POLITECNICO DI TORINO
Repository ISTITUZIONALE

Compress-Align-Detect: Onboard Change Detection From Unregistered Images

Original
Compress-Align-Detect: Onboard Change Detection From Unregistered Images / Inzerillo, G., Valsesia, D., Fiengo, A., Magli, E.. - In: IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING. - ISSN 0196-2892. - 64:(2026). [10.1109/TGRS.2026.3686514]

Availability:
This version is available at: 11583/3011429 since: 2026-05-27T07:24:57Z

Publisher:
IEEE

Published
DOI:10.1109/TGRS.2026.3686514

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)

17 November 2026

Compress-Align-Detect: onboard change detection from unregistered images

Gabriele Inzerillo, Diego Valsesia, *Member, IEEE*, Aniello Fiengo, and Enrico Magli, *Fellow, IEEE*

***Abstract*—Change detection from satellite images typically incurs a delay ranging from several hours up to days because of latency in downlinking the acquired images and generating orthorectified image products at the ground stations; this may preclude real or near real-time applications. To overcome this limitation, we propose shifting the entire change detection workflow onboard satellites. This requires simultaneously solving challenges in data storage, image registration, and change detection with a strict complexity constraint. In this paper, we present a novel and efficient framework for onboard change detection that addresses the aforementioned challenges in an end-to-end fashion with a deep neural network composed of three interlinked submodules: (1) image compression, tailored to minimize onboard data storage resources; (2) lightweight co-registration of non-orthorectified multi-temporal image pairs; and (3) a novel temporally invariant and computationally efficient change detection model. This is the first approach in the literature combining all these tasks in a single end-to-end framework under the constraints dictated by onboard processing. Experimental results compare each submodule with the current state of the art and evaluate the performance of the overall integrated system in a realistic setting on low-power hardware. Compelling change detection results are obtained in terms of F1-score as a function of compression rate, sustaining a throughput of 0.7 Mpixel/s on a 15 W accelerator.**

***Index Terms*—Onboard processing, efficient deep learning, change detection, image co-registration, image compression.**

I. Introduction

Change Detection (CD) has long been investigated in the field of remote sensing because of its fundamental importance in addressing problems such as disaster management, urban planning and monitoring, land cover usage, and much more.

Nowadays, satellites equipped with optical, multispectral or hyperspectral sensors continuously scan the Earth's surface, converting reflected sunlight into raw digital counts. Once received at the ground station, the data undergo a processing pipeline including radiometric calibration and geometric corrections to compensate for sensor and satellite attitude, and orthorectification to remove any distortion due to perspective and Earth's curvature.

Although this ground-based workflow is capable of producing very accurate results, it also incurs significant latency (from a few hours up to days) and is constrained by the need for high-bandwidth links to download all images. At the same time, there is a growing interest in the remote sensing community towards moving inference tasks directly onboard satellites. New-generation satellites are beginning to support lightweight deep learning (DL) models [1–3] for tasks such as segmentation and classification [4–6] to be performed directly in orbit. Onboard processing drastically reduces the data volume to be transmitted, alleviating bandwidth constraints and minimizing end-to-end latency, ultimately enabling truly real-time monitoring. Such capabilities are especially important in the CD domain for time-sensitive scenarios such as the rapid identification of natural disasters, leading to immediate damage assessment and recovery efforts.

However, onboard CD is still largely unexplored as it requires profound integration between multiple tasks and calls for novel frameworks in the way images are captured, stored, and processed onboard. In particular, three system-level tasks need to be addressed: image compression and storage, image registration and detecting change. First, the satellite needs to store and retrieve images acquired at multiple revisits in order to detect change between pairs of them. If the area of interest is large, e.g., for worldwide coverage, the storage requirement can exceed the limited onboard resources, so compressing the images as efficiently as possible without compromising the ability to detect change accurately is crucial. Then, images available onboard are non-orthorectified products which typically present geometric disparities between revisits. While implementing the full orthorectification pipeline onboard is not feasible also due to the unavailability of auxiliary information such as digital elevation models, we need to ensure that image pairs to be used for CD are registered (i.e., aligned) to each other. Finally, an efficient CD model is needed to output a change map.

Indeed, existing efforts towards onboard CD [7–10] only focus on lightweight neural networks operating on registered and orthorectified images, while neglecting registration and onboard storage. A few recent works [11, 12] have proposed CD models for unregistered images. Although these approaches effectively tackle the challenge of detecting changes between misaligned image pairs, they are not designed with onboard deployment in mind. Specifically, they neglect considerations of computational efficiency, featuring significantly higher model complexity than ours, as well as the need to minimize onboard storage requirements.

In addition, we also observe that most deep learning CD models overlook the importance of temporal ordering of input image pairs, with the exception of only a few studies [13–16].

G. Inzerillo, D. Valsesia, E. Magli are with Politecnico di Torino – Department of Electronics and Telecommunications, Italy. email: {name.surname}@polito.it. A. Fiengo is with the European Space Agency.
The activity was selected via the Open Space Innovation Platform (https://ideas.esa.int) as a Co-Sponsored Research Agreement/Study/Early Technology Development and carried out under the Discovery programme of, and funded by, the European Space Agency (contract number: 4000143591). The views expressed in this publication can in no way be taken to reflect the official opinion of the European Space Agency.

CD models are usually trained using image pairs presented in a fixed chronological sequence which biases them toward detecting changes in one specific temporal direction (for example, mainly building construction or mainly building demolition). Consequently, simply swapping the order of the two images at inference time can lead to dramatically degraded predictions. For real usage, an onboard CD model must therefore be agnostic to input ordering and capable of identifying any type of change, without being tied to a predetermined temporal direction.

In this paper, we thus present CAD (**C**ompress-**A**lign-**D**etect), an end-to-end framework for onboard CD that within a single modular neural network handles image compression, registration, and change detection. A single neural network model handling the whole CD pipeline allows optimizing it end-to-end, i.e., to maximize the change detection accuracy under a compression bitrate constraint, rather than optimizing the visual quality of the compressed images. This is clearly advantageous over a system composed of independent modules. Our contributions can be summarized as follows:

- we design a lightweight end-to-end framework for onboard CD that explicitly addresses the two key constraints neglected by prior "onboard CD" works: absence of onboard co-registration and limited onboard storage—by co-optimizing compression, registration, and detection under a rate-constrained, task-driven objective (i.e., coding for machines rather than visual fidelity), and by performing alignment and detection in feature space to reduce latency and memory;
- we introduce Light2Reg (**Light**weight **Reg**ressor for **Reg**istration), an efficient coarse-to-fine homography regressor highly effective owing to a novel joint Reg–CD training strategy that tailors the alignment to the downstream CD objective while remaining fully differentiable and lightweight;
- we propose a novel CD model, TieCD (**T**emporally-**I**nvariant and **E**fficient **C**hange **D**etector), which is mathematically invariant to the temporal image order by construction thanks to the use of permutation-equivariant blocks followed by an order-invariant fusion, yielding identical outputs when swapping the two input images without relying on data augmentation or special losses;
- we provide a comprehensive evaluation of each submodule against strong baselines (including keypoint-matching pipelines and recent CD models) and quantify deployability of the CAD pipeline via the F1–bitrate trade-off, showing that uncompressed accuracy is almost fully achieved at moderate rates, together with ablations (pixel vs. feature space, with/without joint training);
- we report measurements on low-power hardware demonstrating promising throughput and memory usage for in-orbit deployment.

