Raw-to-Processed in Orbit: Fast Onboard Image Reconstruction for EO

Anonymous CVPR submission

Paper ID 13

Abstract

Onboard Artificial Intelligence (AI) is increasingly adopted in Earth Observation (EO) missions to reduce downlink requirements and enable low-latency decision-making. However, converting raw acquisitions into usable products still relies on computationally intensive preprocessing steps that are typically executed on onboard CPU, creating a major bottleneck for resource-constrained CubeSats such as ΦSat-2. In this work, we propose an end-to-end Deep Learning (DL) pipeline that reconstructs processed image products directly from raw data, explicitly designed for deployment on space-qualified AI accelerators. Our approach combines (i) a lightweight learned multispectral band coregistration module and (ii) an efficient wavelet-domain image reconstruction network. Experiments show that the proposed method achieves up to a $6.63\times$ *latency reduction compared to the traditional onboard CPU-based preprocessing chain, while preserving reconstruction fidelity. Overall, these results suggest that learned raw-to-processed reconstruction is a viable and efficient alternative to conventional onboard preprocessing chain, and provide a step toward fully AI-driven onboard EO pipelines.*

1. Introduction

In recent years, the development and study of Artificial Intelligence (AI) models for onboard Earth Observation (EO) satellites has gained significant interest, driven by the emergence of missions equipped with AI-enabled payloads such as ESA's ΦSat missions and EnduroSat's Balkan constellation [12, 19, 24]. Relevant researches usually target high-level vision tasks e.g., object detection, segmentation, classification, etc.. [16, 22, 37]. Such works implicitly assume that the input data are already in a processed and usable form, commonly identified as Level-2 (L2). However, in a real-working scenario, raw sensor acquisitions, known as Level-0 (L0), necessitate lengthy and complex low-level processing steps, such as denoising (of both gaussian and stripe noise), radiometric corrections, coregistration, orthorectification (when applicable), and numerous other steps before they can reliably support downstream tasks or be used for any analysis.

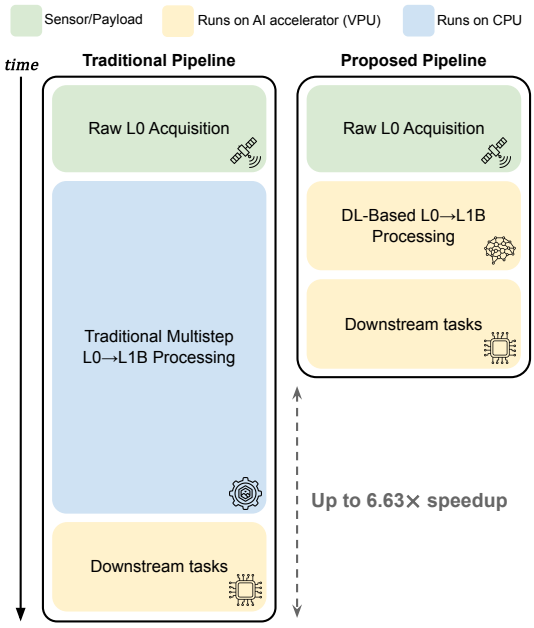

Figure 1. High-level comparison between the traditional onboard processing chain and our proposed alternative. In the baseline pipeline, raw Level-0 (L0) acquisitions are first converted into processed Level-1B (L1B) products through a conventional multi-step CPU preprocessing chain, and only then passed to downstream tasks (e.g., AI applications, analytics, or derived products) on the onboard AI accelerator (VPU). Our approach replaces the CPU preprocessing stage with an efficient DL-based L0→L1B pipeline designed to run directly on space-qualified VPU, reducing the end-to-end latency by minimizing the CPU-bound bottleneck.

At the same time, several recent works [7, 27] explicitly bypass the traditional preprocessing pipeline, operating directly on raw data products. Although promising, these approaches are not always feasible in real mission settings. First, raw data are often mission- and sensor-specific, and paired datasets of raw images and corresponding labels are rare since labeling raw imagery is often non-trivial: sensor artifacts, non-uniform responses, and noise patterns can

obscure semantic content and make annotations ambiguous or inconsistent, requiring expert knowledge and additional calibration steps before labels become trustworthy. Second, many applications still require processed products to ensure interpretability and compatibility with established workflows. As a result, the ability to produce processed, standardized products remains a key requirement in both operational and research environments.

In missions equipped with onboard AI accelerators (e.g., VPUs), most of the available compute is typically devoted to downstream inference, while the raw-to-processed preprocessing pipeline still runs on CPU. As a result, the AI accelerator often remains underutilized during the time-consuming low-level processing stage. This observation motivates exploring whether a learned, lightweight model can approximate the core visual restoration steps of the preprocessing pipeline and run directly on the onboard accelerator, reducing latency and relieving the CPU bottleneck. Despite its relevance, to the best of our knowledge, the idea of using Deep Learning (DL) to perform an end-to-end onboard raw-to-processed pipeline, i.e., replacing the conventional CPU preprocessing chain with a learned model that maps raw products to final processed image products, has not been studied. This gap is likely due to the practical challenges of deploying such models under satellite constraints, including limited compute, memory, and power budgets, as well as the need to preserve product fidelity and compatibility with established processing standards. These limitations are particularly stringent for CubeSats and nanosatellites.

Under this working-case scenario, the ΦSat-2 represents an ideal benchmark platform for studying onboard AI processing, as this satellite has been specifically engineered to support onboard AI workloads through dedicated hardware accelerators (i.e., Intel Movidius Myriad 2 VPU). More in detail, to enable any of the planned AI onboard applications, the satellite still requires upgrading the product level from raw acquisitions to processed, higher-level products (depending on the specific application), with an end-to-end latency of roughly 19–29 minutes for a 4096×4096 image, using the onboard CPU. Acquisition/readout and preprocessing dominate the pipeline, accounting for approximately 60–83% of the total runtime, highlighting low-level processing as a primary bottleneck [25, 26]. Figure 1 summarizes this bottleneck by contrasting the traditional CPU-based preprocessing multistep pipeline with our DL alternative, highlighting the expected reduction in end-to-end latency. These consideration motivate the following question:

can we reconstruct processed products from raw acquisitions with a learned model that is fast and efficient enough for deployment on space-qualified AI hardware?

