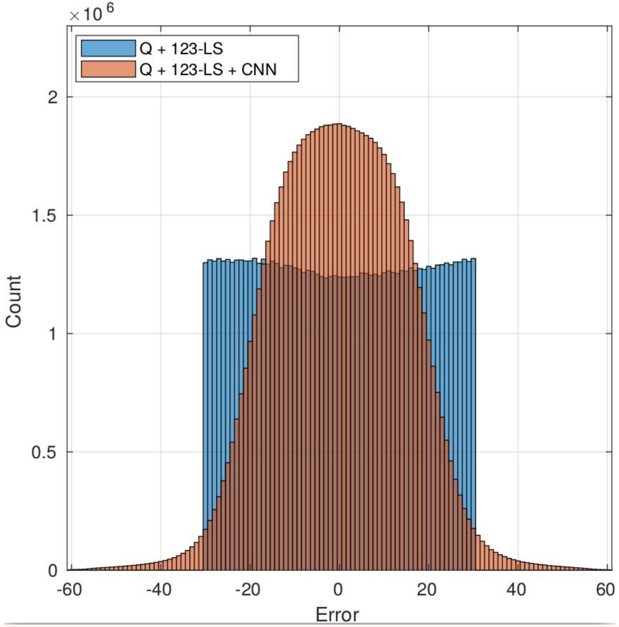
Results

Mean SNR (dB) for 5 test images

	123-NL	123-NL + CNN	Q + 123-LS	Q + 123-LS + CNN	123-NL-RED-NARROW	122-POT
1.5 bpp	52.23 ± 0.49	$\mathbf{53.38} \pm 0.50$	49.58 ± 0.69	51.10 ± 0.73	50.85 ± 0.64	53.13 ± 0.54
2.0 bpp	57.60 ± 0.34	$\mathbf{58.19} \pm 0.36$	57.15 ± 0.33	57.65 ± 0.36	56.36 ± 0.33	55.53 ± 0.31
3.0 bpp	64.88 ± 0.34	$\mathbf{64.93} \pm 0.33$	64.80 ± 0.34	64.92 ± 0.35	63.81 ± 0.32	60.09 ± 0.28
4.0 bpp	71.57 ± 0.36	$\mathbf{71.55} \pm 0.36$	71.53 ± 0.37	71.56 ± 0.36	70.49 ± 0.34	65.33 ± 0.36

Correction estimated by the CNN at Q=61
(AVIRIS Yellowstone sc0, band 47, lines 150-300)





Conclusions

- Pixel prequantization + lossless compression with CCSDS-123.0-B-2 + onground CNN has both strong **rate-distortion** performance and **speed**
- **Equal compression** performance to the **best prediction mode** of the new lossy standard but **faster**
- **Always better** than the **high-throughput mode** of the standard

- CNN can largely **recover the suboptimality** of out-of-loop prequantization, even at medium-low rates
- Carefully consider this scheme when speed is an issue
- When speed is not an issue, CNN can still provide significant gains to CCSDS-123.0-B-2