A preliminary version of this work appeared in [17]. Compared to the earlier version, we (i) redesign all three modules explicitly for onboard constraints and fine-tune them jointly in the feature space with a rate-penalized ("coding-for-machines") objective; (ii) introduce Light2Reg, a tiny coarse-to-fine homography regressor whose effectiveness is unlocked by the proposed joint Reg–CD training; (iii) propose TieCD, a change detector that is temporally invariant by design; and (iv) shift most computations to feature space and broaden the evaluation (WHU-CD, SYSU) with on-device throughput.

II. Background

This work extends beyond CD to integrate several different image processing tasks such as image compression and co-registration, that are necessary for enabling onboard CD. In the following section, we briefly review the related literature for each component, highlighting both state-of-the-art methods and, where possible, also lightweight and efficient approaches suitable for on-edge deployment.

A. Image compression

Image coding standards such as CCSDS 122.0-B-1 [18] and the more recent and widely adopted CCSDS 123.0-B-2 [19] (for multispectral and hyperspectral imagery) are the de-facto choice in spaceborne missions. The CCSDS 122.0-B-1 standard employs a wavelet transform followed by bit-plane encoding and is specifically designed for onboard use with low hardware complexity. CCSDS 123.0-B-2 extends its predecessor by enabling near-lossless compression with a predictive coding scheme. However, despite its broad adoption, the latter exhibits suboptimal rate–distortion performance in the ultra-low-bitrate regime, which is a desirable operating point when images are used not for visual inspection but only to detect change.

DL-based compression methods try to overcome this limitation by learning adaptive non-linear representations and dynamically allocating bits to optimize distortion or perceptual fidelity. The earliest methods [20, 21] adopted autoencoder architectures using convolutional neural networks (CNNs) to obtain a compact latent representation, which is then quantized and entropy-coded at a given rate. Cheng et al. [22] replaced classical entropy models with discretized Gaussian mixture models and incorporated attention modules into the network, providing improved rate-distortion performance comparable to newer traditional compression methods, such as VVC [23]. He et al. [24] observed that some channels contain much more information than others and proposed an unevenly grouped channel-wise context model for entropy coding. Although DL-based compressors have been extensively studied, there are few works specific to onboard compression [25], since in such settings the limited computational resources make the design even more difficult. Valsesia et al. [26] introduced a DL predictive compression architecture that processes hyperspectral data line-by-line, bounding both memory usage and computational complexity, and outperforming the CCSDS 123.0-B-2 standard.

B. Image co-registration

Image co-registration is the process of spatially aligning two images depicting the same scene, or subject, captured from different angles, at different times or even with different modalities. Classical techniques like SIFT [27] typically

detect and describe salient keypoints, which are then used to find correspondences between two views and estimate a geometric transform to align the images. While effective in many scenarios, these methods can struggle under severe appearance changes, repetitive textures, or low-contrast regions, and they often require careful parameter tuning or manual point selection. End-to-end DL networks like SuperGlue [28] introduced graph neural networks over keypoint descriptors to establish correspondences that are significantly more accurate and robust to viewpoint and illumination variations than classical matching algorithms. Building on the foundation of SuperGlue, newer models such as LightGlue [29] offer comparable matching accuracy while reducing computational requirements and inference time; however, they may still be too complex for effective onboard deployment. Ye et al. [30] proposed a multiscale framework for remote sensing multimodal registration trained in an unsupervised fashion, achieving accurate performance. Zhou et al. [31] proposed a unified network to perform CD and image co-registration, inspired by the finding that both image registration and CD focus on extracting discriminative features. In this work we also perform registration and CD jointly, although our approach is quite different from [31], as explained in Sec. III.

C. Change Detection

Traditional CD methods go back much earlier than the development of DL-based techniques; early methods relied on direct pixel-level comparisons, such as simple image differences or ratios, to highlight temporal variations in reflectance [32]. Handcrafted feature approaches used change vector analysis to measure the magnitude and direction of spectral shifts between image pairs [33], and principal component analysis for unsupervised CD [34]. Object-based techniques segmented imagery into homogeneous regions before comparing them to label the changed pixels [35, 36]. Traditional machine learning techniques such as random forests or support vector machines improved robustness but still suffered from shallow feature representation incapable of capturing complex patterns [37, 38]. In parallel, statistical–testing approaches grounded in signal processing have provided principled detectors under explicit probabilistic models, e.g., generalized likelihood ratio tests for coherent CD in SAR [39] and multiple–covariance equality testing frameworks [40].

A huge paradigm shift occurred with DL and vision models: Daudt et al. [41] made a big step forward by introducing the use of Siamese CNNs, originally designed for segmentation, to simultaneously process bitemporal images through parallel encoder branches. Other works have further improved by using new advanced DL layers in the models, such as attention mechanisms and transformers [42, 43, 16, 44]. The current state of the art is undoubtedly represented by the foundation models [45, 46], which are first pretrained on massive remote sensing datasets and then finetuned to perform CD, among other tasks. While foundation models offer excellent performance, they are too complex for resource-constrained scenarios such as onboard processing, which led to the development of lightweight and efficient solutions. LightCDNet [9] and TinyCDNet [10], just to name a few, are two notable examples of lightweight models but with performance comparable to the state of the art.

In Sec. I we mentioned how important the property of temporal invariance is, i.e., for a CD model to always produce the same output regardless of the order in which the two input images are given to the model; only few works [13–16] explicitly target this property (referred to as *temporal symmetry* in these works), promoting it in a soft manner by randomly swapping the input pair during training and using a loss that penalizes differences between the output for the original and the swapped order. Our approach is different: in Sec. III we introduce our novel CD architecture that is mathematically temporally invariant: its internal operations guarantee identical outputs for both possible input orderings, without relying on augmented training or specialized loss functions.

III. Proposed Method

The proposed end-to-end CAD framework is illustrated in Fig. 1, which clearly outlines the three core submodules employed in this work. We begin by presenting a high-level overview of the entire framework, detailing the interaction between the individual components. We then dive into each model, highlighting the specific contributions and design choices that make them both effective and suitable for onboard deployment. Lastly, we explain the training procedure that we adopt to train the whole framework.

A. Overall Framework

The proposed CAD framework consists of three modules: image compression, registration, and CD. Each of these components plays a crucial role in enabling efficient onboard CD.

Given an input image pair $\mathbf{x}_{t_1}, \mathbf{x}_{t_2} \in \mathbb{R}^{C\times H\times W}$, the framework can be summarized as follows:

$$\hat{\mathbf{x}}_{t_1}, \hat{\mathbf{z}}_{t_1} = \mathrm{Comp}(\mathbf{x}_{t_1}) \tag{1}$$

$$\hat{\mathbf{x}}_{t_2}, \hat{\mathbf{z}}_{t_2} = \mathrm{Comp}(\mathbf{x}_{t_2}) \tag{2}$$

$$\mathbf{H}^{(1)} = \mathrm{Reg}(\hat{\mathbf{x}}_{t_1}, \hat{\mathbf{x}}_{t_2}) \tag{3}$$

$$\hat{\mathbf{w}}_{t_2} = \mathrm{Warp}(\hat{\mathbf{z}}_{t_2}; \mathbf{H}^{(1)}) \tag{4}$$

$$\hat{\mathbf{y}} = \mathrm{CD}(\hat{\mathbf{z}}_{t_1}, \hat{\mathbf{w}}_{t_2}) \tag{5}$$

where Comp identifies the compression module, Reg the registration module and CD the change detection module. Without loss of generality, we can consider t_1 to be the latest acquisition and t_2 the previous acquisition. For each input image, the compression module produces two outputs: $\hat{\mathbf{z}} \in \mathbb{R}^{F\times H/2\times W/2}$ representing intermediate feature maps extracted from the penultimate layer of the decoder neural network, and $\hat{\mathbf{x}}$ representing the reconstructed image. In the above notation, $\hat{\mathbf{w}}_{t_2}$ is the spatially warped version of $\hat{\mathbf{z}}_{t_2}$ using the homography matrix $\mathbf{H}^{(1)}$ calculated by the registration module. Finally, $\hat{\mathbf{y}}$ is the predicted change map.