Addressing this question requires not only reconstruction accuracy, but also careful design of the pipeline under strict memory, compute, and runtime constraints. Importantly, the goal of onboard reconstruction is to produce products that are interpretable and semantically consistent to support onboard analysis and inference, rather than matching the highest-quality products typically generated on the ground.

Consequently, in this work, we propose a DL-based raw-to-processed processing pipeline that reproduces the ΦSat-2 [24, 25] processing chain from raw L0 products to L1B[1] products, covering all main visual restoration steps while excluding georeferencing (which we leave to the conventional pipeline, as our focus is on the image formation and artifact removal stage). Our approach is composed of two learned modules; first, a custom coregistration network aligns the raw multispectral L0 spectral bands by predicting for each band a pair of $(\Delta x, \Delta y)$ shifts. Second, an image reconstruction model (NAFNet [3]) reconstructs the target L1B products, learning to correct sensor-dependent artifacts and noise. To favor efficiency, we adopt lightweight architectures with a reduced number of floating point operations (FLOPs) and parameters, enabling fast inference on the PhiSat-2's AI accelerator. In addition, to further improve efficiency, we operate in the wavelet domain. The wavelet decomposition effectively reduces the spatial resolution of the representation, allowing the majority of convolutions to operate on smaller feature maps, thereby substantially decreasing both computational and memory consumption. Beyond efficiency, the wavelet transform is also a powerful tool for decoupling noise from signal across subbands [2, 5, 32], making noise patterns easier to identify and suppress during reconstruction. In summary, the main contributions of this work are following reported.

- We introduce an end-to-end DL pipeline for ΦSat-2 L0→L1B reconstruction, explicitly tailored for onboard execution, which combines a learned band-coregistration module with a wavelet-domain image reconstruction module that operates at reduced effective spatial resolution and improves noise–signal decoupling across subbands.
- We quantitatively demonstrated that the proposed design achieves up to a 6.63% latency boost on the Intel Movidius Myriad 2 space-qualified accelerator while maintaining reconstruction fidelity. Beyond reconstruction quality, we also assess functional consistency in a relevant downstream application by evaluating the Normalized Difference Vegetation Index (NDVI) calculated from the reconstructed products.
- We analyze tradeoffs via ablations on coregistration,

[1] Importantly, for ΦSat-2, the L1B product corresponds to the final processed output delivered by the mission, i.e., following common nomenclature, L1B products are radiometrically corrected but not orthorectified; therefore, they do not reach Level-2.

wavelet representation and model complexity.

The remainder of the paper is organized as follows: Section 2 briefly describes and reviews related works, Section 3 describes the proposed pipeline, the modules, and the training approach, Section 4 reports experimental results and ablations, and Section 5 concludes with a discussion of limitations, possible improvements and future directions.

2. Related Work

2.1. Onboard DL for EO

The rapid development and growing interest in edge AI has led the scientific community to investigate whether DL models can be effectively deployed onboard satellites [4, 8]. The main goals are to enable real-time, low-latency processing, reduce downlink bandwidth, and support more autonomous operations. Many works focus on high-level vision tasks such as object detection [28, 37], change detection [17, 33], and semantic segmentation [9, 11]. While several works address low-level vision tasks, including data compression [1, 14], super-resolution [30], and image denoising [6, 38], they remain relatively limited compared to high-level applications and are still largely underexplored. However, most of these approaches mainly focus on proposing lightweight models and efficient implementations, but they typically assume that the input imagery is already processed and clean.

2.2. Wavelet-Based Processing

Remote sensing images are often affected by a significant amount of noise, including both gaussian and stripe noise, which makes denoising and artifact removal a central step in many processing pipelines. Wavelet transforms [23] have long been used in traditional signal processing for this purpose, as they provide a multi-scale decomposition that is effective for denoising [2, 5, 32], compression [34, 35] and many other tasks [20, 23, 36]. Building on these properties, recent DL approaches have increasingly explored operating in the wavelet domain rather than in the spatial (i.e., pixel) domain, either by directly processing wavelet coefficients or by integrating wavelet representations into neural networks. For instance, some studies [10, 13] work in the wavelet domain to remove stripe noise from infrared images, while other works [21, 31, 39] leverage wavelet features to improve performance on various downstream tasks. More recently, wavelets have also been adopted in generative modeling, including diffusion methods [15, 29, 40]. Overall, these results suggest that working in the wavelet domain can be beneficial both for efficiency and for better decoupling noise from signal, making wavelet-based representations a natural choice for learned restoration pipelines.

3. Methodology

The proposed onboard processing pipeline is shown in Figure 2, which summarizes the main processing stages, i.e., registration and reconstruction. In this section, we first provide a brief overview of the dataset. We then describe the two main components of the pipeline, band registration and image reconstruction, detailing the corresponding models and their pre-training strategies. Finally, we present the complete end-to-end architecture and discuss the joint fine-tuning procedure.

3.1. Data and Problem Setup

To train and evaluate our pipeline, we rely on a publicly available dataset[2] composed by paired ΦSat-2 L0–L1B data. The dataset is composed by a total of 74 acquisitions of size $4096 \times (4096 \cdot k)$ pixels, i.e., each acquisition has fixed width 4096 and height equal to an integer multiple of 4096. Due to the large spatial extent of the original acquisitions, we process the original data by firstly cropping each acquisition into non-overlapping 512×512 patches. We then perform a clean-up step, discarding patches with missing/invalid pixels or corrupted content. Importantly, we further apply a co-registration based on phase correlation to the raw L0 patches, ensuring that the L0 bands are aligned to a common spatial reference prior to training and evaluation. Building on this preprocessed subset, we build two dataset variants. The first one, *OCDataset* (Original Coregistration Dataset), contains the original L1B products, i.e., patches generated with the standard ΦSat-2 coregistration pipeline. The second variant, *PCCDataset* (Phase Correlation Coregistration Dataset), is derived from *OCDataset* by (i) filtering out low-quality samples (e.g., extremely noisy patches, near-empty regions, or uniform artifacts) from the training split, and (ii) re-coregistering the L0–L1 pairs with phase correlation to enforce a pixel-wise correspondence between the two levels. We employ *PCCDataset* both for training the registration module and for reconstruction experiments requiring pixel-wise aligned pairs, while *OCDataset* is used to assess performance under the original onboard coregistration conditions.

