POLITECNICO DI TORINO
Repository ISTITUZIONALE

Scalable Neural Pushbroom Architectures for Real-Time Denoising of Hyperspectral Images Onboard Satellites

Original
Scalable Neural Pushbroom Architectures for Real-Time Denoising of Hyperspectral Images Onboard Satellites / Yi, Z., Piccinini, D., Valsesia, D., Bianchi, T., Magli, E.. - In: IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING. - ISSN 0196-2892. - 64:(2026). [10.1109/TGRS.2026.3686438]

Availability:
This version is available at: 11583/3011428 since: 2026-05-27T07:17:38Z

Publisher:
IEEE

Published
DOI:10.1109/TGRS.2026.3686438

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright
IEEE postprint/Author's Accepted Manuscript

©2026 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collecting works, for resale or lists, or reuse of any copyrighted component of this work in other works.

(Article begins on next page)

19 November 2026

Scalable neural pushbroom architectures for real-time denoising of hyperspectral images onboard satellites

Ziyao Yi, Davide Piccinini, Diego Valsesia, *Member, IEEE*, Tiziano Bianchi, *Member, IEEE*, and Enrico Magli, *Fellow, IEEE*

***Abstract*—The next generation of Earth observation satellites will seek to deploy intelligent models directly onboard the payload in order to minimize the latency incurred by the transmission and processing chain of the ground segment, for time-critical applications. Designing neural architectures for onboard execution, particularly for satellite-based hyperspectral imagers, poses novel challenges due to the unique constraints of this environment and imaging system that are largely unexplored by the traditional computer vision literature. In this paper, we show that this setting requires addressing three competing objectives, namely high-quality inference with low complexity, dynamic power scalability and fault tolerance. We focus on the problem of hyperspectral image denoising, which is a critical task to enable effective downstream inference, and highlights the constraints of the onboard processing scenario. We propose a neural network design that addresses the three aforementioned objectives with several novel contributions. In particular, we propose a mixture of denoisers that can be resilient to radiation-induced faults as well as allowing for time-varying power scaling. Moreover, each denoiser employs an innovative architecture where an image is processed line-by-line in a causal way, with a memory of past lines, in order to match the acquisition process of pushbroom hyperspectral sensors and greatly limit memory requirements. We show that the proposed architecture can run in real-time, i.e., process one line in the time it takes to acquire the next one, on low-power hardware and provide competitive denoising quality with respect to significantly more complex state-of-the-art models. We also show that the power scalability and fault tolerance objectives provide a design space with multiple tradeoffs between those properties and denoising quality.**

I. Introduction

HYPERSPECTRAL images (HSIs) [1] acquire a spatial-spectral signal across a wide range of the electromagnetic spectrum, typically tens or hundreds of wavelengths, thus capturing fine information about materials in the imaged scenes, which makes them widely used in multiple fields [2] such as environmental monitoring [3], urban planning [4], and climate change [5].

Satellite HSIs are typically acquired by means of pushbroom sensors, which consist in a linear array of detectors in the across-track direction, i.e., the direction perpendicular to the satellite movement, for each spectral band. A pushbroom sensor thus acquires one line of an image with all its spectral bands within the sensor exposure time, and then the satellite movement in the along-track direction allows to capture successive lines. For instance, existing hyperspectral satellites such as PRISMA [6], EnMAP [7] or HySIS [8] all use pushbroom sensors with largely similar characteristics. The line acquisition time is dictated by factors such as the desired ground sampling distance (typically 30m per pixel), the speed of the satellite due to its orbit as well as the achievable signal-to-noise ratio of the detectors, and it is around 4.5 ms for the aforementioned missions [9]. Limitations in the detector quality, as well as constraints on the exposure time, typically result in significant amounts of noise affecting the images, particularly at infrared wavelengths. Moreover, slightly different gains between detectors of adjacent pixels result in striping noise which may hinder image analysis. Therefore, in order to fully exploit the content of HSIs, denoising [10] is a critical preprocessing operation.

At the same time, the space industry is undergoing a transformation, since it has recognized that the current system where images are downloaded to the ground segment within the limited windows of transmission and then processed with highly accurate but costly algorithms, leads to very long delays, potentially in the order of days, before products are available to the final users. The next generation of Earth observation satellites will seek to move as much processing onboard the satellites as possible to enable low-delay products such as alerts or detection results, which can be transmitted with priority due to their smaller sizes. However, this goal clashes with the severe constraints that onboard computing platforms face and calls for innovative designs that consider these constraints since their inception, rather than just scaling down models developed for more traditional scenarios. First and foremost, the available power on an Earth observation satellite is limited since it comes from solar panels, and it is also variable depending on the current orbital position and attitude changes [11] with respect to the Sun. Moreover, the effects of radiation needs to be considered for all onboard electronics. Traditionally, dedicated hardware with radiation-hardened designs was used, but, recently, there is a trend towards using commercial off-the-shelf components which allows to use the latest hardware advancements, and building radiation proofing systems around them, e.g., with external shielding and components redundancy where an operation may be performed in parallel by multiple systems and results cross-checked.

In this paper, we study how to design a neural architecture for real-time denoising of hyperspectral images onboard a satellite, so that the acquired images are effectively preprocessed for further downstream tasks executed onboard. We first present

The authors are with Politecnico di Torino – Department of Electronics and Telecommunications, Italy. email: {name.surname}@polito.it. This study was carried out within the FAIR - Future Artificial Intelligence Research and received funding from the European Union Next-GenerationEU (PIANO NAZIONALE DI RIPRESA E RESILIENZA (PNRR) – MISSIONE 4 COMPONENTE 2, INVESTIMENTO 1.3 – D.D. 1555 11/10/2022, PE00000013, CIG B421A95680, CUP E13C22001800001). This manuscript reflects only the authors' views and opinions, neither the European Union nor the European Commission can be considered responsible for them.

the unique design goals that this setting entails and then we introduce an approach which allows to control the tradeoffs between the competing objectives. We validate our approach by considering the HSI denoising task. This is a particularly interesting benchmark because careful denoising of HSIs is critical for any downstream inference task, including ones that could be executed onboard. Due to the high-dimensionality of HSI, it is also a good stress test for our proposed approach since i) HSIs may be complex to handle, especially in terms of memory requirements due to the large number of bands, and ii) the need for real-time operations to match the hyperspectral pushbroom acquisition and render the denoised images readily available for further usage.

More in detail, we discuss how the onboard setting requires to account for three competing design goals: i) high inference quality at low computational complexity; ii) power scalability; iii) fault tolerance. Concerning the first objective, we propose a novel denoiser architecture that is specifically tailored to match the acquisition process of pushbroom sensors. Such architecture works in a causal line-by-line fashion with a lightweight memory mechanism, based on Mamba [12], that allows to exploit the features of past lines to denoise the current one. This design greatly limits computational complexity and memory requirements, and can be tailored to run within the acquisition time of one line to enable real-time operations. This is contrast with conventional network designs which need to process large 2D image tiles, and suffer from high memory requirements and complexity. Furthermore, the power scalability objective is concerned with the peculiarities of power availability in space. A power-scalable model allows to be run at multiple power levels, with consequently varying quality, to match time-varying power availability. As we shall see in the paper, better performance at lower power consumptions may come at the expense of top performance with the full power budget. This is an often neglected aspect in the literature for satellite onboard processing, resulting in suboptimal models that either need to be temporarily turned off or are oversimplified. Finally, fault tolerance is required to make the model robust to radiation-induced errors in the space environment, such as corrupting weight values. We show that the latter two objectives can be tackled by designing a mixture of denoisers rather than a single static model, with a shared routing module that performs attention-based fusion and fault filtering. This allows to place individual denoisers in the mixture on independent accelerators which can be smaller and easier to manufacture, can also be turned off to match the time-varying power requirements, and provide a sort of redundancy to faults.

Our main contributions may thus be summarized as follows:

- We introduce a set of design goals for neural architectures intended for onboard satellite deployment, explicitly addressing real-time constraints, power scalability, memory limits, and radiation-induced faults.
- We develop a line-wise hyperspectral denoising architecture and integrate it into a mixture-of-denoisers framework that together enable real-time processing aligned with pushbroom acquisition, while providing both power scalability and intrinsic fault tolerance.
- We propose training and fault-filtering strategies to support flexible power scalability tradeoffs and robustness under radiation-induced model degradation.

II. Background and Related Work

A. HSI Denoising

HSI denoising is an ill-posed inverse problem, that has a long history of methods attempting the reconstruction of a clean image from noisy observations. Mathematically, the observed image $\mathbf{y}$ is modeled as the summation between noise ϵ and the original image $\mathbf{x}$, i.e., $\mathbf{y} = \mathbf{x} + \epsilon$.

Classical model-based approaches to HSI denoising leverage handcrafted priors such as sparse representations [13], low-rank representations [14], and total variation (TV) minimization [15]. BM3D [16] and BM4D [17] introduce the concept of grouping and collaborative filtering to exploit both local and non-local correlations. LRTFL0 [18] restores HSI by approximating them via low-rank block term decomposition. Takemoto et al. [19] propose Graph-SSTV, a TV-based regularization method that constructs a graph capturing the spatial structure of the target HSI.

In terms of learning-based methods, T3SC [20] designs an architecture that is an unrolled version of a sparse coding method. MAC-Net [21] combines model-based spectral low-rank structures and a deep spatial prior. Pang et al. [22] propose a two-branch architecture to extract the spectral correlation between different bands and individually exploit the global and local spatial features. SERT [23] presents a spectral enhanced rectangle Transformer, driving it to explore the non-local spatial similarity and global spectral low-rank property of HSIs pixels. SSRT [24] combines global spectral correlation and non-local spatial self similarity properties within a single SSRT block. Moreover, state space models (SSMs) have recently been adopted for HSI tasks due to their ability to model long-range dependencies efficiently [12]. Despite their promising performance, these methods remain impractical for onboard deployment in spacecraft, primarily due to constraints on computational resources, which makes real-time processing infeasible. In particular, they all have significant memory requirements as well as large computational complexities in terms of FLOPs per pixel. Moreover, none of them matches the acquisition of pushbroom sensors, causing the need to buffer a large number of lines for their operation. A few works adopt mixtures of models for image restoration tasks, typically to have multiple experts for different subtasks. BDE [25] fuses pretrained denoisers in a pixel-wise manner to handle heterogeneous, real-world noise. Jiang [26] combines several models to improve performance in Single Image Super-Resolution. EnsIR[27] casts ensemble learning with a Gaussian mixture model (GMM) to better predict the weight of each single model.

B. Deep sequence modeling

A classical approach to sequence modeling is based on Causal Convolutions, that confine each filter's receptive field to previous timesteps only, preserving the autoregressive order [28]. Alternatively, Recurrent Neural Networks (RNNs) operate by recursively updating a hidden state as new inputs arrive.

A key limitation of RNNs arises when dealing with long sequences, where the models may suffer from vanishing or exploding gradients. Long Short-Term Memory (LSTM) networks [29], and Gated Recurrent Units (GRUs) [30] introduced gating mechanisms to address this issue, but they can still face difficulties in modeling very long-range dependencies.