Onboard CD requires the satellite to store images until either the next revisit of a location of interest or for as long as change might need to be detected. Since onboard storage is limited, we seek to optimize its usage for the CD task, i.e., spending

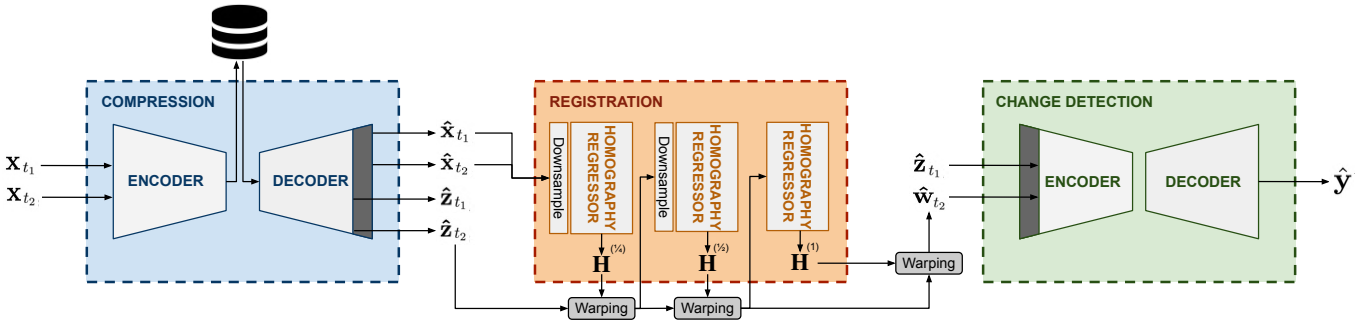


Fig. 1: The proposed CAD framework takes as input a non-co-registered image pair, $\mathbf{x}_{t_1}$ and $\mathbf{x}_{t_2}$, acquired at different times. In a realistic onboard scenario, the earlier image $\mathbf{x}_{t_1}$ is stored in compressed form and subsequently reconstructed via the decoder of the compression module; however, for clarity, we depict both images as parallel inputs. The reconstructed images, $\hat{\mathbf{x}}_{t_1}$ and $\hat{\mathbf{x}}_{t_2}$, are then passed to the co-registration module, which sequentially regresses three increasingly accurate homography matrices $\mathbf{H}^{(1/4)}, \mathbf{H}^{(1/2)}, \mathbf{H}^{(1)}$. The final homography $\mathbf{H}^{(1)}$ is used to warp the feature map $\hat{\mathbf{z}}_{t_2}$ extracted from the decoder of the compression module onto the grid of $\hat{\mathbf{z}}_{t_1}$. The aligned features are then passed to the CD module, TieCD, which outputs the final change map $\hat{\mathbf{y}}$.

the least bitrate needed to ensure good CD performance. This motivates the use of a neural image compression module, which, when end-to-end trained with the rest of the framework, allows optimizing the encoded latent representation for CD accuracy rather than simple visual quality. Each input image $\mathbf{x}$ is compressed by an encoder generating a compact latent representation, which is then quantized, entropy coded and stored in a long-term onboard storage system. When needed for CD, this representation is fetched from storage matching metadata about the geographic location and decoded to reconstruct both the full-resolution image $\hat{\mathbf{x}}$. Notice that the feature map output $\hat{\mathbf{z}}$ from the penultimate layer is also returned to enable a faster pipeline for registration and change detection that does not require decoding all the way back to pixel space.

After decompression, image co-registration is handled by the proposed registration module, called Light2Reg, which receives the reconstructed image pair $\hat{\mathbf{x}}_{t_1}$ and $\hat{\mathbf{x}}_{t_2}$ and regresses a 3×3 homography matrix $\mathbf{H}^{(1)}$. This matrix represents a projective spatial transformation (including translations, rotations, scaling, skewing, and perspective distortion) which is then used to warp the spatial grid of feature map $\hat{\mathbf{z}}_{t_2}$ over the grid of $\hat{\mathbf{z}}_{t_1}$, producing $\hat{\mathbf{w}}_{t_2}$. Unlike classical image alignment pipelines [29, 28, 47, 27], which are based on complex keypoint detection and iterative matching algorithms, this approximate alignment method is very computationally efficient and can be optimized end-to-end, making it suitable for the subsequent CD task. As shown in Fig. 1, full-resolution images are only reconstructed to compute the homography matrix, but warping and subsequent change detection are both performed directly in feature space, which is essential to make the framework faster and more efficient.

B. Image compression module

The modular design of our CAD framework allows using any state-of-the-art compression neural network, provided it meets complexity constraints and can be optimized end-to-end with a Lagrangian cost including rate and a task-specific loss (e.g., distortion, or, in our case, CD cross-entropy).

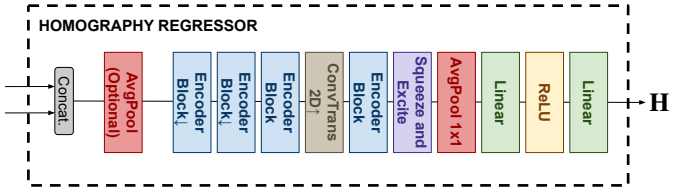


Fig. 2: Design of the homography regressor in Light2Reg. After concatenating the two input images an optional average pooling can be performed. The encoder block, the main convolutional component of the regressor, is illustrated in Fig. 3.

In our experiments, we used as compression module a custom version of the well-known Scale Hyperprior architecture [21], where the number of feature channels was reduced to keep the model efficient (150 instead of 192 for main encoder/decoder, 225 instead of 320 for hyperprior entropy bottleneck). The model has a consolidated structure for compression tasks consisting of two encoders: a main encoder generates feature representations from input images, while a hyperprior entropy model learns the data distribution of input images. After the quantization step and the entropy coding, a decoder reconstructs the input. Although it does not achieve the rate–distortion performance of current state-of-the-art methods, its reconstruction quality remains more than adequate at a relatively small complexity, representing a balanced trade-off between compression quality and efficiency. In Sec. IV we prove this claim by comparing it with newer state-of-the-art compression architectures such as ELIC [24].

C. Image co-registration module

DL–based co-registration methods that achieve the highest accuracy typically rely on keypoint extraction and matching (e.g., SuperPoint [48] + SuperGlue [28]), but these pipelines are computationally intensive, involve non-differentiable matching steps, and cannot be trained end-to-end. To enable a lighter, fully differentiable approach, we

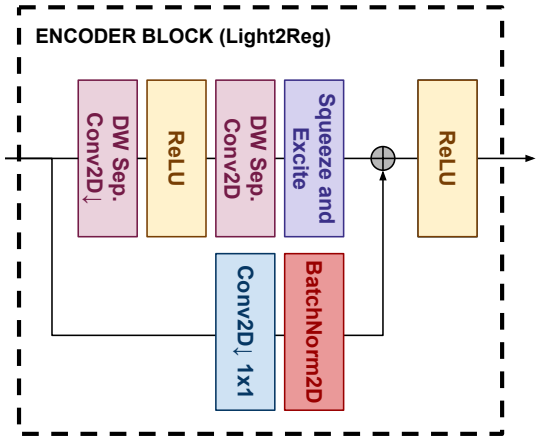


Fig. 3: Design of the efficient encoder block employed in Light2Reg. Depthwise separable convolutions and just one 1×1 2D convolution make this component extremely light.