3.2. Band Registration

Registration Network Since the L0 multispectral bands are not perfectly aligned during the acquisition phase, we first estimate and compensate for inter-band shifts before reconstruction. More in detail, given an L0 patch with 8 spectral bands, we select one band as the reference band (in our case the last band) and predict 2D shifts $(\Delta x, \Delta y)$ for each of the remaining 7 bands. We use a lightweight shift regressor, shown in Figure 3, that takes as input a pair of single-band images (target band and reference band,

[2] placeholder_link

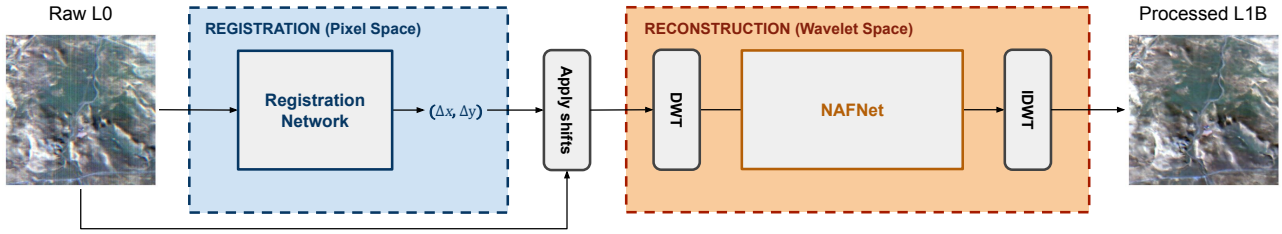

Figure 2. Overall L0→L1B pipeline; it is composed of two main stages. First, a registration module in pixel space predicts per-band offsets $(\Delta x, \Delta y)$ through a custom registration network, and the estimated shifts are applied to align the multispectral bands. Second, the aligned L0 images, are mapped to the wavelet domain using the discrete wavelet transform (DWT) and processed by a NAFNet reconstruction backbone. The reconstruced wavelet representation is then converted back to the image domain using the inverse transform (IDWT). For better visualization, we display only the RGB bands for both the L0 and L1B images.

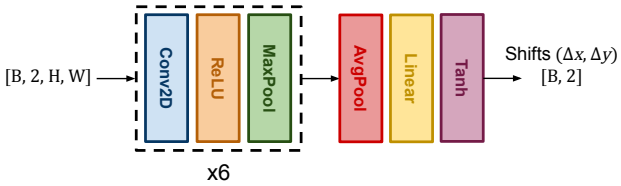

Figure 3. Shift regressor architecture used in the registration module. The network takes as input a target–reference pair stacked as a 2-channel tensor $[B, 2, H, W]$ and regresses the corresponding translation $(\Delta x, \Delta y) \in \mathbb{R}^{B\times 2}$. It consists of a repeated Conv2D–ReLU–MaxPool block, followed by average pooling and a linear layer. A $\tanh(\cdot)$ activation bounds the predicted shifts within a maximum range. The same regressor is used for both the coarse and fine stages in the coarse-to-fine registration scheme.

stacked as a 2-channel tensor) and outputs the corresponding shift. The core network is intentionally small and it aggressively downsamples the input through a sequence of convolution and pooling layers, followed by global average pooling and a linear head that regresses the 2D displacement. To keep the predicted shifts bounded in an interval of reasonable shifts, the final output is scaled with a $\tanh(\cdot)$ non-linearity and multiplied by a maximum displacement parameter. Additionally, to improve robustness across a broader range of misalignments, we adopt a coarse-to-fine strategy. Our registration strategy is composed of two steps: a first shift regressor predicts a coarse shift within a larger range, which is then applied to the target band using an integer translation. A second shift regressor predicts a fine correction within a small range, and the final shift is obtained by summing the two predictions. Then, for multispectral inputs, we implement the same core regressor for all bands by reshaping the input so that all (target, reference) pairs are processed in parallel.

Training Procedure We pre-train the registration network by applying random integer shifts to the *PCCDataset* patches, so that the ground truth displacements are known exactly. Starting from an L0 patch $x_{\mathrm{L0}} \in \mathbb{R}^{C\times H\times W}$, we independently shift each of the first $C-1$ bands by a random integer displacement $(\Delta x, \Delta y)$, while keeping the reference band (the last one) fixed.

Let $\mathbf{s}_c = (\Delta x_c, \Delta y_c) \in \mathbb{Z}^2$ denote the ground truth shift applied to spectral band c (with $\mathbf{s}_C = \mathbf{7}$ for the reference band), and $\hat{\mathbf{s}}_c$ the corresponding prediction. The network is trained to regress the applied shifts using an ℓ_1 loss,

$$\mathcal{L}_{\mathrm{reg}} = \frac{1}{C}\sum_{c=0}^{C-1} \left\|\hat{\mathbf{s}}_c - \mathbf{s}_c\right\|_1 . \tag{1}$$

After estimating the shifts, we obtain the co-registered raw L0 image by shifting each band accordingly, i.e.,

$$\bar{x}_{\mathrm{L0}} = \mathcal{T}(x_{\mathrm{L0}}; \hat{\mathbf{s}}) , \tag{2}$$

where $\mathcal{T}(\cdot)$ denotes per-band shift using the predicted $\hat{\mathbf{s}} = \{\hat{\mathbf{s}}_c\}_{c=0}^{C-1}$.

Importantly, we perform this pre-training in the spatial domain (i.e., pixel space) to minimize registration errors, since accurate band alignment is crucial for the subsequent reconstruction stage. This choice is also consistent with the actual ΦSat-2 onboard pipeline, where coregistration is implemented by applying integer shifts to all bands except the last one. For this reason, we keep the original spatial resolution during registration, which directly supports integer-valued shift regression and avoids the loss of spatial precision that would arise when operating at reduced resolution (e.g., in the wavelet domain).