Transformers [31] rely on attention mechanisms, instead of recurrence, which enable input-dependent operations that capture long-range interactions across the sequence. They have become the standard for sequence processing in many domains, such as for text [32], image [33] and video generation [34] . However, the quadratic computational and memory complexity of the attention with respect to sequence length remains a major bottleneck. Various techniques have been proposed to mitigate this, including sparse attention [35] and linear attention [36], but the scalability challenges persist.

SSMs have recently gained popularity due to their linear complexity with the sequence length. They model a sequence-to-sequence mapping that transforms an input sequence of features into an output sequence by an implicit internal latent state and that evolves through linear dynamics specified by system matrices. Gu et al. [37] demonstrated that carefully constraining the SSM parameters substantially improves model performance in practice. Extending this line of research [38], [39], [40], Gu et al. [41] further introduced Selective SSMs, an enhanced approach where SSM parameters dynamically depend on input data, allowing for a selectivity mechanism that effectively determines when and how information should be propagated or discarded throughout a sequence. Mamba has already shown considerable results across several domains, including audio and language, and has been extended into the area of computer vision[42], [43]. In the field of hyperspectral imagery, Mamba has also been deployed with very promising results [44], [45], [46], [47]. Nonetheless, the deployment of Mamba Blocks for denoising images in an incremental, line-by-line fashion remains unexplored.

C. Onboard processing

In recent years, onboard processing has gained significant attention as an effective strategy for performing real-time satellite data analysis. By processing imagery directly onboard the satellite rather than transmitting extensive raw datasets to Earth, satellites can achieve more agile operational responsiveness, rapidly detect critical phenomena, and enable autonomous decision-making capabilities. Yao et al. [48] introduced lightweight deep learning models on small satellites for maritime vessel detection. Giuffrida et al. [49] demonstrated the use of CNNs on the Phi-Sat-1 mission for real-time cloud detection. Building upon these foundational studies, Ziaja et al. [50] provided comprehensive benchmarks of neural network architectures tailored for space-grade computing devices, and [51] proposed a streamlined variational autoencoder for efficient onboard change detection. Moreover, in [52], the authors presented a modular framework specifically designed for multitask inference onboard satellites. Recently, Valsesia et al. [53] proposed onboard compression of hyperspectral images with a predictive coding mechanism that predicts the next line using RWKV [39], while Piccinini et al. [54], [55] introduced an onboard super-resolution strategy that frames the problem as a previous low resolution line interpolation one. Existing works do not paint a complete picture of onboard processing as they mostly focus on scaled-down designs to limit complexity but neither they consider power scaling or fault resilience goals nor they match the line-wise nature of pushbroom acquisiton.

III. Method

This section presents the main contributions of our work. We start with a formal introduction of the goals that one must consider when designing neural architectures for onboard HSI processing, and we focus on the scenario of a satellite with a hypespectral pushbroom sensor. In the following subsections, we present the ingredients of the proposed design matching those goals and highlight how their competing nature leads to multiple tradeoffs. We thus define the following design goals:

Goal 1 [Low-complexity high-quality inference]
The model should be designed to be run in real-time on low-power hardware matching the pushbroom acquisition dynamic. The quality of the output should be as competitive with the state-of-the-art as possible given the complexity constraint.

Goal 2 [Power scalability]
The model should be able to run at different power levels, allowing switching to a lower or higher level during an orbit.

Goal 3 [Fault tolerance]
The model should be designed to be resilient to radiation-induced faults, detecting them when they happen and providing degraded outputs rather than breaking down.

The first goal guides the design to maximize the effectiveness of the neural model under a complexity constraint. For HSI processing, it is important that the model limits memory consumption, which could otherwise be significant due to the large number of spectral bands, and that the amount of operations to be performed is compatible with the line acquisition time, for real-time performance.

Power scalability suggests that during an orbit (or the satellite lifetime) different power constraints are available. The objective seeks to maximize the effectiveness of the model by producing an output, albeit degraded, even when a lower power is available. Notice that the specific amount of time spent at different power levels may guide the design to prioritize one over the other (e.g., if the maximum power is rarely available, design could prioritize the performance of the model at lower power levels).

Fault tolerance requires the design to take into account radiation-induced faults, according to some fault model. For example, according to [57], each neural network weight sitting in long-term memory may be corrupted. The corruption is usually represented with a large deviation from the nominal value, as it would happen in case of flips in the most significant bits. A fault-tolerant model should be able to withstand some level of faults with a slightly degraded output rather than breaking down.

A. Neural Network Architecture

In this section we present the proposed neural network architecture for onboard HSI denoising that satisfies the

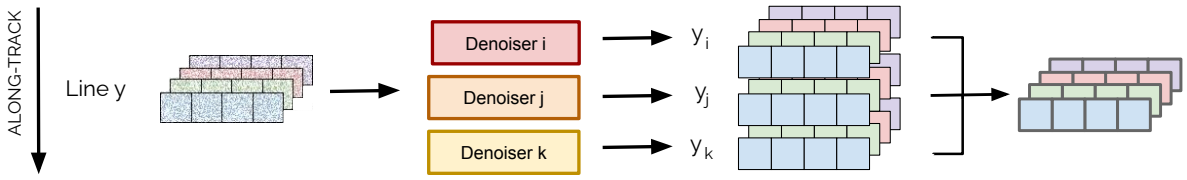
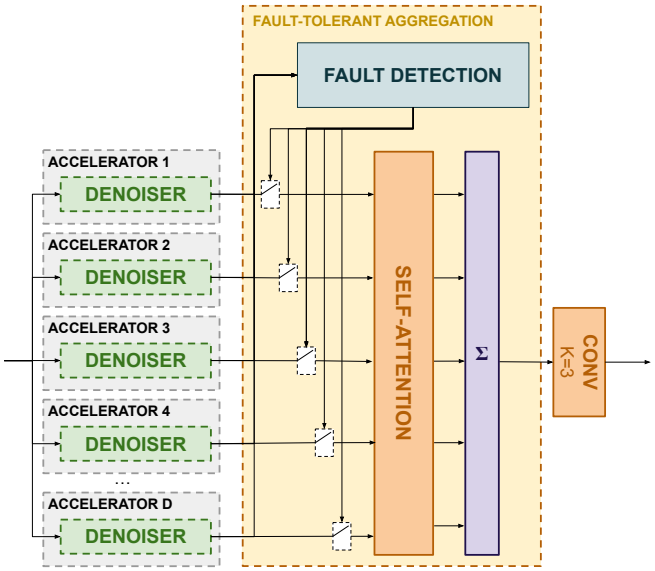
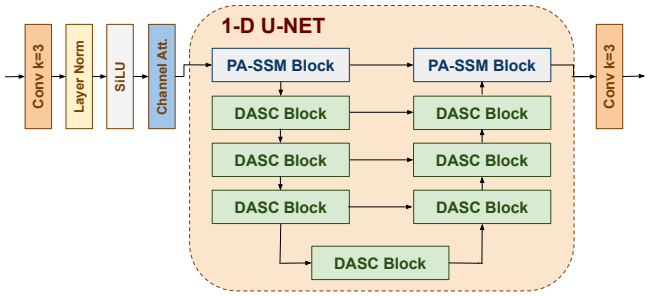


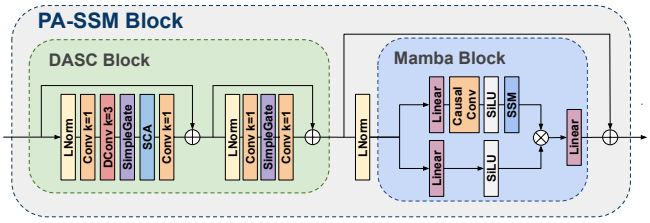
Fig. 1: The acquired line gets processed by each independent line-based denoiser, which makes use of a compact memory of past lines to refine it. The different denoised lines are then aggregated to form the output.



(a) Overall architecture



(b) Denoiser Architecture



(c) PA-SSM Block

Fig. 2: Proposed design. a) overall architecture with a mixture of D denoisers running in parallel: each denoiser is placed on an independent hardware accelerator that can be turned on/off for power scaling; the output features are then checked for faults and aggregated with an attention mechanism. b) individual denoiser architecture, which is a 1-D UNet processing a line with a memory of past lines, consisting of 1-D DASC Blocks [56] and 1-D PA-SSM Blocks (c)).

previously outlined design goals. The overall workflow is shown in Fig. 1: the acquired line is processed by a mixture of independent denoisers, the outputs are dynamically aggregated and averaged. A high-level overview of the architecture is presented in Fig. 2a. In the remainder of the paper, we denote by N_l, N_c, N_b the number of along-track lines, across-track columns and spectral bands of each image, respectively.

The architecture is designed as a mixture of D lightweight denoisers operating in parallel. The denoisers share their architecture, but they have independent parameters, i.e., they are independently trained from different random initializations to ensure a complementary ensemble. The feature maps produced by each not-faulty denoiser are aggregated by means of a residual self-attention operation and averaged before being projected to the image space. As common practice in denoising architectures, this constitutes as an estimate of the noise, which is then subtracted from the noisy input to output the denoised image. More formally, let $\mathbf{y}$ be the noisy image, and let $\mathbf{h}^{(d)} \in \mathbb{R}^{1\times N_c\times F}$ be the feature maps output by denoiser d, then the aggregation stage computes:

$$\hat{\mathbf{x}} = \mathbf{y} - \hat{\mathbf{n}} \tag{1}$$

$$\hat{\mathbf{n}} = \text{conv}\left(\frac{1}{|\mathcal{D}|}\sum_{d\in\mathcal{D}}\mathbf{z}^{(d)}\right) \tag{2}$$

$$\mathbf{z}^{(d)} = \text{SA}(\{\mathbf{h}^{(i)}\}_{i\in\mathcal{D}})^{(d)} + \mathbf{h}^{(d)} \tag{3}$$

$$\begin{aligned}\mathcal{D} &= \mathcal{F}(\{\mathbf{h}^{(i)}\}_{i=1}^{D}) \\ &= \{i \in \{1,2,\dots,D\} \text{ s.t. } i \text{ is not faulty}\}\end{aligned} \tag{4}$$

where $\mathcal{F}$ is the fault detection method described in Sec. III-C, which determines if a denoiser is affected by a fault and returns the set of working denoisers $\mathcal{D}$. The self-attention (SA) operation is the scaled dot-product attention [31], operating on the sequence of $|\mathcal{D}|$ not-faulty denoisers, for each pixel.

The reason for a mixture design can be traced to Goals 2 and 3, which suggest a design not relying on a single large model. In particular, each individual denoiser can be placed on a small independent accelerator, so that each can be turned on or off to meet the power requirement. A training process to tune the desired tradeoff between performance at low power

level or performance at high power levels can be devised (see Sec. III-B). Moreover, a mixture design is inherently more resilient to faults since a fault in one denoiser will not affect the others, if the denoiser can be reliably filtered from contributing to the output. Indeed, one may think of this mixture design as being inspired by the traditional approach used in space of redundant computations, where multiple computers run the same computation in parallel and check each other's output. However, in this case there is no repeated calculation as each denoiser has its own weights and running more than one improves the final result.