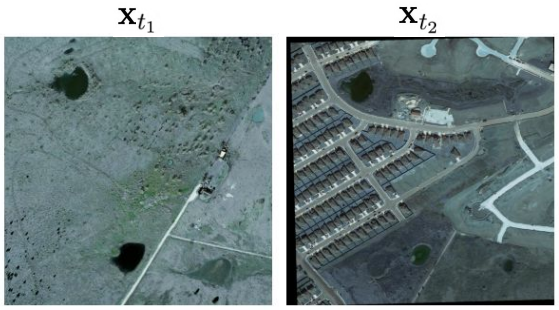


Fig. 4: Example of unregistered transformed image pair (Levir-CD). In addition to the geometric distortions that misalign the images, the semantic content is significantly different: note how $\mathbf{x}_{t_1}$ contains many more elements than $\mathbf{x}_{t_2}$ and how little remains in common.

approximate the geometric transform between image pairs with a direct homography regression. Our lightweight co-registration model, named Light2Reg, follows a coarse-to-fine strategy, inspired by the structure of [30], using a cascade of three lightweight homography regressors (depicted in Fig. 2). Differently from [30], our design prioritizes onboard efficiency and tight coupling with CD: each regressor is built from depthwise-separable convolutions with squeeze and excite (SE) attention and a small number of channels per layer, yielding in total ≈1.06 M parameters and 45.18 GFLOPs across the three scales (see Table III); moreover, we predict homographies from reconstructed images but apply them in feature space and jointly train the regressor with the CD module (*Reg–CD*, Sec. III-E), which substantially narrows the gap to heavier keypoint-matching pipelines while remaining fully differentiable and end-to-end trainable. These regressors operate at three different spatial resolutions: $1/4$, $1/2$, and full resolution.

We train Light2Reg in two stages: (i) *Solo* unsupervised pre-training on synthetically warped pairs by minimizing Eq. (25), and (ii) *Reg-CD* joint finetuning with the CD module using Eq. (27). At inference, Light2Reg follows the same coarse-to-fine procedure across three scales $1/4 \rightarrow 1/2 \rightarrow 1$, producing $\mathbf{H}^{(1/4)}, \mathbf{H}^{(1/2)}, \mathbf{H}^{(1)}$ as in Fig. 1.

We begin by downsampling the reconstructed images by a factor of 4 and processing them with a small CNN to extract coarse features, from which we predict an approximate homography matrix at quarter resolution, i.e., $\mathbf{H}^{(1/4)}$. The homography regressor, and its internal encoder block, are depicted in Fig. 2 and Fig. 3. It first concatenates the input images along the channel dimension and then uses a sequence of convolutional layers. Notice that it uses efficient separable convolutions to keep complexity low and channel attention [49] ("Squeeze and Excite" block) to introduce input adaptivity.

The estimated $\mathbf{H}^{(1/4)}$ transformation is applied to the feature maps in the $\hat{\mathbf{z}}_{t_2} \in \mathbb{R}^{F \times H/2 \times W/2}$ tensor extracted from the penultimate layer of the compression decoder. The warped features are then downsampled by a factor of 2 and processed by a second homography regressor with the same architecture but a higher number of features to extract a finer homography matrix $\mathbf{H}^{(1/2)}$. This is used to further warp the feature maps and a third pass uses the last regressor, again with a higher number of features, to correct any remaining registration errors down to the pixel level and generating the last homography matrix $\mathbf{H}^{(1)}$. This matrix is used for the final warping of the feature maps of the second image onto the spatial grid of $\hat{\mathbf{z}}_{t_1} \in \mathbb{R}^{F \times H/2 \times W/2}$, producing feature maps $\hat{\mathbf{w}}_{t_2} \in \mathbb{R}^{F \times H/2 \times W/2}$. Tensors $\hat{\mathbf{w}}_{t_2}$ and $\hat{\mathbf{z}}_{t_1}$ serve as the input to the change detection module. Since they already represent downscaled image features, the CD module can skip some layers, usually dedicated to shallow feature extraction, reducing complexity.

An important remark concerns image registration. We assume that the geometric relationships between images acquired at multiple revisits can be approximated by a projective transformation. This is, in general, suboptimal and more careful registration models could be devised if additional side information was available such as elevation models, etc. However, it is unrealistic to assume that such information is available or usable with low complexity onboard. Moreover, we remark that, to the best of our knowledge, there are currently no public large datasets that provide non-orthorectified, unregistered satellite images for change detection. Therefore, our experiments rely on simulations based on random projective transformations to the input image pairs. Although this synthetic approach can never fully reproduce real onboard data, applying strong distortions to images is a common technique in the remote sensing image co-registration literature [30, 31, 50, 51], ensuring that the reference and source data are strongly misaligned and making the co-registration task anything but trivial. Notice that since we are in the CD setting, the geometric transformations are applied to multitemporal images where the content variations make registration more complex. An example from our testing data is shown in Fig. 4 where significant change has occurred to the same area in addition to the geometric transformation.

D. Change detection module

Beyond efficiency, crucial for onboard deployment, our CD model is built from the driving idea that the model itself should be invariant to temporal permutations of the input image pair, i.e., the change map should be the same irrespective of the ordering of the two images. This property is often overlooked in several state-of-the-art designs which leads to overfitting the temporal order. Especially when limited training data are available, a non-invariant model may overfit a particular ordering dominant in the training set (e.g., building construction rather than demolition) leading to poor performance on the reverse ordering. Our experimental results prove that several state-of-the-art models degrade to unusable levels when the ordering of the input images is reversed. Some works [13–16] acknowledge the importance of the temporal invariance property, and tackle it by training-time augmentation where the ordering of image pair is randomly swapped during training and by using losses that minimize the error given by both orderings $(\mathbf{x}_{t_1}, \mathbf{x}_{t_2})$ and $(\mathbf{x}_{t_2}, \mathbf{x}_{t_1})$. These solutions can improve a model's robustness to input order, and may be effective in some cases, but they only provide a soft invariance. Furthermore, they attempt to learn the desired invariance from the data themselves, rather than encoding it in the design, possibly leading to a suboptimal use of data, especially when their quantity is limited.

Our proposed Temporally-Invariant and Efficient CD (TieCD) module represents a lightweight architecture for onboard CD which also guarantees strict mathematical invariance by construction, regardless of training data, training procedure or losses. We achieve temporal invariance through the usage of equivariant and invariant functions, whose definitions are given below.

Definition 1 (Invariance)**.** *A function $f : X \to Y$ is said to be invariant to the actions g of a group $\mathcal{G}$ if*

$$f(g \circ x) = f(x) \quad \textit{for all } x \in X,\, g \in \mathcal{G}.$$

Definition 2 (Equivariance)**.** *A function $f : X \to Y$ is said to be equivariant to the actions g of a group $\mathcal{G}$ if*

$$f(g \circ x) = g \circ f(x) \quad \textit{for all } x \in X,\, g \in \mathcal{G}.$$

While the simplest way to design a model invariant to input ordering would be to use only invariant layers throughout the network, this approach has a major limitation: there are only a limited number of simple functions invariant to permutations, and they are often too simple to capture complex spatial features necessary for CD. To address this, we adopt a more flexible strategy: we design the architecture stacking layers that are equivariant to permutation, and at the very end we apply a global invariant operation, ensuring that the final output is strictly independent of the input order, while still allowing the network to model rich and expressive features.