3.3. Image Reconstruction

Reconstruction Network The image reconstruction model is based on the well-known NAFNet [3] architecture. The latter model is a U-Net-like network that has been successfully applied to a wide range of low-level vision tasks, including denoising, super-resolution, and

deblurring. A key aspect of such architecture is its strong computational efficiency. More in detail, it is designed to achieve high restoration quality with a lightweight backbone, avoiding non-linear activation functions and relying instead on simple operations such as linear transforms and multiplications. These properties make NAFNet particularly suitable for our onboard setting, where compute and memory budgets are limited.

Wavelet-Based Formulation To reduce computational complexity, we formulate the reconstruction task in the wavelet domain (please refer to Figure 4). Given an input patch $x \in \mathbb{R}^{B\times C\times H\times W}$, we apply a single-level 2D discrete wavelet transform (DWT) independently to each spectral band. This produces yields one low-frequency subband (LL) and three high-frequency subbands (LH, HL, HH), each at half spatial resolution. The four resulting subbands are then concatenated along the channel dimension, producing the compact multi-scale representation $\mathcal{W}(x) = \tilde{x} \in \mathbb{R}^{B\times 4C\times \frac{H}{2}\times \frac{W}{2}}$, ordered as $[\mathrm{LL}, \mathrm{LH}, \mathrm{HL}, \mathrm{HH}]$ for each band. The reconstruction model $f_\theta(\cdot)$ is then trained to map the wavelet coefficients of the L0 input to the wavelet coefficients of the target L1B product, i.e., $\hat{y} = f_\theta(\tilde{x}_{\mathrm{L0}}), \quad y = \tilde{x}_{\mathrm{L1B}}$. Beyond efficiency, this formulation helps decoupling the signal and noise across subbands, making structured artifacts (e.g. stripes) easier to model and suppress. Figure 4 highlights how stripe and gaussian noise become more separable in the wavelet domain: in the L0 sample, vertical striping concentrates in the HL subband, while it is attenuated in the processed L1B product.

Training Procedure We pre-train the reconstruction model on paired L0–L1B patches from the *PCCDataset*, where the pairs are already co-registered; consequently, no additional registration is required during training. Given a paired sample $(x_{\mathrm{L0}}, x_{\mathrm{L1B}})$, we compute their wavelet representations $\tilde{x}_{\mathrm{L0}} = \mathcal{W}(x_{\mathrm{L0}})$ and $\tilde{x}_{\mathrm{L1B}} = \mathcal{W}(x_{\mathrm{L1B}})$, and we predict $\hat{y} = f_\theta(\tilde{x}_{\mathrm{L0}})$ with target $y = \tilde{x}_{\mathrm{L1B}}$. The reconstruction objective is defined as

$$\mathcal{L}_{\mathrm{rec}}(\hat{y}, y) = \mathcal{L}_{\ell_1}(\hat{y}, y) + \alpha\, \mathcal{L}_{\mathrm{edge}}(\hat{y}, y) \tag{3}$$

where $\mathcal{L}_{\ell_1}(\hat{y}, y) = \|\hat{y} - y\|_1$ is a standard element-wise ℓ_1 loss. The scalar α weights an additional edge-aware term, which we include to better preserve structural content while discouraging stripe-like artifacts. Specifically, we define $\mathcal{L}_{\mathrm{edge}}$ via gradient matching:

$$\begin{aligned}\mathcal{L}_{\mathrm{edge}}(\hat{y}, y) = 0.1 \left\|\nabla_x \mathcal{G}(\hat{y}) - \nabla_x \mathcal{G}(y)\right\|_1 \\ + \left\|\nabla_y \mathcal{G}(\hat{y}) - \nabla_y \mathcal{G}(y)\right\|_1\end{aligned} \tag{4}$$

where ∇_x, ∇_y denote horizontal and vertical image derivatives (implemented with Sobel filters) and $\mathcal{G}(\cdot)$ is a light Gaussian smoothing operator. In our setting, stripe noise is predominantly vertical; therefore, we adopt a fixed, stripe-oriented variant of $\mathcal{L}_{\mathrm{edge}}$, while α remains the only tunable weight controlling its overall contribution.

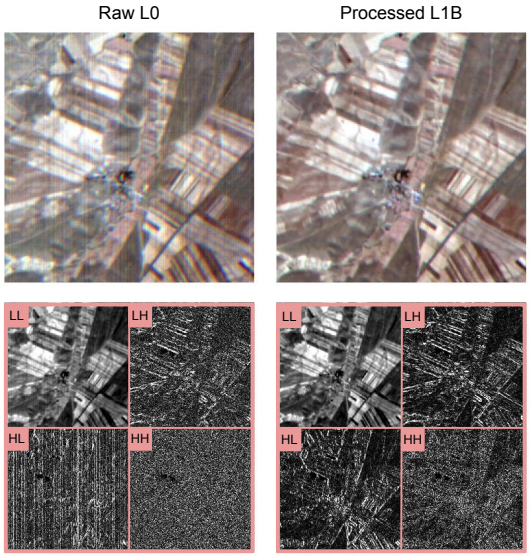

Figure 4. Pixel vs. wavelet domain visualization. Top: RGB patch in pixel space, raw L0 (left) and processed L1B (right). Bottom: single-level 2D wavelet decomposition into four subbands: LL contains the low-frequency content, while LH, HL, and HH capture high-frequency details at different orientations. The stronger striping in L0 is clearly visible as a periodic vertical pattern concentrated in the HL subband, while the gaussian noise is mainly captured in the HH subband.