The design of each denoiser is guided by Goal 1 in that it needs to be lightweight and match the pushbroom acquisition of the image. The primary challenge is in exploiting both local and long-range spatial context, while respecting stringent constraints on memory footprint and computational cost. For this reason, we propose a novel lightweight architecture that processes an image line-by-line, as it is acquired by the sensor. More in detail, the currently-acquired line with all its spectral bands is processed to extract joint across-track/spectral features. However, the model also needs a memory of past lines in order to exploit inter-line, possibly long-range, correlations among such features. For this purpose we exploit Mamba [41], a recently-proposed SSM, which allows to efficiently capture a memory of the features of past lines in a compact state. As mentioned in Sec. II, Mamba offers a great tradeoff between complexity and performance for this scenario compared to other sequence processing options such as transformers for which KV caching would require large amounts of memory, outperforming RNNs and simple causal convolution. An overview of the denoiser architecture is presented in Fig. 2b.

Since the model processes one line at a time, all the operations are spatially 1-D, with the along-track dimension fixed to 1. First, the N_b spectral channels are projected into a more compact feature space of size F via a learnable 1-D convolution, which also allows to manage memory consumption. In fact, since redundancy in the spectral dimension is generally very high, a modest number of features is generally suitable to properly represent the original bands. This yields shallow features for each row, which are then refined by LayerNorm, followed by a non-linearity (SiLU) and a traditional channel attention block [58]. These features from the current line are then processed by a 1-D U-Net, which largely consists in blocks (DASC Block) inspired by the NAFNet architecture for image restoration [56], adapted to 1-D processing. Each DASC Block provides a lightweight attention-like mechanism to extract across-track/spectral features via the SimpleGate and simplified channel attention operations. We refer the reader to [56] for details of such layers. Downsampling is implemented using a stride-2 1-D convolution, whereas the upsampling makes use of a stride-2 1-D transpose convolution. Skip connections are implemented with feature addition.

The first and last blocks of the U-Net, i.e., the ones operating at the full spatial resolution, are extended with a residual Mamba Block. The Mamba Block has an internal arbitrary feature expansion factor E, while the last linear layer of the block returns to the original feature size F. Mamba leverages a causal convolution with a Selective State-Space module to model the inter-line context with minimal state storage. This operation is performed in parallel for each feature vector representing a pixel in the current line. More formally, let l be the index of the currently processed line. Then we have

$$\mathbf{v'}_l = \mathrm{CausalConv}\big(\mathbf{v}_l\big) \in \mathbb{R}^{N_c \times (FE) \times 1}, \tag{5}$$

where the operation is a 1-D causal convolution with kernel size K that leverages an internal state comprised of a memory of K lines of features, i.e. the $K-1$ previous lines and the current one, to produce tensor $\mathbf{v'_i}$. After a SiLU nonlinearity, the features are fed to the SSM module, where they undergo the following operations:

$$\mathbf{h}_l(t) = \mathbf{A}\mathbf{h}_{l-1}(t) + \mathbf{B}\mathbf{v'}_l(t), \tag{6}$$

$$\mathbf{v''}_l(t) = \mathbf{C}\mathbf{h}_l(t) + \mathbf{D}\mathbf{v'}_l(t), \tag{7}$$

where $\mathbf{h} \in \mathbb{R}^{N_c \times (FE) \times D}$ is the internal state of the model and carries on the knowledge of all the past lines. The dimension D is called the state factor, and controls how big the internal state has to be to effectively store and process all past and current information. Notice that the SSM works in parallel for each pixel in the line, leading to the internal state tensor $\mathbf{h}$ having an independent memory for each across-track pixel. Thanks to the properties of the Mamba Block, the model can effectively store and retrieve information from the past lines.

B. Power Scalability

The power scalability goal we set out for onboard design postulates that our proposed model needs to be able to run at different power levels. This is achieved by designing a D-denoiser mixture model using D different hardware accelerators. Each of these accelerators can be turned on or off as desired, thus some may be disabled to reduce power consumption as needed.

Since the power budget is typically variable over time, we now show how knowledge of its distribution can be used to guide the training process of the mixture model to different tradeoffs. In particular, we show that training can optimize the mixture to use a smaller number of denoisers, at the expense of the peak performance when more denoisers are used, or viceversa. In other words, the training process can skew the performance of the mixture towards subsets of denoisers that are regarded as a more important scenario in the power budget: a scenario in which it is rare that all denoisers are used, but most of the time 2 or 3 are, will seek to maximize the performance of the mixture with the latter number of denoisers, at the expense of performance with all denoisers. To achieve this, we propose a denoiser sampling scheme for training, which randomly samples only a subset of denoisers to be used for a minibatch. A user-defined probability distribution is designed to determine the probability of using $N \leq D$ denoisers in each training minibatch. Specifically, we design this distribution as a discrete exponential controlled by a user-defined parameter λ. Let us call s the number of sampled denoisers, then the probability of using $1 \leq N \leq D$ denoisers for the current training iteration is:

$$\mathbb{P}\left(s = N\right) = \frac{e^{\lambda N}}{\sum_{j=1}^{D} e^{\lambda N}} \tag{8}$$

We call λ the power-scalability factor which can be setup by the system designer according to the desired power scalability properties. Notice that when $\lambda = 0$ all subset cardinalities are equally likely, meaning that there is no preferred power level. When λ assumes large positive values, training will be more likely to always choose to use all denoisers thus the system will optimize for the maximum power level. Conversely, large negative values will optimize the system to use one or few denoisers at the expense of performance when all denoisers are used.

C. Fault Tolerance

As anticipated in Eq. 4, the aggregation function of the mixture model can discard denoisers deemed faulty in order to be resilient to radiation-induced errors. We thus present a detection method that implements function $\mathcal{F}$ in the aforementioned equation. We assume a fairly common fault model in which a weight of a denoiser is perturbed by a large deviation, emulating a bit flip in a significant bit. For simplicity, we also assume that the probability of a fault happening in the aggregation and fault filtering steps is small compared to a fault in one or more denoisers. We propose a detection method that relies on the key, and query projections of the pixel features that are used for aggregation by means of self-attention. In particular, we look at the diagonal of the attention score matrix softmax $\left(\mathbf{k}\mathbf{q}^T/\sqrt{F}\right)$, i.e., the scalar product between the key and query projections of each denoiser (after row softmax normalization). Since an attention matrix is generated for each pixel, we look for spatial variance of each value on the diagonal as a metric of reliability. A low spatial variance should represent the nominal case in which the SA aggregation weight for the current denoiser does not vary wildly over the space dimension. A high variance instead is indicative of a fault as it creates inconsistent features. Hence, the proposed implementation of fault detection is:

$$j \in \mathcal{D} = \mathcal{F}(\{\mathbf{h}^{(i)}\}_{i=1}^{D}) \quad \text{iff} \quad \mathbb{V}ar\left[A_{jj}\right] > \tau \tag{9}$$

$$A = \text{softmax}\left(\mathbf{k}\mathbf{q}^T/\sqrt{F}\right), \tag{10}$$

$$\mathbf{k} = [\mathbf{h}^{(1)}; \ldots; \mathbf{h}^{(D)}]W_k,$$

$$\mathbf{q} = [\mathbf{h}^{(1)}; \ldots; \mathbf{h}^{(D)}]W_q$$

Notice that while the full $D \times D$ attention matrix is computed for fault detection, only the $|\mathcal{D}|$ not-faulty denoisers are used in the aggregation with a reduced $|\mathcal{D}| \times |\mathcal{D}|$ attention matrix (see Eq. 3). The threshold τ can be cross-validated at training time to optimize the accuracy of detection process or other metrics such as precision or recall.

IV. Experimental Results

A. Experimental Setting

For training, we use the ICVL dataset [63], which contains 204 hyperspectral images with a spatial resolution of 1392×1300 and 31 spectral bands. Following the protocol in [10], we select 100 images for training. For evaluation, we employ both synthetic and real-world test sets to assess generalization and robustness. The synthetic data includes 50 images from ICVL, Houston 2018 [64], and Pavia City Center [65]. The real noise dataset includes images from Gaofen-5 Wuhan HSI [66] and Earth Observing-1 HSI [67]. To ensure fair comparison across different methods, we strictly adhere to the standardized noise generation procedure defined in [24]. In particular two main kinds of noise are considered: i) non-i.i.d. Gaussian Noise with three standard deviation ranges ($\sigma \in [0, 15]$, $\sigma \in [0, 55]$, and $\sigma \in [0, 95]$); ii) mixture noise containing non-i.i.d. Gaussian noise with $\sigma \in [0, 95]$, impulse noise applied to one-third of the bands with intensities ranging from 10% to 70%, stripe noise affecting 5% to 15% of columns in one-third of the bands, deadline artifacts on 5% to 15% of columns in one-third of the bands. All noise types are applied to the ICVL dataset, while only mixture noise is used for the Houston 2018 and Pavia City Center datasets.

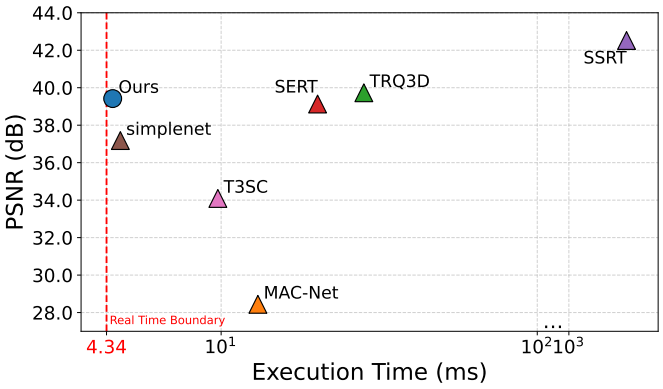


Fig. 3: Runtime on Nvidia Jetson Orin Nano normalized for a $1 \times 1000 \times 66$ line. 4.34ms is the Line Acquisition Time of the PRISMA satellite.

Before training the ensemble, we independently pretrain five of the proposed denoisers for 400 epochs using different random seeds and 64×64 patches. We use $F = 96$ for the each mixture denoiser and $F = 220$ for the one large denoiser. Concerning Mamba, we set $K = 4$, $E = 1$ and $D = 16$ as the kernel of causal convolution, the expansion and state factors, respectively. We use the Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and an initial learning rate of 5×10^{-4}, which is halved at epoch 30 and then every subsequent 100 epochs. For synthetic datasets, performance is quantitatively assessed using: Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM), Spectral Angle Mapper (SAM). For real-noise datasets, we adopt two no-reference image quality metrics, TOPIQ NR [68], and CLIPIQA+ [69]. Training has been performed on a single A40 GPU and requires approximately 5 days. Code will be released at https://github.com/diegovalsesia/scalable-pushbroom-architectures.