The proposed architecture is based on the classical U-Net structure [52] and it is depicted in Fig. 5. Unlike traditional approaches, the two temporal inputs are not concatenated along the channel dimension, as any projection along this dimension would break equivariance. Instead, the two feature maps are treated as a length-2 sequence (stacked on the batch dimension, as in Fig. 5) and are processed in parallel with shared weights; temporal information is mixed only through a permutation-equivariant operation (TESA), while strict invariance is enforced at the end by the Temporal Fusion Gate (TFG).

The main network blocks (Temporally-Equivariant Downsampling/Upsampling Blocks) are TEDB/TEUB illustrated in Fig. 5. All their operations are independently applied to each element of the temporal pair, except for the temporally equivariant self-attention (TESA) block that mixes the features of the two inputs in a permutation-equivariant way to create joint representations. This block, depicted in Fig. 6, applies self-attention [53] to the length-2 sequence of feature maps; swapping the input pair simply permutes the two outputs, preserving equivariance. More in detail, shared 1×1 convolutions followed by spatial averaging produce the query matrix $\mathbf{q} \in \mathbb{R}^{2\times F}$ and the key matrix $\mathbf{k} \in \mathbb{R}^{2\times F}$ (consistently with Fig. 6). In parallel, a shared 1×1 convolution produces the value matrix $\mathbf{v} \in \mathbb{R}^{2\times FHW}$ (obtained by vectorizing the spatial dimensions). The temporally-mixed values are then computed as

$$\mathbf{s} = \mathrm{Softmax}\left(\frac{\mathbf{q}\mathbf{k}^T}{\sqrt{F}}\right)\mathbf{v}, \tag{6}$$

where $\mathbf{s} \in \mathbb{R}^{2\times FHW}$ is finally reshaped back to $2\times F\times H\times W$ to continue the forward pass in each TEDB/TEUB.

In the bottleneck of our TieCD U-Net (Fig. 5), we apply three spatial attention blocks in sequence, namely *Global Self-Attention → Multiscale Self-Attention → Multihead Attention*. Importantly, these three attentions operate independently on each element of the temporal pair (i.e., they are purely spatial and do not perform temporal mixing, unlike TESA).

The Global Self-Attention block enhances features by computing long-range spatial dependencies through self-attention and applying a residual refinement. Given an input feature map of size $F \times H \times W$, we obtain the query and key matrices $\mathbf{q}_\mathrm{G}, \mathbf{k}_\mathrm{G} \in \mathbb{R}^{HW\times \frac{F}{8}}$ by applying 1×1 convolutions that reduce channels to $\frac{F}{8}$ and then reshaping to $HW \times \frac{F}{8}$. Analogously, the value matrix $\mathbf{v}_\mathrm{G} \in \mathbb{R}^{HW\times F}$ is obtained via a 1×1 convolution projected to F channels and reshaped to $HW\times F$. In formulas:

$$\mathbf{A} = \mathrm{Softmax}\left(\frac{\mathbf{q}_\mathrm{G}\cdot\mathbf{k}_\mathrm{G}^\top}{\sqrt{F/8}}\right) \in \mathbb{R}^{HW\times HW}, \tag{7}$$

$$\mathbf{o} = \mathbf{A}\mathbf{v}_\mathrm{G} \in \mathbb{R}^{HW\times F}. \tag{8}$$

After reshaping $\mathbf{o}$ back to $F\times H\times W$, a separable convolution, BatchNorm and a Mish non-linearity [54] are applied to compute the output of the block.

The subsequent Multiscale Self-Attention block takes as input the output of the Global Self-Attention block and applies the same global self-attention mechanism at three different scales: the original scale, and two additional scales obtained via undecimated averaging filters of size 2×2 and 4×4. The three resulting $F \times H \times W$ outputs are stacked along the channel dimension to form $\mathbf{o}_\text{cat} \in \mathbb{R}^{3F\times H\times W}$, which is then processed with a 2D convolution, BatchNorm and

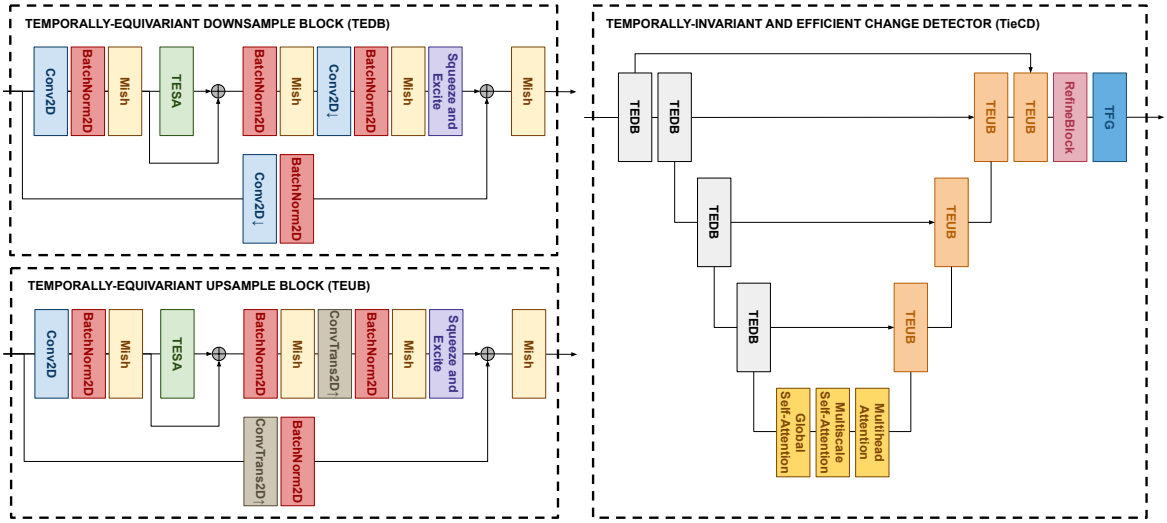


Fig. 5: *Left*: designs of the Temporally-Equivariant Downsample Block (TEDB) and Temporally-Equivariant Upsample Block (TEUB), used as main building blocks of TieCD. *Right*: design of Temporally-Invariant and Efficient Change Detector (TieCD). By concatenating the input images on the batch dimension and performing a series of equivariant operations on the order of permutations we maintain the equivariance property. Invariance is eventually achieved using a global temporally invariant operation: the Temporal Fusion Gate (TFG).

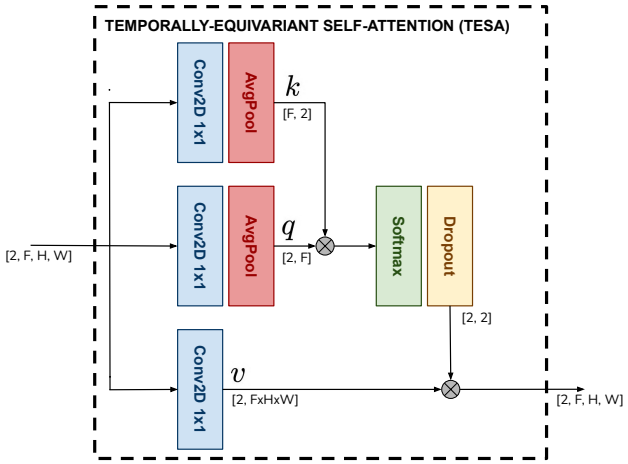


Fig. 6: Design of the temporally equivariant self-attention (TESA) block. It computes pairwise attention scores within each temporal pair; swapping the input pair ($\mathbf{x}_{t_1} \leftrightarrow \mathbf{x}_{t_2}$) simply permutes the outputs in the same exact way, satisfying the formal definition of equivariance.