3.4. End-to-End Pipeline

The complete end-to-end pipeline is illustrated in Figure 2. Starting from a raw L0 image $x_{\mathrm{L0}} \in \mathbb{R}^{C\times H\times W}$, the pipeline consists of three main stages. First, the registration module estimates shifts and aligns the C spectral bands to a common reference frame, producing a co-registered raw patch $\bar{x}_{\mathrm{L0}}$. Second, we apply a single-level 2D discrete wavelet transform $\mathcal{W}(\cdot)$ independently to each band, obtaining a compact wavelet representation $\tilde{x}_{\mathrm{L0}} = \mathcal{W}(\bar{x}_{\mathrm{L0}}) \in \mathbb{R}^{4C\times \frac{H}{2}\times \frac{W}{2}}$. Finally, the reconstruction network f_θ operates directly on wavelet coefficients to predict the corresponding processed product in the wavelet domain, $\hat{y} = f_\theta(\tilde{x}_{\mathrm{L0}})$, which is then mapped back to the image domain via the inverse transform $\mathcal{W}^{-1}(\cdot)$ to obtain the reconstructed L1B estimate $\hat{x}_{\mathrm{L1B}} = \mathcal{W}^{-1}(\hat{y})$.

Training Procedure To train the full end-to-end pipeline, we initialize both modules with their corresponding pre-trained weights. We then fine-tune both models on *PCCDataset* samples for which ground truth band shifts are known. Given a training pair $(x_{\mathrm{L0}}, x_{\mathrm{L1B}})$ and the ground

truth shifts $\mathbf{s}$, the registration network predicts $\hat{\mathbf{s}}$, which are applied to obtain the co-registered raw input $\bar{x}_{\mathrm{L0}} = \mathcal{T}(x_{\mathrm{L0}}; \hat{\mathbf{s}})$. The reconstruction network then operates on wavelet coefficients, yielding $\hat{x}_{\mathrm{L1B}} = \mathcal{W}^{-1}\big(f_\theta(\mathcal{W}(\bar{x}_{\mathrm{L0}}))\big)$.

The fine-tuning total loss jointly optimizes registration accuracy and reconstruction quality via a weighted combination of the two losses:

$$\mathcal{L} = \beta\, \mathcal{L}_{\mathrm{rec}} + (1-\beta)\, \mathcal{L}_{\mathrm{reg}}, \tag{5}$$

where $\mathcal{L}_{\mathrm{rec}}$ is the reconstruction loss 3 (computed between $\hat{y} = f_\theta(\mathcal{W}(\bar{x}_{\mathrm{L0}}))$ and $y = \mathcal{W}(x_{\mathrm{L1B}})$), and $\mathcal{L}_{\mathrm{reg}}$ is the registration loss 1 (computed between $\hat{\mathbf{s}}$ and $\mathbf{s}$). Here, $\beta \in [0, 1]$ controls the trade-off between reconstruction fidelity and alignment accuracy during end-to-end training.

4. Results

In this section, we report quantitative and qualitative results for the proposed end-to-end raw-to-processed pipeline, as well as for its two individual modules (registration and reconstruction). We then analyze computational complexity and runtime on the Intel Movidius Myriad 2 VPU, and conclude with additional ablation studies.

4.1. Implementation Details

All custom models and experiments were implemented in PyTorch. Unless otherwise stated, we use the original NAFNet implementation without architectural modifications[3] Training is performed on a single NVIDIA A40 GPU (48 GB). To assess deployment feasibility on space-qualified hardware, we additionally benchmark inference on the Intel Movidius Myriad 2 VPU, i.e., the same accelerator used onboard ΦSat-2. For all training runs, we adopt the Adam optimizer [18] with a fixed learning rate of 3×10^{-4}. Unless otherwise stated, we operate in the wavelet domain using a single-level 2D DWT with Haar wavelet. When losses are computed in the wavelet domain, we report metrics by mapping predictions back through the inverse transform in the image domain.

For experiments requiring misaligned inputs (i.e., without co-registration), we manually misalign the data by applying random integer shifts to the input bands during training. During the evaluation, we apply a fixed and deterministic set of shifts to the same images to ensure consistency and fair comparability between different experiments.

4.2. End-to-End Pipeline Results

We evaluate the complete pipeline on both *OCDataset* and *PCCDataset*. Figure 5 shows representative qualitative examples for L1B and NDVI reconstruction, while Tables 1 and 2 report the corresponding quantitative results.

[3] placeholder_link

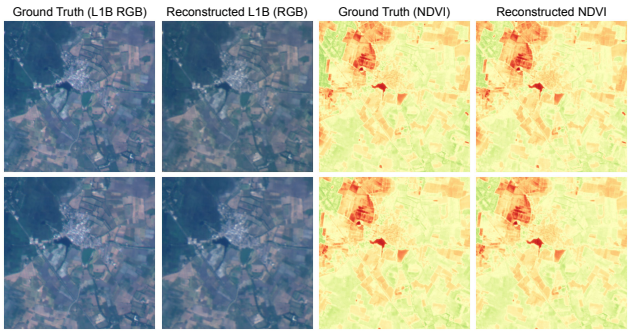

Figure 5. Qualitative end-to-end results for L1B reconstruction (RGB rendering) and NDVI. Top: sample from *OCDataset*. Bottom: same sample from *PCCDataset*.

Overall, the pipeline achieves substantially higher scores on *PCCDataset* (Table 1). This is expected, since our registration module is trained to mimic the phase correlation coregistration used to build *PCCDataset*. In contrast, *OCDataset* employs the original onboard coregistration, which does not perfectly align with our learned registration. This domain gap mainly affects the upstream band alignment, and propagates to both L1B reconstruction and NDVI, justifying the obtained lower metrics on *OCDataset*.

Additionally, the band-wise analysis reported in Table 2 further clarifies the behavior of the registration stage. Band 7 has a zero registration error by construction, as it serves as the fixed reference and is never shifted. For the remaining bands, the MAE increases as their spectral content diverges from the NIR reference. In particualr, the largest displacement errors are observed for visible bands (1–3), which have radiometric features that significantly differ from band 7. This cross-spectral mismatch inherently complicates the shift estimation, and this effect is consistently reflected in the per-band reconstruction metrics.