B. Quantitative comparisons with state of the art

In this section, we compare our proposed method against a broad spectrum of state-of-the-art HSI denoising techniques, ranging from traditional model-based approaches, including BM4D[17], LRTFL$_0$ [18], FastHyDe [13], and E-3DTV [59], to recent deep learning methods. Concerning the latter, we evaluate convolutional models such as T3SC [20], 3D convolutional networks like MAC-Net [21], transformer-based architectures

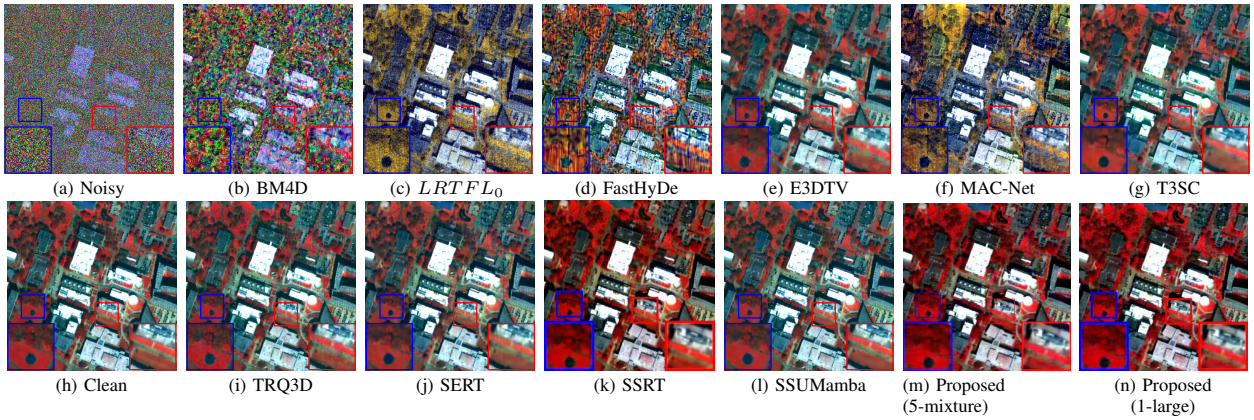


Fig. 4: Denoising results on the Houston 2018 HSI with mixture noise. The false-color images are generated by combining bands 35th, 20th, and 5th.

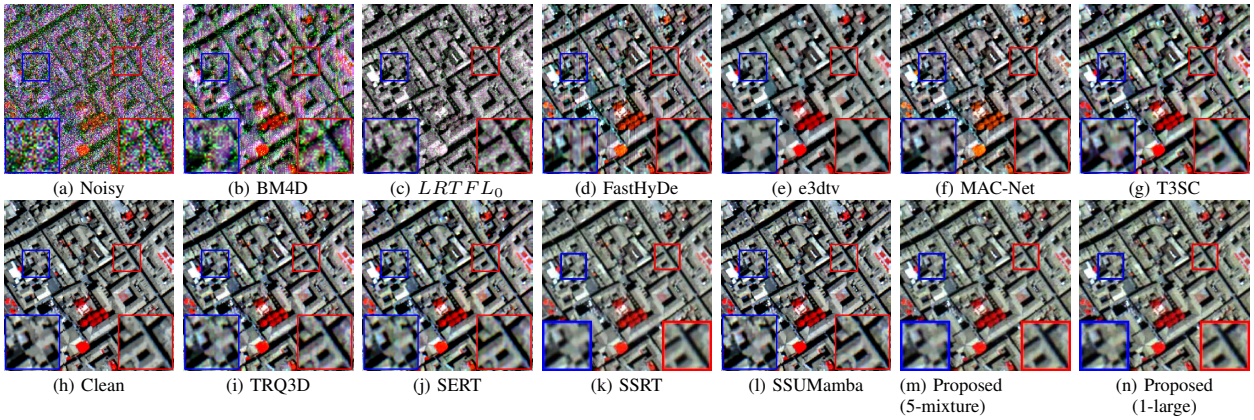


Fig. 5: Denoising results on the Pavia city center with mixture noise, bands 65,45,25.

including TRQ3D [22], SERT [23], and SSRT [24], as well as the SSM-based SSUMamba [12] and LaMamba [62]. Moreover, a very simple and lightweight baseline (SimpleNet) capable of real-time operations was adapted from SIDUNet, a mobile-optimized denoiser originally developed for RAW image enhancement in the AIM 2025 challenge [61].

Our experiment tests the best achievable quality by the proposed architecture, without power scalability tradeoffs or faults. In addition to the results of a mixture of 5 denoisers, we also benchmark the line-wise processing concept by reporting the performance of an individual denoiser with the same complexity as the mixture. Tables I and II show the complexity of the tested models as well as image quality under the various noise settings. We remark that the proposed approach is significantly less complex than existing methods at only 100K FLOPs/pixel, as we seek real-time runtime on low-power hardware, as later validated in Fig. 3. Nevertheless, the proposed approach offers competitive image quality, sometimes even outperforming recent more complex designs. Fig. 4, 5, 6 and 7 shows a qualitative comparison of denoised images.

Finally, we also remark that while SimpleNet provides a lightweight real-time baseline, it is a static architecture lacking the power scalability and fault tolerance mechanisms of our approach. Our proposed design introduces additional architectural components explicitly to address these constraints. Moreover, a 2 dB increase in PSNR from SimpleNet to the proposed design is quite a significant improvement in image quality, including spectral fidelity.

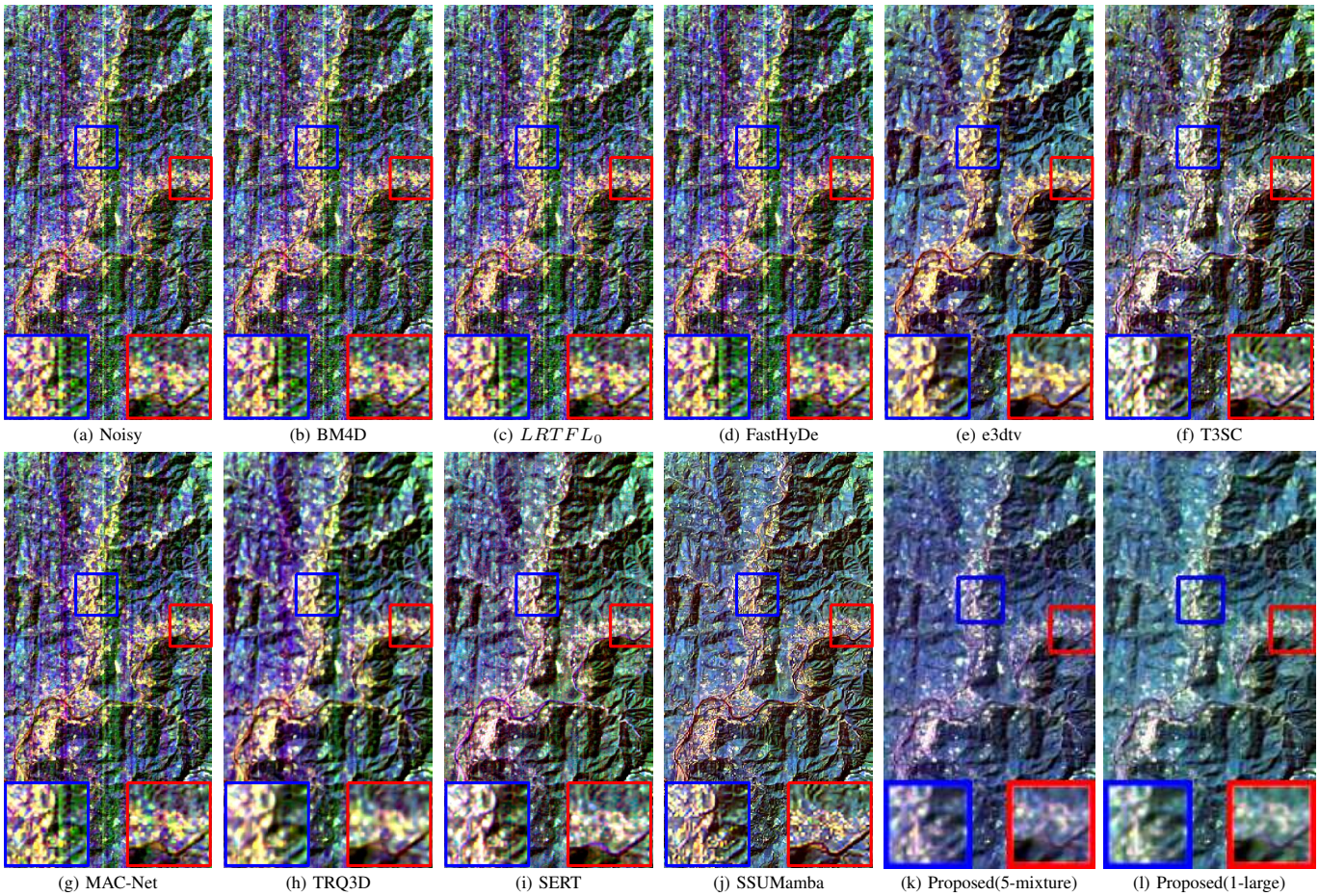
C. Test on low-power hardware

We now study the computational efficiency of the proposed method with respect to the state of the art.

Fig. 8 highlights the superior memory efficiency of our method and its scaling as function of input lines, columns, and bands, showing that large images can be efficiently processed onboard. Notably, thanks to the line-wise design our method has constant memory requirements for any number of acquired lines.

Additionally, we tested runtime on a low-power system (Nvidia Jetson Orin Nano - 15W) that could be considered representative of onboard capabilities. All the architectures have been reduced to 16-bit floating point precision for both weights and activations. We use the Line Acquisition Time of the PRISMA satellite, which is 4.34 ms to acquire a line of size $1 \times 1000 \times 66$ as a reference for real time performance. Results are shown in Table III. The table[1] shows the average time to process one line of size $1 \times 1000 \times 66$. For methods not working on a

[1]We remark that the SSUMamba [12] method is not present in Table III because making the authors' model run on the Jetson platform would require major code modifications.



(a) Noisy (b) BM4D (c) $LRTFL_0$ (d) FastHyDe (e) e3dtv (f) T3SC

(g) MAC-Net (h) TRQ3D (i) SERT (j) SSUMamba (k) Proposed(5-mixture) (l) Proposed(1-large)

Fig. 6: Denoising results on Earth Observing-1 HSI. The false-color images are generated by combining bands 163, 96, and 30

line-by-line basis we process an image of size $1000 \times 1000 \times 66$ and divide the total runtime by the number of lines. Fig. 3 presents the results as a quality against runtime tradeoff. We can see that only the proposed method is capable of achieving the real-time processing requirement of 4.34 ms. These results confirm that, while theoretically providing slightly better image quality, state-of-the-art algorithms are unsuitable for onboard usage as they require impractically large amounts and memory and cannot run in real time, constituting a bottleneck, and the proposed approach is the only suitable for the task.

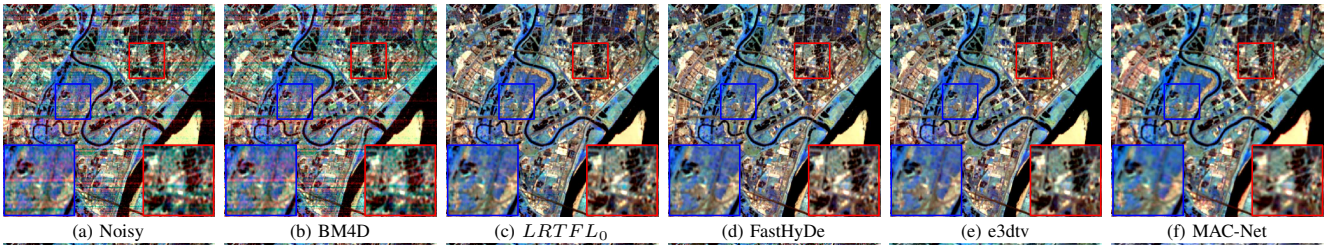
Notably, while our experiments on the Jetson Orin Nano successfully validate the real-time latency capabilities and the memory efficiency of the proposed method, they do not hardware-validate the physical power scalability. Because the Jetson platform is a GPU and does not support physically power-gating specific compute cores to eliminate leakage current, the power-saving mechanism described in Section III-B turning off specific hardware accelerators currently remains a theoretical and architectural design feature. Experiments on power scalability will be validated in our future work.