Mish non-linearity to produce the multiscale-attended features $\mathbf{u} \in \mathbb{R}^{F\times H\times W}$.

Finally, the features $\mathbf{u} \in \mathbb{R}^{F\times H\times W}$ produced by the Multiscale Self-Attention block are fed to a Multihead Attention block:

$$\mathbf{u}' = \text{reshape}(\mathbf{u}) \in \mathbb{R}^{HW\times F}, \tag{9}$$
$$\mathbf{h}_1 = \mathbf{u}' + \text{MHSA}(\text{LN}_1(\mathbf{u}')), \tag{10}$$
$$\mathbf{h}_2 = \mathbf{h}_1 + \text{MLP}(\text{LN}_2(\mathbf{h}_1)), \tag{11}$$
$$\mathbf{p} = \text{reshape}(\mathbf{h}_2) \in \mathbb{R}^{F\times H\times W}. \tag{12}$$

Here, MHSA is the multi-head self-attention mechanism, MLP is a two-layer linear network with GeLU activation, and LN_i denotes layer normalization.

The decoder portion of the temporally equivariant part of the U-Net architecture is terminated by a Refinement Block, whose goal is to refine the feature maps:

$$\mathbf{r}_1 = \text{Mish}(\text{BN}_1(\text{Conv}_{5\times5}(\mathbf{p}))), \tag{13}$$
$$\mathbf{r}_2 = \text{Mish}(\text{BN}_2(\text{Conv}_{3\times3}(\mathbf{r}_1))), \tag{14}$$
$$\mathbf{e} = \text{Conv}_{1\times1}(\mathbf{r}_2). \tag{15}$$

The refinement block is applied independently to each element of the temporal pair, thus producing $\mathbf{e} \in \mathbb{R}^{2\times F\times H\times W}$; we denote its two elements as $\mathbf{e}_{t_1}, \mathbf{e}_{t_2} \in \mathbb{R}^{F\times H\times W}$. After the refinement block, swapping the input order simply permutes $\mathbf{e}_{t_1}$ and $\mathbf{e}_{t_2}$, preserving equivariance.

The last block of the network (Temporal Fusion Gate, TFG) fuses $\mathbf{e}_{t_1}$ and $\mathbf{e}_{t_2}$ into a single representation that is strictly invariant to input ordering. TFG computes the invariant output $\hat{\mathbf{y}}$ via:

$$\mathbf{a} = \tfrac{1}{2}\big(\mathbf{e}_{t_1} + \mathbf{e}_{t_2}\big), \tag{16}$$
$$\mathbf{d} = \big|\mathbf{e}_{t_1} - \mathbf{e}_{t_2}\big|, \tag{17}$$
$$\mathbf{c} = \tfrac{1}{2}\big(\mathbf{a} + \mathbf{d}\big), \tag{18}$$
$$\mathbf{m} = \sigma\big(\text{Conv}(\mathbf{c})\big), \tag{19}$$
$$\hat{\mathbf{y}} = \mathbf{m} \odot \mathbf{a}. \tag{20}$$

Equations (16) and (17) compute symmetric statistics via average and absolute difference, respectively, which are then merged by (18) into combined features $\mathbf{c}$. Equation (19) produces a modulation mask $\mathbf{m} \in [0, 1]^{F\times H\times W}$, and (20) yields the invariant output. It is easy to verify that $\hat{\mathbf{y}}$ is invariant to the ordering of $\mathbf{e}_{t_1}$ and $\mathbf{e}_{t_2}$.

E. Training Procedure

Training a complex and modular framework such as CAD is far from trivial. Direct end-to-end optimization straight from random initialization may be suboptimal as it is hard for the sub-modules to learn the specifics of their functionality for the CD objective alone. Also, it is easy for gradients to vanish or explode over such a long pipeline. For these reasons, it is crucial to carefully design the training strategy, not just for each individual module, but for the framework as a whole. We start by pretraining each module independently, optimizing it with its own respective loss function, which we now describe, in order to let it learn the desired functionality within the framework.

For the compression module we follow a well-established approach [20, 21] and minimize the following rate–distortion loss:

$$\mathcal{L}_{\mathcal{C}} = \lambda R + D, \tag{21}$$

where

$$R = \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}}\big[-\log p_{\hat{\mathbf{b}}}(\hat{\mathbf{b}})\big] \tag{22}$$

is the expected bitrate of the quantized latent code $\hat{\mathbf{b}}$ (entropy-coded for onboard storage), and

$$D = \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}}\big\|\mathbf{x} - \hat{\mathbf{x}}\big\|_2^2 \tag{23}$$

is the mean squared reconstruction error between the original image $\mathbf{x}$ and its reconstruction $\hat{\mathbf{x}}$. The hyperparameter λ balances rate and distortion, leading to different operating points.

Concerning the registration module, we recall that Light2Reg generates three progressively more precise homography matrices. The registration loss aggregates the misalignment errors at each of the three scales $s \in \{1/4,\, 1/2,\, 1\}$ by comparing the reference image patches $\mathbf{x}_{t_1}^{(s)}$ to their warped counterparts $\mathbf{w}_{t_2}^{(s)}$. Suppose that $\mathbf{H}^{(s)}$ denotes the 3×3 homography matrix predicted at resolution s, then

$$\mathbf{w}_{t_2}^{(s)} = \text{Warp}\big(\mathbf{x}_{t_2}^{(s)};\, \mathbf{H}^{(s)}\big). \tag{24}$$

The total registration loss is then defined as

$$\begin{aligned}\mathcal{L}_{\mathcal{R}} = {} & \beta_1 \big\|\mathbf{x}_{t_1}^{(1)} - \mathbf{w}_{t_2}^{(1)}\big\|_2^2 \\ & + \beta_2 \big\|\mathbf{x}_{t_1}^{(1/2)} - \mathbf{w}_{t_2}^{(1/2)}\big\|_2^2 \\ & + \beta_3 \big\|\mathbf{x}_{t_1}^{(1/4)} - \mathbf{w}_{t_2}^{(1/4)}\big\|_2^2.\end{aligned} \tag{25}$$

Here, the weights $\beta_1, \beta_2, \beta_3$ balance the contribution of each scale, with $\beta_i \geq 0$ and $\sum_{i=1}^{3} \beta_i = 1$. By penalizing squared differences at progressively finer resolutions, the network first corrects large misalignments on coarse grids and then refines them to subpixel precision at full resolution. To pretrain the registration network independently we minimize the loss in (25).

Lastly, to pretrain the CD module we optimize a standard cross-entropy loss:

$$\mathcal{L}_{\text{CD}} = -\sum_i \left[\mathbf{y_i} \log(\hat{\mathbf{y}}_\mathbf{i}) + (1 - \mathbf{y_i}) \log(1 - \hat{\mathbf{y}}_\mathbf{i})\right], \tag{26}$$

where $\mathbf{y_i} \in \{0, 1\}$ is the ground-truth label indicating the presence or absence of change on pixel i, and $\hat{\mathbf{y}}_\mathbf{i} \in [0, 1]$ is the predicted probability. Notice that pretraining the CD module requires adding two layers at the beginning to let it operate in the full-resolution pixel space rather than in the downsampled feature space that we described in Sec. III-D. These layers are then removed for the integration in the pipeline.