Importantly, we benchmark latency on the Intel Movidius Myriad 2. In our setup, the proposed pipeline runs almost entirely on the accelerator (with the exception of the integer shift application, which is executed on CPU) and requires on average 3.58 minutes per full 4096×4096 image. On average, the reconstruction module accounts for 2.04 minutes, while the registration module requires only 1.34 minutes; the remaining time is due to DWT/IDWT and additional I/O and runtime overhead. This compares favorably against the traditional multi-step onboard processing chain, where the preprocessing stage takes roughly 740–1425 seconds (i.e., 12.35–23.75 minutes) depending on the target processing level [25, 26], yielding an overall speed-up of approximately $3.5\times$ to $6.6\times$.

4.3. Reconstruction Ablations

In this subsection, we study how key training choices affect the reconstruction network, focusing on data augmentation,

Table 1. Quantitative performance of the end-to-end pipeline on L1B reconstruction and NDVI reconstruction on *OCDataset* and *PCCDataset*. MAE shifts are computed w.r.t. the reference band (band 7) when ground truth shifts are available.

	L1 Reconstruction				**NDVI Reconstruction**		
Dataset	**PSNR**	**SSIM**	**MS-SSIM**	**MAE shifts** $(\Delta x, \Delta y)$	**PSNR**	**SSIM**	**MS-SSIM**
OCDataset	28.05 dB	86	90.87	N.A.	32.76 dB	85.27	93.20
PCCDataset	30.49 dB	91.23	95.86	(1.46, 1.61) px	36.30 dB	90.05	96.19

Table 2. Band-wise L1B reconstruction performance and registration error. MAE shifts are reported w.r.t. the reference band, hence band 7 has zero shift by definition.

	OCDataset			*PCCDataset*			
Band	**PSNR**	**SSIM**	**MS-SSIM**	**PSNR**	**SSIM**	**MS-SSIM**	**MAE shifts** $(\Delta x, \Delta y)$
0	30.03 dB	92.77	96.30	32.83 dB	95.92	97.89	(1.35, 1.16) px
1	28.17 dB	89.70	89.28	32.10 dB	94.24	95.93	(2.18, 2.64) px
2	29.13 dB	87.04	90.86	31.30 dB	91.92	95.41	(1.68, 1.82) px
3	29.24 dB	83.14	90.91	30.52 dB	88.15	94.61	(1.64, 1.73) px
4	28.18 dB	83.27	90.48	30.13 dB	87.40	95.55	(1.46, 1.72) px
5	27.73 dB	77.95	86.53	30.15 dB	87.03	94.61	(0.95, 1.16) px
6	27.05 dB	74.71	82.87	30.16 dB	86.50	94.22	(0.94, 1.11) px
7	33.66 dB	96.21	98.83	35.28 dB	97.95	99.35	(0, 0) px

Table 3. Edge loss during reconstruction training.

Edge Loss	**PSNR**	**SSIM**
–	31.81 dB	95.22
✓	32.25 dB	95.40

Table 4. Geomtric augmentations during reconstruction training.

Augmentations applied	**PSNR**	**SSIM**
Random Rotations + Flips	32.25 dB	95.40
Flips	33.87 dB	97.07

loss design and comparing the training of the registration network in the wavelet and image domain.

Edge Loss Ablation Table 3 reports the effect of enabling the proposed edge loss term (Equation 4) in the reconstruction loss. More in detail, the addition of the $\mathcal{L}_{\text{edge}}$ term yields a consistent improvement, increasing PSNR from 31.81 dB to 32.25 dB and SSIM from 95.22 to 95.40. This suggests that gradient matching helps preserve structural transitions while reducing residual striping, complementing the pixel-wise $\mathcal{L}_{\ell_1}$ loss.

Data Augmentations Ablation Since EO imagery represents physical measurements, we avoid photometric augmentations (e.g., color jitter, contrast changes) that would alter the radiometric meaning of the signal. Therefore, we restrict augmentation to geometric transforms that preserve the underlying measurements. Table 4 reports the quantitative reconstruction performance under two augmentation strategies: (i) random rotations, and (ii) horizontal/vertical flips only.[4] As shown in Table 4, training with flips only yields a substantial improvement (+1.62 dB PSNR and +1.67 SSIM), which is also reflected qualitatively in Figure 6. We attribute this improvement to the vertical orientation of the stripe noise. Applying horizontal flips preserves this direction, allowing the model to learn a stable stripe prior. In contrast, random rotations change the stripe direction across samples. This prevents the model from consistently associating vertical edges with noise, which ultimately degrades destriping performance.

Wavelet vs Image Domain We ablate the training domain of the reconstruction network by comparing the results of the reconstruction network in the wavelet domain (loss computed directly on DWT coefficients as shown in Section 3) against the same network trained on the image domain (i.e., loss computed on reconstructed images), while keeping the architecture and training protocol fixed. Since

[4] We do not consider other geometric transforms (e.g., arbitrary rotations or scaling) since they would require resampling and would effectively modify the spatial resolution and spectral content of the data.

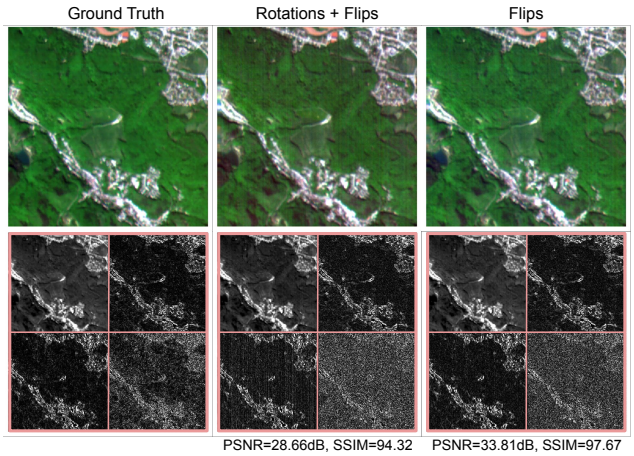

Figure 6. Qualitative ablation on geometric augmentation for the reconstruction network. Using only vertical and horizontal flips yields visibly stronger destriping: the periodic vertical stripe pattern is reduced, as visible in the HL subband.