D. Power Scalability

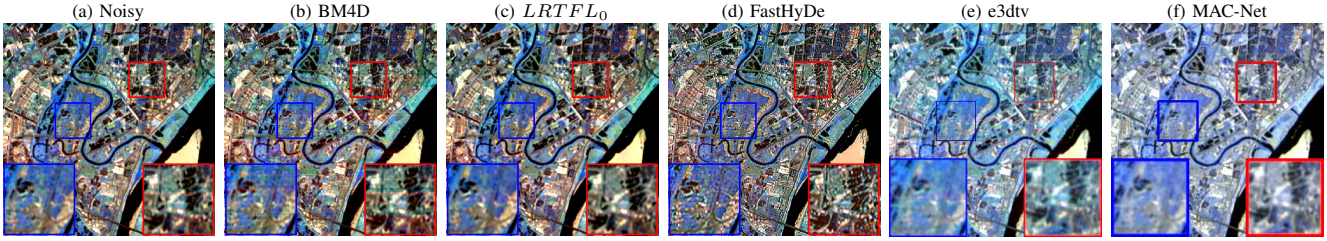
We now analyze the power scalability properties of the proposed design. Concerning power scalability, we trained our mixture model using different power scaling factors $\lambda \in \{-1, -0.5, 0, 0.5, 1\}$ on the ICVL training set with a mixture noise pattern. The results are presented in Fig. 9, where the solid curves denotes the average performance and the error bars denote the variability due to the specific selection of subset of denoisers. We can observe that models trained with smaller λ values exhibit better quality at low power (fewer denoisers used) compared to models where high λ values are used, which conversely perform better in the high power regimes. This offers different tradeoffs to the system designer based on the expected distribution of power over time. Specifically, these models perform better when only a single denoiser is used and show a steeper improvement as the number of denoisers increases. However, this improved scalability sacrificed peak performance, as models with lower λ values tend to underperform when the full ensemble is utilized.

E. Fault Resilience

Regarding fault resilience, we use the PyTorchFI [70] library to simulate hardware faults by randomly injecting errors into the model weights. Focusing specifically on 1D convolutional and linear layers. Across a single denoiser, a total of 817,920 weights are subject to fault injection. Fig. 10 shows how the PSNR of the denoised image degrades as a function of the probability of perturbing a weight, when no fault detection is performed, when faults are detected and filtered with the



(a) Noisy (b) BM4D (c) $LRTFL_0$ (d) FastHyDe (e) e3dtv (f) MAC-Net



(g) T3SC (h) TRQ3D (i) SERT (j) SSUMamba (k) Proposed(5-mixture) (l) Proposed(1-large)

Fig. 7: Denoising results on the GF5-Wuhan HSI with mixture noise,The false-color images are generated by combining bands 152, 83, and 33.

TABLE I: Quantitative results on ICVL dataset under four noise conditions: Gaussian noise with $\sigma \in [0, 15]$, $\sigma \in [0, 55]$, $\sigma \in [0, 95]$, and Mixture Noise.

Method	Params (M)	FLOPs/pixel	$\sigma \in [0, 15]$			$\sigma \in [0, 55]$			$\sigma \in [0, 95]$			Mixture Noise		
			PSNR↑	SSIM↑	SAM↓	PSNR↑	SSIM↑	SAM↓	PSNR↑	SSIM↑	SAM↓	PSNR↑	SSIM↑	SAM↓
BM4D [17]	-	-	44.39	0.9683	0.0692	37.63	0.9008	0.1397	34.71	0.8402	0.1906	23.36	0.4275	0.5476
$LRTFL_0$[18]	-	-	43.41	0.9315	0.0570	35.63	0.8125	0.1914	32.83	0.7482	0.3014	30.93	0.8378	0.3613
FastHyDe [13]	-	-	48.08	0.9917	0.0404	42.86	0.9800	0.0630	40.84	0.9734	0.0771	27.58	0.7250	0.4534
E-3DTV [59]	-	-	46.05	0.9811	0.0560	40.20	0.9505	0.0993	37.80	0.9297	0.1317	34.90	0.9041	0.1468
FastHyMix[60]	-	-	49.29	0.9929	0.0350	43.25	0.9763	0.0764	40.69	0.9590	0.1078	27.08	0.6803	0.4688
SimpleNet[61]	4.3M	13.5K	46.01	0.9884	0.0400	41.91	0.9751	0.0516	40.43	0.9682	0.0581	37.18	0.9381	0.0839
T3SC [20]	0.83M	0.93M	49.68	0.9912	0.0486	45.15	0.9810	0.0652	43.10	0.9734	0.0747	34.09	0.9052	0.2340
MAC-Net [21]	0.43M	14.8K	48.21	0.9915	0.0387	43.74	0.9768	0.0528	41.24	0.9577	0.0841	28.44	0.7393	0.4154
TRQ3D [22]	0.68M	0.44M	46.43	0.9878	0.0437	44.64	0.9840	0.0487	43.54	0.9806	0.0523	39.73	0.9491	0.0869
SERT [23]	1.9M	0.41M	50.18	0.9977	0.0278	46.34	0.9951	0.0373	44.47	0.9929	0.0447	39.13	0.9679	0.0963
SSRT [24]	10.2M	2.78M	52.12	0.9950	0.0225	47.85	0.9894	0.0319	46.20	0.9862	0.0375	42.52	0.9570	0.0986
SSUMamba [12]	10.4M	23.6M	51.34	0.9946	0.0256	46.85	0.9882	0.0375	45.36	0.9853	0.0439	43.07	0.9726	0.0710
LaMamba [62]	24.2M	11.5M	51.87	0.9949	0.0233	47.25	0.9888	0.0366	45.69	0.9856	0.0418	43.00	0.9726	0.0751
Proposed (5-mixture)	5.3M	100K	49.07	0.9915	0.0338	43.65	0.9803	0.0480	42.27	0.9751	0.0560	39.42	0.9468	0.0811
Proposed (1-large)	5.3M	100K	48.93	0.9917	0.0339	44.07	0.9816	0.0472	42.20	0.9741	0.0561	39.67	0.9506	0.0772

method in Sec. III-C and with an oracle fault detector that exactly knows which denoisers are faulty. We can notice that the proposed method is quite robust and performs close to the oracle method, which demonstrates the output quality degradation as the fault probability increases. This behavior is supported by the statistical analysis in Fig. 11, where the variance-based criterion provides a clear separation between faulty and non-faulty denoisers, although the overlap increases at higher fault rates, explaining the residual gap with respect to the oracle. We use a threshold of 0.01 because it gives the best tradeoff between detecting faulty denoisers (high true positive rate) and avoiding false alarms (low false positive rate) in different probability setting. Notably, the proposed fault detection and pruning strategy significantly mitigates the decline in denoising performance, highlighting the robustness of our ensemble model under fault-prone conditions.

In addition to showing that the proposed fault detection approach leads to superior image quality in presence of faults, we perform a statistical analysis to validate our hypothesis that the spatial variance of diagonal attention values does increase in presence of faults. To this end, we conducted an additional experiment in which faults were randomly injected into model weights with probabilities 1×10^{-7}, 5×10^{-7}, and 1×10^{-6}. For each probability level, we performed 30 independent trials and computed the spatial variances of diagonal attention values for both non-faulty and faulty denoisers on each test image in the test set. The resulting distributions are shown in Fig.11.

Across all fault probabilities, we observe that correct denoisers exhibit tightly clustered variance with lower values, indicating stable and coherent attention behavior. Instead, faulty denoisers produce significantly larger variances, forming a distinct distribution that is well separable from the one of non-faulty denoisers. The two distributions overlap more as the fault probability increases, rendering detection more difficult, as expected.

To further validate the robustness of our fault detection mechanism across different image characteristics, particularly in high-frequency scenes such as dense urban environments, we calculated the false positive rate using the EO-1[67] and Pavia City Center [65] datasets. These represent normal-frequency and high-frequency imagery, respectively. As illustrated in Fig. 12, the false positive rate remains largely consistent across both scene types under varying fault probabilities. The maximum observed difference between the two datasets is merely 0.023 at a fault probability of 10^{-6}, demonstrating that our variance-based thresholding method is reliable and does not erroneously

TABLE II: Quantitative comparison of HSI denoising methods on other datasets

Method	Params (M)	FLOPs/pixel	Houston 2018			PAVIA CITY CENTER			GAOFEN-5 WUHAN		EARTH OBSERVING-1	
			PSNR↑	SSIM↑	SAM↓	PSNR↑	SSIM↑	SAM↓	TOPIQ NR↑	CLIPIQA+↑	TOPIQ NR↑	CLIPIQA+↑
BM4D [17]	-	-	22.76	0.4762	0.5168	21.70	0.5128	0.5297	0.3828	0.5827	0.5184	0.5011
$LRTFL_0$[18]	-	-	28.75	0.8038	0.2221	26.49	0.8147	0.3703	0.3877	0.6021	0.5177	0.5204
FastHyDe [13]	-	-	27.07	0.7757	0.4518	26.78	0.8361	0.4040	0.3887	0.6021	0.5235	0.5187
E-3DTV [59]	-	-	30.64	0.8570	0.1323	30.44	0.8941	0.1134	0.3815	0.6055	0.4838	0.5355
FASTHyMix [60]	-	-	25.35	0.7131	0.4593	26.52	0.7982	0.4151	0.3894	0.2704	-	-
Simplenet [61]	4.3M	13.5K	30.03	0.8781	0.1832	29.36	0.8484	0.2278	0.3628	0.5444	0.4552	0.4904
T3SC [20]	0.83M	0.93M	29.84	0.8751	0.1943	28.69	0.8656	0.2135	0.4274	0.5735	0.5258	0.4892
MAC-Net [21]	0.43M	14.8K	28.83	0.7963	0.2356	27.74	0.8724	0.3222	0.3477	0.5549	0.5231	0.4964
TRQ3D [22]	0.68M	0.44M	32.55	0.9194	0.1241	28.23	0.8665	0.1961	0.3840	0.5956	0.3536	0.4437
SERT [23]	1.9M	0.41M	31.31	0.9296	0.1517	32.16	0.9544	0.1384	0.3824	0.6081	0.4654	0.5161
SSRT [24]	10.2M	2.78M	34.71	0.9387	0.0915	35.10	0.9493	0.1036	-	-	-	-
SSUMamba [12]	10.4M	23.6M	34.74	0.9452	0.0993	35.70	0.9546	0.1057	0.4969	0.6481	0.6074	0.5559
LAMamaba [62]	24.2M	11.5M	35.10	0.9462	0.0950	36.67	0.9634	0.1000	-	-	0.5853	0.5092
Proposed (5-mixture)	5.3M	100K	30.39	0.8843	0.1577	33.23	0.9202	0.1808	0.3902	0.5940	0.5114	0.5184
Proposed (1-large)	5.3M	100K	29.86	0.8753	0.1772	33.30	0.9173	0.1854	0.3824	0.5992	0.5026	0.5161