After individually pretraining each module, we proceed with a joint pretraining of the registration and CD modules. This joint training of these two modules serves two main purposes. First, it improves the performance on both tasks which mutually benefit from the joint optimization, as we show in Sec. IV. Second, it defines an upper bound on CD accuracy for the entire framework when no compression is applied. This provides a reference point for evaluating the whole framework, including the compression module, to quantify the potential performance degradation introduced by compression, which we recall is necessary, as it significantly reduces onboard storage utilization. For the joint registration and CD pretraining (referred to as *Reg-CD*), we employ a composite loss function that combines the registration loss from Eq. (25) and the CD loss defined in Eq. (26). The total loss is formulated as:

$$\mathcal{L}_{\text{RCD}} = \alpha \mathcal{L}_{\text{CD}} + (1 - \alpha)\mathcal{L}_{\mathcal{R}}, \tag{27}$$

where $\alpha \in [0, 1]$ is a hyperparameter that balances the contributions of the change detection loss $\mathcal{L}_{\text{CD}}$ and the registration loss $\mathcal{L}_{\mathcal{R}}$, encouraging the network to improve both alignment and detection accuracy simultaneously.

Finally, the whole framework, comprising compression, registration, and CD, is finetuned end-to-end. For this purpose, we adopt a rate-penalized Lagrangian objective that jointly optimizes the three modules. Specifically, we combine the change detection loss $\mathcal{L}_{\text{CD}}$, the registration loss $\mathcal{L}_{\mathcal{R}}$, and the rate of the compressed image representation defined in Eq. (22). The total finetuning loss is defined as:

$$\mathcal{L}_{\text{total}} = [\alpha \mathcal{L}_{\text{CD}} + (1 - \alpha)\mathcal{L}_{\mathcal{R}}] + \lambda R. \tag{28}$$

Looking at our loss for the whole framework, we notice how we initially pretrain the compression module using a conventional rate–distortion loss (as described in Eq. (21)), allowing us to learn an effective representation to encode the input images. However, unlike standard image compression pipelines that aim to reconstruct perceptually faithful images, our objective changes in the full end-to-end setting. Once the registration and change detection modules are also trained, we finetune the entire architecture jointly using the total loss. In this stage, we no longer optimize for reconstruction quality; instead, we treat both CD and registration accuracy as distortion measures. This shift reflects a *coding-for-machines* [55] approach, where the compression model is optimized to reconstruct the most relevant features to promote accurate CD, rather than generating visually faithful reconstructions of the input images.

IV. Experimental setting and main results

In this section we present the experimental settings and results of the entire framework as well as an extensive analysis of both performance and complexity. FLOPs and number of

parameters reported in this section are computed using the *calflops* [56] package with input images of size $3 \times 512 \times 512$.

A. Datasets

To train and evaluate the effectiveness of the proposed method, and each submodule, we used three datasets for different purposes. We used CloudSEN12 [57] to pretrain only the registration network alone. Levir-CD [42] was used to perform compression pretraining, CD pretraining, joint *Reg-CD* pretraining, end-to-end finetuning and evaluation of the entire framework. CloudSEN12 [57] comprises over 49,000 multi-temporal image patches, spanning clear scenes, thick and thin clouds, cloud shadows, along with dense pixel-level annotations. Although its primary purpose is cloud segmentation, the diversity and scale of the dataset make it well suited for pretraining our registration module. Each image is labeled with a flag that indicates the cloud coverage; to pretrain the registration network we used only the subset of images labeled as "low-cloudy", "almost-clear", "cloud-free" to avoid the registration of scenes containing only clouds. Furthermore, we just used RGB spectral bands, to maintain consistency with the Levir-CD dataset. Levir-CD [42] is a widely used benchmark for the CD task. It consists of 637 high-resolution (spatial resolution is 0.5 meters) 1024×1024 8-bit RGB images. To pretrain the compression module, our CD model and jointly pretrain registration and CD networks (*Reg-CD*) we followed the standard splits provided by the authors of the dataset, randomly cropping training images to 512×512 patches.

In addition, to further evaluate our CD module and the overall CAD framework, we tested it also on the WHU-CD [58] dataset, which consists of high-resolution aerial image pairs captured over Wuhan, China, and on the SYSU [59] dataset. WHU-CD mainly includes urban and suburban areas with a variety of land cover changes such as building construction, demolition, and road expansion. Due to its large size and rich structural details, WHU-CD serves as a challenging benchmark for assessing both the spatial accuracy and robustness of change detection models. SYSU [59] comprises bi-temporal optical remote-sensing image pairs collected across multiple regions in China, spanning urban, suburban, and natural scenes with heterogeneous change types (e.g., building construction/demolition, road modifications, and land-cover transitions), thereby providing broader appearance variations (illumination/seasonal) and a complementary benchmark for testing generalization.

B. Implementation details

All models were implemented, and experiments were conducted, using the PyTorch library. For training we used one NVIDIA A40 48GB GPU, while for evaluation under a low-power, resource-constrained setup, we used the NVIDIA Jetson Orin Nano, a commercial 8GB embedded system designed for AI inference. For the latter, we conducted our evaluation using the 15 W power budget mode. For all the trainings we used the Adam optimizer [60].

To simulate the lack of orthorectification and co-registration in images, we applied strong affine and perspective transformations using the *torchvision* library. For training involving CD, the transformations are applied to one of the images in the image pair, while for pretraining of only the registration module we applied transformations to the single reference image to obtain the source image. It is important to remark that CAD was trained by randomly sampling different spatial transformations at each epoch in order to make it robust. For this reason, we want to test our proposed framework under a variety of random distortions in order to assess its robustness. Thus, we evaluate CAD with 100 independent testing runs, each with a different set of distortions, and report error bars on the results related to this variability. Error bars in all figures report one standard deviation around the mean. In comparative experiments where we test different models, the same sets of distortions are applied to all the models, to guarantee a fair comparison.

We list here all the implementation details related to each evaluation and training setup:

- *Compression module*: compression pretraining and evaluation were performed on the Levir-CD dataset, using the default train, evaluation and test splits. The learning rate was set to 10^{-4} for the encoder–decoder model and 10^{-5} for the entropy bottleneck model. Models were trained for 400 epochs with a batch size of 8. Different λ values were used, leading to different models with different rate–distortion trade-offs. The chosen efficient compression module is based on the Scale Hyperprior architecture [21] and differs from the standard configuration only in its channel width (i.e., number of latent channels), which we reduce for onboard efficiency while keeping the architecture unchanged. In particular, we use 150 latent channels for the main encoder–decoder branch (instead of the original 192) and 225 channels for the hyperprior entropy-bottleneck branch (instead of the original 320). In the ablations, we assess other compression models (ELIC [24], Factorized Prior [21]) for which we used the implementations provided in the *compressai* library [61].
- *Registration module*: we adopt a two-stage training procedure for Light2Reg. Stage 1 *(Solo)*: unsupervised pretraining on CloudSEN12, where we generate a "source" image by applying random projective transformations to a reference image and train Light2Reg to regress the homography by minimizing the multi-scale loss in Eq. (25). Models were trained for 500 epochs with a batch size of 24 and a learning rate of 10^{-4}. Stage 2 *(Reg–CD)*: joint finetuning of Light2Reg together with the CD module on Levir-CD using the combined loss in Eq. (27) (with $\alpha = 0.3$), which adapts the registration to the downstream change detection objective.
- *CD module*: we study two variants of TieCD, with different levels of complexity, namely TieCD-L and TieCD-S. The former has a total complexity of 58.98 GFLOPs, while the latter is a highly compact variant using only half the number of features for a total of 15.25 GFLOPs. We

pretrained them for 1500 epochs, with a batch size of 8, and 10^{-5} as starting learning rate which we decreased using a step scheduler ($\gamma = 0.3$) after 1000 epochs. For the pretraining step, no distortions were applied to the image pairs. We used random rotation, flipping (both horizontal and vertical) and color jittering as training augmentations. To train and evaluate other baseline CD models, we used standard configurations and implementations from the OpenCD [62] library. Notably, this implementation of many models achieves even better results than those reported by the original authors.