Figure 7. Mean value per column averaged over *PCCDataset*. The wavelet-domain reconstruction follows the processed L1B (ground truth) profile more closely than the image-domain variant, indicating improved suppression of vertical striping.

striping is predominantly vertical, it induces a characteristic column-wise bias that can be analyzed by averaging pixel values along rows. Figure 7 reports the mean value per image column, further averaged over *PCCDataset*, for the processed L1B target and the two reconstructions. The wavelet-domain model closely tracks the L1B profile across the entire swath, whereas the image-domain model exhibits larger deviations and a more pronounced column-dependent drift. This indicates that operating in the wavelet domain facilitates suppression of vertical striping, yielding a reconstruction whose large-scale column statistics better match the processed L1B product.

5. Discussion

The proposed pipeline significantly accelerates L1B product generation by leveraging inference on an AI accelerator. Despite these promising results, several limitations remain and provide directions for future work. First, registration accuracy is not yet optimal (Tables 1 and 2) and likely represents the main bottleneck of the pipeline. Reducing the residual misalignment (currently around 1.5 px on average) would directly improve reconstruction quality and downstream metrics. One possible approach is to move beyond a single fixed reference band and adopt a cascaded (or pairwise) strategy, where each band is registered against a spectrally closer band. This would reduce cross-spectral appearance differences and could yield more reliable shift estimates.

A second limitation concerns residual denoising artifacts, especially for Gaussian noise (see Figure 6). In particular, the HH subband (bottom-right) reveals a noticeable salt-and-pepper texture compared to the ground truth, indicating that high-frequency noise is not fully suppressed.

Additionally, our empirical analysis revealed that one spectral band (the blue band, i.e., 1) is consistently more blurred than the others. While this issue is not directly addressed by the proposed pipeline, it suggests an interesting complementary direction for future work: incorporating a dedicated deblurring stage to recover high-frequency content in the blue channel. This could further improve both visual quality and downstream products.

Finally, our pipeline focuses on the visual L0→L1B restoration steps and does not address georeferencing. Extending the approach to produce a fully georeferenced product is an interesting direction, but would likely require incorporating ancillary metadata (e.g., attitude/orbit information) and potentially a multimodal formulation.

6. Conclusion

We presented a DL-based raw-to-processed (L0 $\rightarrow$ L1B) onboard processing pipeline tailored for onboard constraints. The method combines a lightweight multispectral band-registration module (integer per-band shifts w.r.t. a fixed reference) with an efficient wavelet-domain NAFNet reconstruction backbone, enabling fast restoration while reducing compute through a single-level DWT/IDWT formulation. Experiments on *OCDataset* and *PCCDataset* show that the pipeline reconstructs both L1B imagery and derived NDVI with good fidelity. Importantly, benchmarking on the Intel Movidius Myriad 2 VPU shows an average latency of 3.58 minutes for a 4096×4096 image, corresponding to a $3.5\times$ to $6.6\times$ speed-up over the traditional ΦSat-2 CPU-based preprocessing chain. Overall, these results suggest that learned raw-to-processed processing is a viable path toward faster and more autonomous in-orbit generation of usable EO products for downstream analytics and AI applications.

References

[1] Vinicius Alves de Oliveira, Marie Chabert, Thomas Oberlin, Charly Poulliat, Mickael Bruno, Christophe Latry, Mikael Carlavan, Simon Henrot, Frederic Falzon, and Roberto Camarero. Reduced-complexity end-to-end variational autoencoder for on board satellite image compression. *Remote Sensing*, 13(3):447, 2021. 3

[2] GY Chen, Tien D Bui, and Adam Krzyzak. Image denoising using neighbouring wavelet coefficients. *Integrated Computer-Aided Engineering*, 12(1):99–107, 2005. 2, 3

[3] Liangyu Chen, Xiaojie Chu, Xiangyu Zhang, and Jian Sun. Simple baselines for image restoration. In *European conference on computer vision*, pages 17–33. Springer, 2022. 2, 4

[4] Bharadwaj Chintalapati, Arthur Precht, Sougata Hanra, Rene Laufer, Marcus Liwicki, and Jens Eickhoff. Opportunities and challenges of on-board ai-based image recognition for small satellite earth observation missions. *Advances in space research*, 75(9):6734–6751, 2025. 3

[5] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on image processing*, 16(8):2080–2095, 2007. 2, 3

[6] Vinicius Alves de Oliveira, Marie Chabert, Thomas Oberlin, Charly Poulliat, Mickael Bruno, Christophe Latry, Mikael Carlavan, Simon Henrot, Frederic Falzon, and Roberto Camarero. Satellite image compression and denoising with neural networks. *IEEE Geoscience and Remote Sensing Letters*, 19:1–5, 2022. 3

[7] Roberto Del Prete, Gabriele Meoni, Manuel Salvoldi, Domenico Barretta, Maria Daniela Graziano, Nicolas Longépé, and Alfredo Renga. Enhanced maritime monitoring via onboard processing of raw multi-spectral imagery by deep learning. In *IGARSS 2024-2024 IEEE International Geoscience and Remote Sensing Symposium*, pages 1713–1717. IEEE, 2024. 1

[8] Lorenzo Diana and Pierpaolo Dini. Review on hardware devices and software techniques enabling neural network inference onboard satellites. *Remote Sensing*, 16(21):3957, 2024. 3

[9] Lorenzo Diana, Jia Xu, and Luca Fanucci. Oil spill identification from sar images for low power embedded systems using cnn. *Remote Sensing*, 13(18):3606, 2021. 3

[10] Zeshan Fayyaz, Daniel Platnick, Hannan Fayyaz, and Nariman Farsad. Deep unfolding for iterative stripe noise removal. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE, 2022. 3

[11] Tharindu Fernando, Clinton Fookes, Harshala Gammulle, Simon Denman, and Sridha Sridharan. Toward on-board panoptic segmentation of multispectral satellite images. *IEEE Transactions on Geoscience and Remote Sensing*, 61: 1–12, 2023. 3

[12] Gianluca Giuffrida, Luca Fanucci, Gabriele Meoni, Matej Batič, Léonie Buckley, Aubrey Dunne, Chris Van Dijk, Marco Esposito, John Hefele, Nathan 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*, 60:1–14, 2021. 1