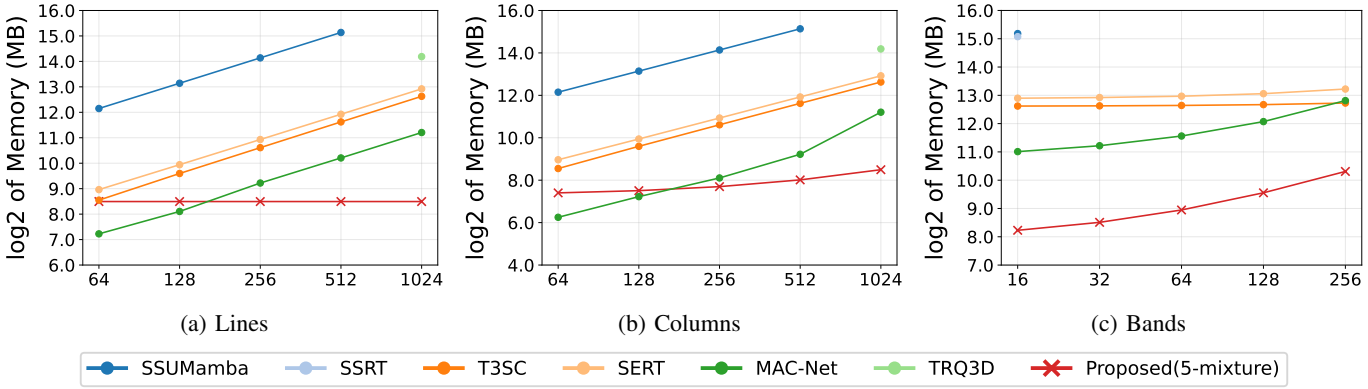


Fig. 8: Memory usage comparison across three dimensions: number of lines, and columns, spectral bands. Image size is a) $N_l \times 1024 \times 31$, b) $1024 \times N_c \times 31$, c) $1024 \times 1024 \times N_b$. Note that SSRT and TRQ3D support only square inputs and TRQ3D only supports a 31-band input. SSRT and SSUMamba run out of memory when the number of bands reaches 32.

flag complex, highly-textured scenes as faulty.

Overall, since the analysis has been conducted on a large number of images with several fault realizations, it is quite robust in showing evidence that the spatial variance of diagonal attention values does increase in presence of faults and its thresholding is an effective method for fault filtering.

F. Ablation of memory mechanism

We tested the proposed method (1-large configuration) with other memory mechanisms, namely Causal Convolution and LSTM layers, in place of Mamba layers. Specifically, we fixed all the rest of the architecture and every hyperparameter and just substituted the two Mamba layers first with two Causal Convolution with kernel 4 or with two LSTM layers having hidden the dimension equal to the model feature dimension. This allowed for the preservation of the line based acquisition system. We trained the Causal Convolution and LSTM models from scratch, employing the same training procedure used for the Mamba model. The results are shown in Table IV. As expected, the selectivity of Mamba yields better performance whilst requiring fewer computational resources.

G. Sensitivity to Number of Denoisers D

Different numbers of denoisers lead to different tradeoffs in terms of peak performance, power scalability, and fault resilience. For example, using too few denoisers reduces the granularity of power scaling and limits fault tolerance, while using too many can reduce peak performance and increase power requirements due to the need for additional accelerators. Our main results adopted five denoisers ($D = 5$) as a balanced configuration that offers a good tradeoff among performance, scalability, and robustness in our baseline experiments. In this section, we investigate the sensitivity of such properties to different numbers of denoisers, by also considering a mixture with a small number of denoisers ($D = 3$) and a mixture with

TABLE III: Inference speed comparison.

Methods	**Inference Time**	**PSNR**
T3SC	9.76 ms	34.09 dB
MAC-Net	13.07 ms	28.44 dB
TRQ3D	28.28 ms	39.73 dB
SERT	20.19 ms	39.13 dB
SimpleNet	4.80 ms	37.18 dB
Ours	**4.54 ms**	**39.42 dB**

TABLE IV: Comparison with other sequence modeling architectures. Results obtained on ICVL with mixture of noise.

Model	**PSNR**	**FLOPs/pixel**
CausalConv	39.17 dB	121.6 K
LSTM	39.52 dB	145.6 K
Mamba	**39.67 dB**	**100 K**

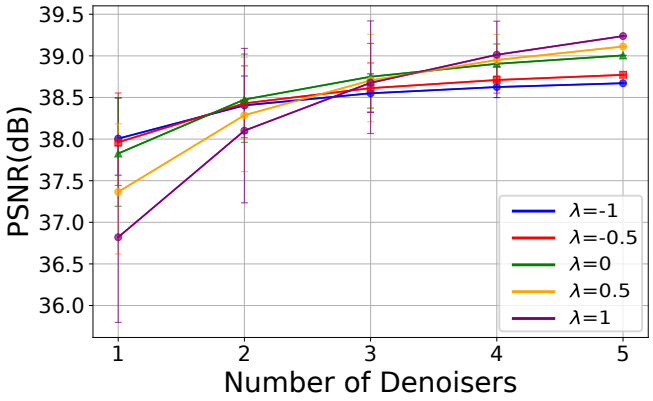


Fig. 9: Power scalability tradeoffs. Power scalability factor λ controls prioritization of high-power or low-power regimes.

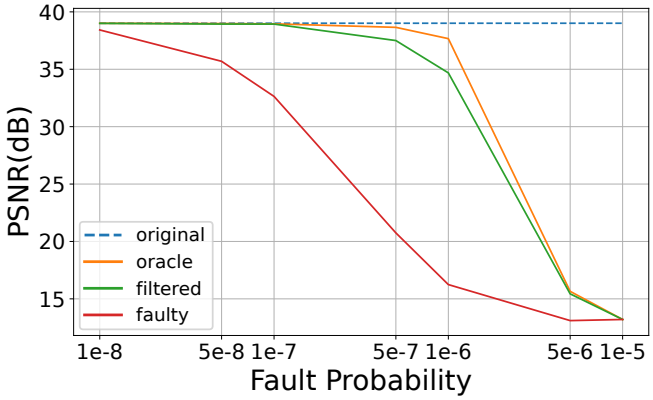


Fig. 10: Output PSNR for the original model (no faults), the oracle fault detector, the proposed fault detector, and the unfiltered (faulty) mixture.

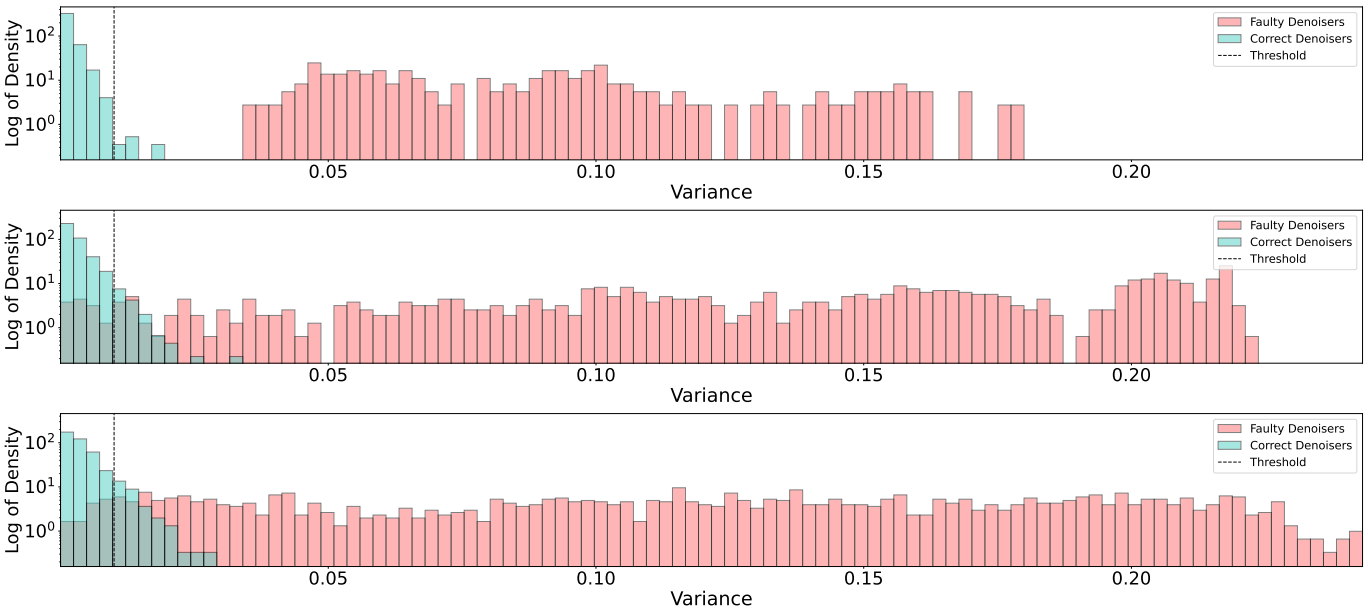


Fig. 11: Variance density distribution with probabilities 1×10^{-7}, 5×10^{-7}, and 1×10^{-6}

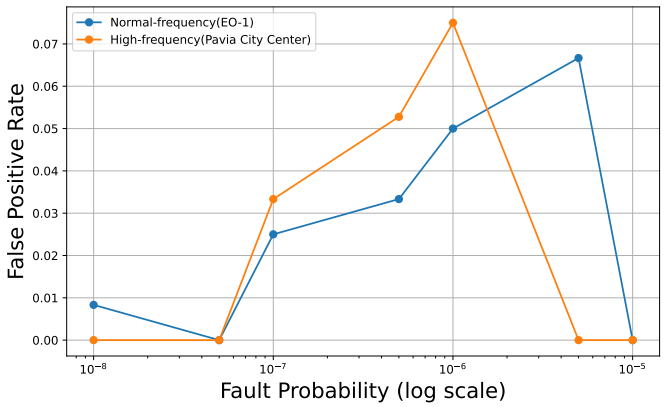


Fig. 12: False positive rate as a function of fault probability on textured/non-textured scenes.

a relatively large number of denoisers ($D = 7$). We remark that as we scale the number of denoisers, the total model size (total number of parameters) remains constant for fairness of comparisons. This also matches a hypothesis of an FPGA-like hardware configuration where total maximum power and logic gates are fixed, and we split the resources depending on D.

TABLE V: Performance Comparison with different denoisers number in ICVL with mixture noise

Model	PSNR	SSIM	SAM
$D = 1$	39.67 dB	0.9506	0.0772
$D = 3$	39.45 dB	0.9473	0.0833
$D = 5$	39.42 dB	0.9468	0.0811
$D = 7$	39.29 dB	0.9470	0.0857

Peak performance. In this experiment, we evaluate the peak performance with different numbers of denoisers, following the same setting as in Table I. As shown in Table V, increasing the number of denoisers leads to slight degradations in PSNR and SAM. This is expected due to the fixed number of total parameters and the general behavior of ensembles of neural networks, where a few larger models outperform large

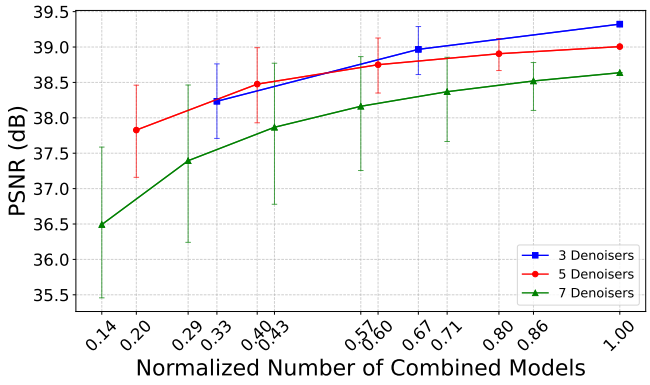


Fig. 13: Sensitivity of Power Scalability to total number of denoisers D. The x axis can be interpreted as a percentage of maximum power.

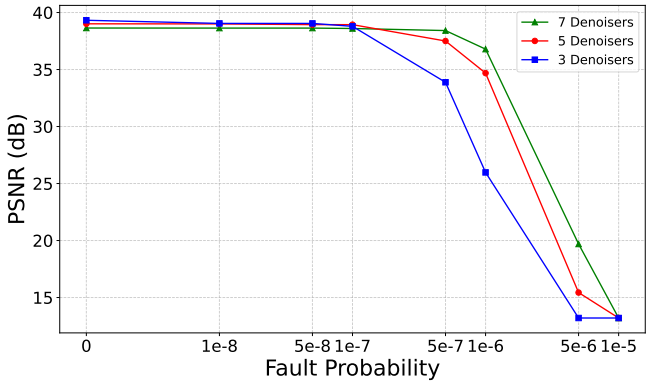


Fig. 14: Sensitivity of fault detection to total number of denoisers D.

ensembles of smaller models.

Power Scalability. We now investigate the power scalability behavior as a function of the number of denoisers. For this experiment, we set $\lambda = 0$ and use the ICVL training set with mixture noise pattern. The results are shown in Fig. 13. Similarly to Fig. 9, the solid curve denotes the average performance and the error bar presents the variability. However, the horizontal axis in Fig. 13 denotes the normalized number of denoisers (the number of used denoisers divided by D) and can be interpreted as the percentage of maximum power actually used. We observe that increasing the number of denoisers improves power scalability by providing finer-grained performance levels at different power budgets. However, this comes at the cost of slightly reduced image quality, consistent with the peak-performance analysis.

Fault Resilience. In this section, we replicate the experiment in Sec. IV-E to compare the resilience to faults of the $D = 3, 5, 7$ mixture configurations. The results are shown in Fig.14. We can clearly see that using a larger number of denoisers improves the fault resilience of the system by leading to superior image quality at a given fault probability.

V. Conclusions and Limitations

We addressed the design problem of neural networks onboard satellites by postulating three goals: i) high inference quality at low complexity; ii) power scalability; iii) fault tolerance. Based on that, we presented a mixture-of-denoisers model with a line-wise architecture for HSI denoising that meets the demands of real-time processing, while addressing the power scalability and fault tolerance goals. The main limitation of the proposed design is the causality of the line-wise processing which cannot exploit the information of future lines, limiting quality. Future work will address hybrid designs that while maintaining the line-based approach for memory efficiency may use bidirectional processing when coupled with a small buffer of lines.

References

[1] S.-E. Qian, "Hyperspectral satellites, evolution, and development history," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 7032–7056, 2021.

[2] M. J. Khan, H. S. Khan, A. Yousaf, K. Khurshid, and A. Abbas, "Modern trends in hyperspectral image analysis: A review," *Ieee Access*, vol. 6, pp. 14 118–14 129, 2018.

[3] W. A. Obermeier, L. W. Lehnert, M. Pohl, S. M. Gianonni, B. Silva, R. Seibert, H. Laser, G. Moser, C. Müller, J. Luterbacher *et al.*, "Grassland ecosystem services in a changing environment: The potential of hyperspectral monitoring," *Remote Sensing of Environment*, vol. 232, p. 111273, 2019.

[4] C. Weber, R. Aguejdad, X. Briottet, J. Avala, S. Fabre, J. Demuynck, E. Zenou, Y. Deville, M. S. Karoui, F. Z. Benhalouche *et al.*, "Hyperspectral imagery for environmental urban planning," in *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2018, pp. 1628–1631.

[5] M. Durairaj, K. Rubenraju, B. R. Krishna, M. S. A. Yajid, B. Jackson, J. S. Babu, and K. L. Padmavathi, "Sustainable agriculture-based climate change training models using remote hyperspectral image with machine learning model," *Remote Sensing in Earth Systems Sciences*, pp. 1–10, 2024.

[6] S. Cogliati, F. Sarti, L. Chiarantini, M. Cosi, R. Lorusso, E. Lopinto, F. Miglietta, L. Genesio, L. Guanter, A. Damm *et al.*, "The prisma imaging spectroscopy mission: overview and first performance analysis," *Remote sensing of environment*, vol. 262, p. 112499, 2021.

[7] M. Brell, L. Guanter, K. Segl, D. Scheffler, N. Bohn, A. Bracher, M. A. Soppa, S. Foerster, T. Storch, M. Bachmann, and S. Chabrillat, "The enmap satellite –data product validation activities," in *2021 11th Workshop on Hyperspectral Imaging and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, 2021, pp. 1–5.

[8] eoPortal Directory. (2024) Hysis (hyperspectral imaging satellite). Accessed: 2025-05-02. [Online]. Available: https://www.eoportal.org/satellite-missions/hysis#launch

[9] A. S. Italiana, "Prisma algorithm theoretical basis document (atbd)," 2021, https://prisma.asi.it/missionselect/docs/PRISMA%20ATBD_v1.pdf [Accessed: 07/05/2025].

[10] K. Wei, Y. Fu, and H. Huang, "3-d quasi-recurrent neural network for hyperspectral image denoising," *IEEE transactions on neural networks and learning systems*, vol. 32, no. 1, pp. 363–375, 2020.

[11] E. Lagunas, F. G. Ortiz, G. Eappen, S. Daoud, W. A. Martins, J. Querol, S. Chatzinotas, N. Skatchkovsky, B. Rajendran, and O. Simeone, "Performance evaluation of neuromorphic hardware for onboard satellite communication applications," *IEEE Wireless Communications*, vol. 31, no. 6, pp. 78–84, 2024.

[12] G. Fu, F. Xiong, J. Lu, and J. Zhou, "Ssumamba: Spatial-spectral selective state space model for hyperspectral image denoising," *IEEE Transactions on Geoscience and Remote Sensing*, 2024.

[13] L. Zhuang and J. M. Bioucas-Dias, "Fast hyperspectral image denoising and inpainting based on low-rank and sparse representations," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 11, no. 3, pp. 730–742, 2018.

[14] H. Zhang, W. He, L. Zhang, H. Shen, and Q. Yuan, "Hyperspectral image restoration using low-rank matrix recovery," *IEEE transactions on geoscience and remote sensing*, vol. 52, no. 8, pp. 4729–4743, 2013.

[15] S. Yang and Z. Shi, "Hyperspectral image target detection improvement based on total variation," *IEEE Transactions on Image Processing*, vol. 25, no. 5, pp. 2249–2258, 2016.

[16] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-d transform-domain collaborative filtering," *IEEE Transactions on Image Processing*, vol. 16, no. 8, pp. 2080–2095, 2007.

[17] M. Maggioni, V. Katkovnik, K. Egiazarian, and A. Foi, "Nonlocal transform-domain filter for volumetric data denoising and reconstruction," *IEEE Transactions on Image Processing*, vol. 22, no. 1, pp. 119–133, 2013.

[18] F. Xiong, J. Zhou, and Y. Qian, "Hyperspectral restoration via l_0 gradient regularized low-rank tensor factorization," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 12, pp. 10 410–10 425, 2019.

[19] S. Takemoto, K. Naganuma, and S. Ono, "Graph spatio-spectral total variation model for hyperspectral image denoising," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2022.

[20] T. Bodrito, A. Zouaoui, J. Chanussot, and J. Mairal, "A trainable spectral-spatial sparse coding model for hyperspectral image restoration," *Advances in Neural Information Processing Systems*, vol. 34, pp. 5430–5442, 2021.

[21] F. Xiong, J. Zhou, Q. Zhao, J. Lu, and Y. Qian, "Mac-net: Model-aided nonlocal neural network for hyperspectral image denoising," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–14, 2022.

[22] L. Pang, W. Gu, and X. Cao, "Trq3dnet: A 3d quasi-recurrent and transformer based network for hyperspectral image denoising," *Remote Sensing*, vol. 14, no. 18, p. 4598, 2022.

[23] M. Li, J. Liu, Y. Fu, Y. Zhang, and D. Dou, "Spectral enhanced rectangle transformer for hyperspectral image denoising," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 5805–5814.

[24] G. Fu, F. Xiong, J. Lu, J. Zhou, J. Zhou, and Y. Qian, "Hyperspectral image denoising via spatial-spectral recurrent transformer," *IEEE Transactions on Geoscience and Remote Sensing*, 2024.

[25] P. Liu, H. Zhang, J. Wang, Y. Wang, D. Ren, and W. Zuo, "Robust deep ensemble method for real-world image denoising," *Neurocomputing*, vol. 512, pp. 1–14, 2022.

[26] J. Jiang, Y. Yu, Z. Wang, S. Tang, R. Hu, and J. Ma, "Ensemble super-resolution with a reference dataset," *IEEE Transactions on Cybernetics*, vol. 50, no. 11, pp. 4694–4708, 2020.

[27] S. Sun, W. Ren, Z. Liu, H. Park, R. Wang, and X. Cao, "Ensir: An ensemble algorithm for image restoration via gaussian mixture models," *Advances in Neural Information Processing Systems*, vol. 37, pp. 133 487–133 517, 2024.

[28] A. Van Den Oord, N. Kalchbrenner, and K. Kavukcuoglu, "Pixel recurrent neural networks," in *International conference on machine learning*. PMLR, 2016, pp. 1747–1756.

[29] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[30] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[31] A. Vaswani, "Attention is all you need," *Advances in Neural Information Processing Systems*, 2017.

[32] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample, "Llama: Open and efficient foundation language models," 2023. [Online]. Available: https://arxiv.org/abs/2302.13971

[33] A. Hatamizadeh, J. Song, G. Liu, J. Kautz, and A. Vahdat, "Diffit: Diffusion vision transformers for image generation," in *European Conference on Computer Vision*. Springer, 2024, pp. 37–55.

[34] T. Brooks, B. Peebles, C. Holmes, W. DePue, Y. Guo, L. Jing, D. Schnurr, J. Taylor, T. Luhman, E. Luhman *et al.*, "Video generation models as world simulators," *OpenAI Blog*, vol. 1, p. 8, 2024.

[35] R. Child, S. Gray, A. Radford, and I. Sutskever, "Generating long sequences with sparse transformers," *arXiv preprint arXiv:1904.10509*, 2019.