[13] Juntao Guan, Rui Lai, and Ai Xiong. Wavelet deep neural network for stripe noise removal. *IEEE Access*, 7:44544–44554, 2019. 3

[14] Giorgia Guerrisi, Fabio Del Frate, and Giovanni Schiavon. Artificial intelligence based on-board image compression for the φ-sat-2 mission. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 16:8063–8075, 2023. 3

[15] Yi Huang, Jiancheng Huang, Jianzhuang Liu, Mingfu Yan, Yu Dong, Jiaxi Lv, Chaoqi Chen, and Shifeng Chen. Wavedm: Wavelet-based diffusion models for image restoration. *IEEE Transactions on Multimedia*, 26:7058–7073, 2024. 3

[16] Gabriele Inzerillo, Diego Valsesia, and Enrico 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. 1

[17] Gabriele Inzerillo, Diego Valsesia, Aniello Fiengo, and Enrico Magli. Compress-align-detect: onboard change detection from unregistered images. *arXiv preprint arXiv:2507.15578*, 2025. 3

[18] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. 6

[19] A Kolev, I Ivanov, I Nedelev, and V Danchev. Endurosat one-the first bulgarian cubesat mission. In *AIP Conference Proceedings*, page 160010. AIP Publishing LLC, 2019. 1

[20] Hui Li, BS Manjunath, and Sanjit K Mitra. Multisensor image fusion using the wavelet transform. *Graphical models and image processing*, 57(3):235–245, 1995. 3

[21] Qiufu Li, Linlin Shen, Sheng Guo, and Zhihui Lai. Wavelet integrated cnns for noise-robust image classification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7245–7254, 2020. 3

[22] Xiaojie Ma, Kefeng Ji, Boli Xiong, Linbin Zhang, Sijia Feng, and Gangyao Kuang. Light-yolov4: An edge-device oriented target detection method for remote sensing images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 14:10808–10820, 2021. 1

[23] Stephane G Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE transactions on pattern analysis and machine intelligence*, 11(7):674–693, 2002. 3

[24] N Melega, Bernardo A Carnicero, N Longepe, A Paskeviciute, V Marchese, Oriol Aragon, Irina Babkina, Alessandro Marin, Jakub Nalepa, Leonie Buckley, et al. Implementation of the ϕsat-2 on board image processing chain. In *Sensors, systems, and next-generation satellites XXVII*, pages 264–276. SPIE, 2023. 1, 2

[25] Nicola Melega, Nicolas Longepe, Agne Paskeviciute, Valentina Marchese, Oriol Aragon, Irina Babkina, Alessandro Marin, Jakub Nalepa, Leonie Buckley, Giorgia Guerrisi, et al. Development and implementation of the ϕsat-2 mission. In *Small Satellites Systems and Services Symposium (4S 2024)*, pages 1244–1254. SPIE, 2025. 2, 6

[26] Nicola Melega, Agnė Paškevičiūtė-Kidron, Nicolas Longepe, Mirko Albani, Louis Dadswell, George Hine,

Jacob Bullard, , and Alessandro Marin. In-orbit commissioning results of the esa sat-2 mission. In *Small Satellite Conference*, 2025. 2, 6

[27] Gabriele Meoni, Roberto Del Prete, Federico Serva, Alix De Beusscher, Olivier Colin, and Nicolas Longépé. Unlocking the use of raw multispectral earth observation imagery for onboard artificial intelligence. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 17:12521–12537, 2024. 1

[28] Yanhua Pang, Yamin Zhang, Qinglei Kong, Yi Wang, Bo Chen, and Xibin Cao. Socdet: A lightweight and accurate oriented object detection network for satellite on-orbit computing. *IEEE Transactions on Geoscience and Remote Sensing*, 61:1–15, 2023. 3

[29] Hao Phung, Quan Dao, and Anh Tran. Wavelet diffusion models are fast and scalable image generators. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10199–10208, 2023. 3

[30] Davide Piccinini, Diego Valsesia, and Enrico Magli. Onboard hyperspectral super-resolution with deep pushbroom neural network. *Remote Sensing*, 17(21):3634, 2025. 3

[31] Michaël Ramamonjisoa, Michael Firman, Jamie Watson, Vincent Lepetit, and Daniyar Turmukhambetov. Single image depth prediction with wavelet decomposition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11089–11098, 2021. 3

[32] Sylvain Sardy, Paul Tseng, and Andrew Bruce. Robust wavelet denoising. *IEEE transactions on signal processing*, 49(6):1146–1152, 2001. 2, 3

[33] Chahira Serief, Youcef Ghelamallah, and Youcef Bentoutou. Deep-learning-based system for change detection onboard earth observation small satellites. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 16:8115–8124, 2023. 3

[34] Jerome M Shapiro. Embedded image coding using zerotrees of wavelet coefficients. *IEEE Transactions on signal processing*, 41(12):3445–3462, 2002. 3

[35] David S Taubman, Michael W Marcellin, and Majid Rabbani. Jpeg2000: Image compression fundamentals, standards and practice. *Journal of Electronic Imaging*, 11(2): 286–287, 2002. 3

[36] Michael Unser. Texture classification and segmentation using wavelet frames. *IEEE Transactions on image processing*, 4(11):1549–1560, 1995. 3

[37] Yuan Yao, Zhiguo Jiang, Haopeng Zhang, and Yu Zhou. Onboard ship detection in micro-nano satellite based on deep learning and cots component. *Remote Sensing*, 11(7):762, 2019. 1, 3

[38] Ziyao Yi, Davide Piccinini, Diego Valsesia, Tiziano Bianchi, and Enrico Magli. Scalable neural pushbroom architectures for real-time denoising of hyperspectral images onboard satellites. *arXiv preprint arXiv:2601.05020*, 2026. 3

[39] Xiangyu Zhao, Peng Huang, and Xiangbo Shu. Wavelet-attention cnn for image classification. *Multimedia Systems*, 28(3):915–924, 2022. 3

[40] Junsheng Zhou, Weiqi Zhang, Baorui Ma, Kanle Shi, Yu-Shen Liu, and Zhizhong Han. Udiff: Generating conditional unsigned distance fields with optimal wavelet diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21496–21506, 2024. 3