[36] A. Katharopoulos, A. Vyas, N. Pappas, and F. Fleuret, "Transformers are rnns: Fast autoregressive transformers with linear attention," in *International conference on machine learning*. PMLR, 2020, pp. 5156–5165.

[37] A. Gu, K. Goel, and C. Ré, "Efficiently modeling long sequences with structured state spaces," *arXiv preprint arXiv:2111.00396*, 2021.

[38] D. Y. Fu, T. Dao, K. K. Saab, A. W. Thomas, A. Rudra, and C. Re, "Hungry hungry hippos: Towards language modeling with state space models," in *The Eleventh International Conference on Learning Representations*.

[39] B. Peng, E. Alcaide, Q. G. Anthony, A. Albalak, S. Arcadinho, S. Biderman, H. Cao, X. Cheng, M. N. Chung, L. Derczynski *et al.*, "Rwkv: Reinventing rnns for the transformer era," in *The 2023 Conference on Empirical Methods in Natural Language Processing*.

[40] M. Poli, S. Massaroli, E. Nguyen, D. Y. Fu, T. Dao, S. Baccus, Y. Bengio, S. Ermon, and C. Ré, "Hyena hierarchy: Towards larger convolutional language models," in *International Conference on Machine Learning*. PMLR, 2023, pp. 28 043–28 078.

[41] A. Gu and T. Dao, "Mamba: Linear-time sequence modeling with selective state spaces," *arXiv preprint arXiv:2312.00752*, 2023.

[42] L. Zhu, B. Liao, Q. Zhang, X. Wang, W. Liu, and X. Wang, "Vision mamba: Efficient visual representation learning with bidirectional state space model," in *Forty-first International Conference on Machine Learning*.

[43] Y. Liu, Y. Tian, Y. Zhao, H. Yu, L. Xie, Y. Wang, Q. Ye, J. Jiao, and Y. Liu, "Vmamba: Visual state space model," *Advances in neural information processing systems*, vol. 37, pp. 103 031–103 063, 2024.

[44] Y. Li, Y. Luo, L. Zhang, Z. Wang, and B. Du, "Mambahsi: Spatial-spectral mamba for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, 2024.

[45] J. Yao, D. Hong, C. Li, and J. Chanussot, "Spectralmamba: Efficient mamba for hyperspectral image classification," *arXiv preprint arXiv:2404.08489*, 2024.

[46] J. Sheng, J. Zhou, J. Wang, P. Ye, and J. Fan, "Dualmamba: A lightweight spectral-spatial mamba-convolution network for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, 2024.

[47] L. Huang, Y. Chen, and X. He, "Spectral-spatial mamba for hyperspectral image classification," *arXiv preprint arXiv:2404.18401*, 2024.

[48] Y. Yao, Z. Jiang, H. Zhang, and Y. Zhou, "On-board ship detection in micro-nano satellite based on deep learning and cots component," *Remote Sensing*, vol. 11, no. 7, p. 762, 2019.

[49] G. Giuffrida, L. Fanucci, G. Meoni, M. Batič, L. Buckley, A. Dunne, C. Van Dijk, M. Esposito, J. Hefele, N. Vercruyssen *et al.*, "The ϕ-sat-1 mission: The first on-board deep neural network demonstrator for satellite earth observation," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–14, 2021.

[50] M. Ziaja, P. Bosowski, M. Myller, G. Gajoch, M. Gumiela, J. Protich, K. Borda, D. Jayaraman, R. Dividino, and J. Nalepa, "Benchmarking deep learning for on-board space applications," *Remote Sensing*, vol. 13, no. 19, p. 3981, 2021.

[51] V. Ruzicka, A. Vaughan, D. De Martini, J. Fulton, V. Salvatelli, C. Bridges, G. Mateo-Garcia, and V. Zantedeschi, "Ravaen: unsupervised change detection of extreme events using ml on-board satellites," *Scientific reports*, vol. 12, no. 1, p. 16939, 2022.

[52] G. Inzerillo, D. Valsesia, and E. Magli, "Efficient onboard multi-task ai architecture based on self-supervised learning," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2024.

[53] D. Valsesia, T. Bianchi, and E. Magli, "Onboard deep lossless and near-lossless predictive coding of hyperspectral images with line-based attention," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 62, pp. 1–14, 2024.

[54] D. Piccinini, D. Valsesia, and E. Magli, "Towards deep line-based architectures for onboard hyperspectral image super-resolution." in *IGARSS 2025-2025 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2025.

[55] ——, "Onboard hyperspectral super-resolution with deep pushbroom neural network," *arXiv preprint arXiv:2507.20765*, 2025.

[56] L. Chen, X. Chu, X. Zhang, and J. Sun, "Simple baselines for image restoration," in *European conference on computer vision*. Springer, 2022, pp. 17–33.

[57] Y. Cai, M. Cai, Y. Wu, J. Lu, Z. Bian, B. Liu, and S. Cui, "Evaluation and mitigation of weight-related single event upsets in a convolutional neural network," *Electronics*, vol. 13, no. 7, p. 1296, 2024.

[58] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam: Convolutional block attention module," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 3–19.

[59] J. Peng, Q. Xie, Q. Zhao, Y. Wang, D. Meng, and Y. Leung, "Enhanced 3dtv regularization and its applications on hyper-spectral image denoising and compressed sensing," *arXiv preprint arXiv:1809.06591*, 2018.

[60] L. Zhuang and M. K. Ng, "Fasthymix: Fast and parameter-free hyperspectral image mixed noise removal," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 8, pp. 4702–4716, 2021.

[61] F. Li, J. Li, M. V. Conde, B. Besbinar, V. Hosu, D. Iso, and R. Timofte, "Aim 2025 challenge on real-world raw image denoising," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025, pp. 5685–5698.

[62] P. Duan, Y. Luo, X. Kang, and S. Li, "Lamamba: Linear attention mamba for hyperspectral image denoising," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 63, pp. 1–13, 2025.

[63] B. Arad and O. Ben-Shahar, "Sparse recovery of hyperspectral signal from natural rgb images," in *European Conference on Computer Vision*. Springer, 2016, pp. 19–34.

[64] Y. Zhang, W. Li, M. Zhang, S. Wang, R. Tao, and Q. Du, "Graph information aggregation cross-domain few-shot learning for hyperspectral image classification," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–14, 2022.

[65] N. Audebert, B. Le Saux, and S. Lefèvre, "Deep learning for classification of hyperspectral data: A comparative review," *IEEE geoscience and remote sensing magazine*, vol. 7, no. 2, pp. 159–173, 2019.

[66] Y.-N. Liu, D.-X. Sun, X.-N. Hu, X. Ye, Y.-D. Li, S.-F. Liu, K.-Q. Cao, M.-Y. Chai, W.-Y.-N. Zhou, J. Zhang *et al.*, "The advanced hyperspectral imager: Aboard china's gaofen-5 satellite," *IEEE Geoscience and Remote Sensing Magazine*, vol. 7, no. 4, pp. 23–32, 2019.

[67] E. M. Middleton, S. G. Ungar, D. J. Mandl, L. Ong, S. W. Frye, P. E. Campbell, D. R. Landis, J. P. Young, and N. H. Pollack, "The earth observing one (eo-1) satellite mission: Over a decade in space," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 6, no. 2, pp. 243–256, 2013.

[68] C. Chen, J. Mo, J. Hou, H. Wu, L. Liao, W. Sun, Q. Yan, and W. Lin, "Topiq: A top-down approach from semantics to distortions for image quality assessment," *IEEE Transactions on Image Processing*, 2024.

[69] J. Wang, K. C. Chan, and C. C. Loy, "Exploring clip for assessing the look and feel of images," in *AAAI*, 2023.

[70] A. Mahmoud, N. Aggarwal, A. Nobbe, J. R. S. Vicarte, S. V. Adve, C. W. Fletcher, I. Frosio, and S. K. S. Hari, "Pytorchfi: A runtime perturbation tool for dnns," in *2020 50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*, 2020, pp. 25–31.



Ziyao Yi received the M.Sc. degree in Electrical Engineering from Washington University in St. Louis, USA, 2019. Prior to beginning the Ph.D., he worked for four years as a Computer Vision Algorithm Engineer in industry, where he developed and optimized computer vision and deep learning algorithms for various applications. Currently, he is pursuing the Ph.D. degree at Politecnico di Torino, Italy. His primary research interests include learning-based image/video restoration and the development of efficient deep learning models and algorithms.



Davide Piccinini (S'25) received the Master degree in Mathematics in 2023 from Università di Torino, Dipartimento di Matematica. He later completed a fellowship in the Image Processing and Learning group of the Politecnico di Torino, Turin, Italy, where he is currently pursuing a Ph.D. degree in deep learning.



Diego Valsesia (S'13-M'17) received the Ph.D. degree in electronic and communication engineering from the Politecnico di Torino, in 2016. He is currently an Associate Professor with the Department of Electronics and Telecommunications (DET), Politecnico di Torino. His main research interests include processing of remote sensing images, and deep learning for inverse problems in imaging. He is a Senior Area Editor for the IEEE Transactions on Image Processing, for which he received the 2023 Outstanding Editorial Board Member Award. He is a member of the EURASIP Technical Area Committee for Signal and Data Analytics for Machine Learning and a member of the ELLIS society. He was the recipient of the IEEE ICIP 2019 Best Paper Award, the IEEE Multimedia 2019 Best Paper Award.



Tiziano Bianchi received the M.Sc. degree (Laurea) in electronic engineering and the Ph.D. degree in information and telecommunication engineering from the University of Florence, Italy, in 2001 and 2005, respectively. From 2005 to 2012, he was a Research Assistant with the Department of Electronics and Telecommunications, University of Florence. He joined Politecnico di Torino in 2012 as an Assistant Professor. He is currently a Full Professor with the Politecnico di Torino. He has authored over 100 papers in international journals and conference proceedings. His research interests include multimedia security technologies, multimedia forensics, processing of remote sensing images. He is currently Senior Area Editor for the IEEE Transactions on Information Forensics and Security and Senior Area Editor for the Journal of Visual Communication and Image Representation. He was the recipient of the IEEE Multimedia 2019 Best Paper Award and the 2021 and 2022 Best Associate Editor Award of the Journal of Visual Communication and Image Representation.



Enrico Magli (S'97-M'01-SM'07-F'17) received his PhD from Politecnico di Torino, Italy, and is a Full Professor in the same university, where he leads the Image Processing and Learning group, performing research in the fields of artificial intelligence for image and video processing, with application to satellite imaging and computer vision. He is also co-founder and President of ToothPic, a high-tech company in the field of cybersecurity. He is a Fellow of the IEEE and a Fellow of the ELLIS Society for the advancement of artificial intelligence in Europe. He is a Senior Associate Editor of IEEE Journal on Selected Topics in Signal Processing, and a former Associate Editor of the IEEE Transactions on Multimedia and the IEEE Transactions on Circuits and Systems for Video Technology. He was the recipient of the IEEE Geoscience and Remote Sensing Society 2011 Transactions Prize Paper Award, the IEEE ICIP 2015 Best Student Paper Award, the IEEE ICIP 2019 Best Paper Award, the IEEE Multimedia 2019 Best Paper Award.