- *Joint Reg-CD*: to perform the joint pretraining of Light2Reg and TieCD we used as starting weights the ones obtained from the independent pretraining of each module. We minimized the loss in Eq. (27) by using $\alpha = 0.3$. Since in this setting compression is not performed and thus feature maps from the compressor are not available, the two modules work with all inputs in the pixel space, avoiding passing feature maps to TieCD. For this experiment, batch size was set to 4, starting learning rate to $3 \cdot 10^{-5}$, weight decay to 10^{-5}, and the number of epochs to 1500. After 1000 epochs we multiplied the starting learning rate by 0.3.
- *Entire Framework*: to finetune the whole CAD framework we started from the pretrained weights obtained from the compression pretrain, for the compression model, and from the joint *Reg-CD* weights for Light2Reg and TieCD. Since the entire framework performs compression, we conducted many experiments testing different λ values and producing as many sets of weights for all the three submodules. Notice that, without loss of generality, variable bitrate training techniques known in the literature could also be used. In the total loss (Eq. 28) we set $\alpha = 0.3$. For all experiments, batch size was set to 4, learning rate to $3 \cdot 10^{-5}$, and number of epochs to 1000.

C. Evaluation metrics

Since we perform three different tasks to achieve onboard CD, in this work we employ different quantitative metrics to evaluate not only the effectiveness of CD, but also compression and registration.

a) Peak Signal-to-Noise Ratio (PSNR): to evaluate distortion in compression we report PSNR:

$$\text{PSNR} = 10 \log_{10}\Big(\frac{L^2}{\text{MSE}}\Big), \tag{29}$$

where L is the maximum pixel value (e.g., 255 for 8-bit images).

b) F1-score: for CD, we focus on the positive (change) class and compute the micro-averaged F_1-score:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{30}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{31}$$

$$F_1 = 2 \cdot \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}, \tag{32}$$

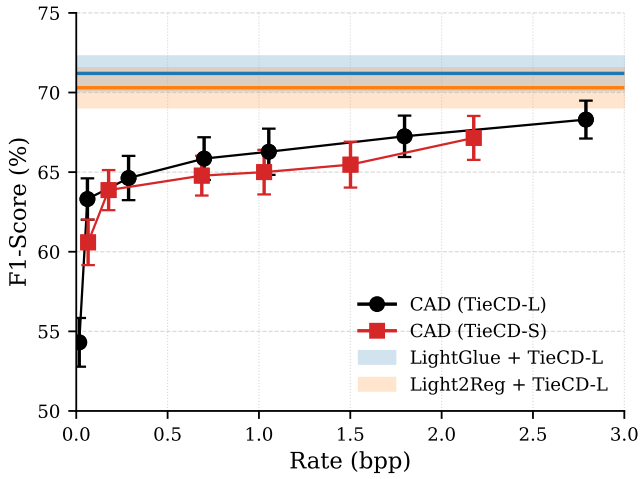


Fig. 7: Trade-off between compression rate and F1-score on Levir-CD dataset. Each marker on the solid curves shows the mean F1-score (with error bars indicating standard deviation) achieved by the entire CAD framework at a given compression rate in bits-per-pixel, employing both TieCD-S and TieCD-L. The two horizontal lines correspond to the mean F1-score of the framework without compression (i.e., *Reg-CD*), with shaded regions showing their respective standard deviations.

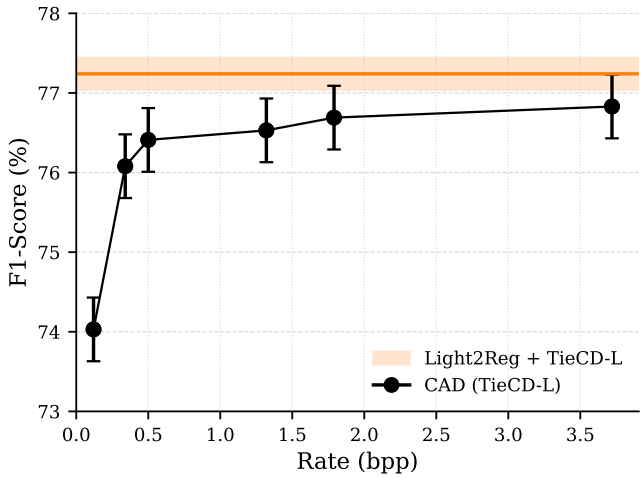


Fig. 8: Trade-off between compression rate and F1-score on SYSU dataset.

where TP, FP, and FN denote the total true positives, false positives, and false negatives, respectively, for the change class.

D. Framework main results

Fig. 7 shows the main quantitative results for the proposed CAD framework tested on Levir-CD dataset. We report the F1-score achieved on the CD task as a function of the rate of the compressed image representations in bits-per-pixel. We evaluate two variants of CAD framework, one using the smaller TieCD-S as CD module and one with the larger TieCD-L, reporting mean score values and standard

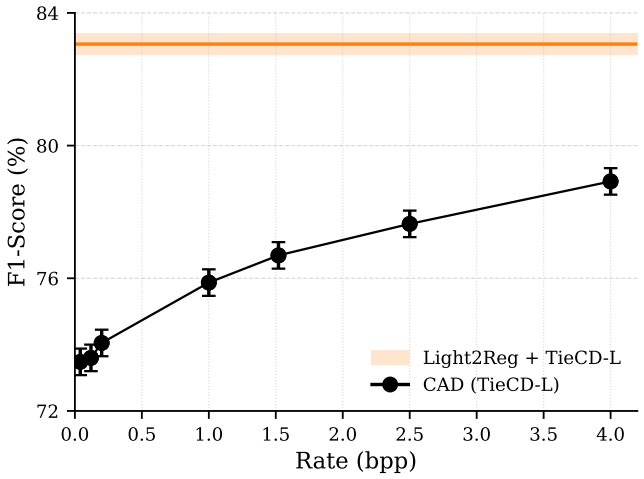


Fig. 9: Trade-off between compression rate and F1-score on WHU-CD dataset.

deviation error bars accounting for the variability in spatial transformations. The horizontal lines represent benchmark results obtained from experiments without compression, i.e., in *Reg-CD* configuration, representing a theoretical upper bound to the CAD framework in the case where there is no loss due to compression. Two baselines are shown: one uses our proposed Light2Reg registration network (in orange), which is highly efficient and tailored for onboard deployment; the other uses LightGlue [29] (in blue), a lighter and more accurate version of SuperGlue [28] that effectively represents the state of the art in image matching and registration. We notice that the significantly more expensive LightGlue only allows a modest improvement in F1-score (+0.9, as shown in Table III), demonstrating that Light2Reg, although much lighter, provides good image co-registration for the CD task. In Subsec. V-B we further analyze the differences between LightGlue and Light2Reg, also from a computational point of view. We can notice that at a rate of approximately 3 bpp, CAD is close to the performance of the uncompressed system. Moreover, a slow, smooth decrease in F1-score is reported as rate is lowered until at very low rates (below 0.1 bpp), F1-score falls by about 10 percentage points. These results suggest that CD can be performed onboard with small requirements in terms of storage as good accuracy can be obtained even for very high compression ratios. Similar behavior holds across the other CD benchmark datasets. On SYSU (Fig. 8), the F1–rate curve increases steadily and approaches the *Reg-CD* upper bound at higher bitrates; error bars remain tight, indicating robustness to the applied spatial perturbations. On WHU-CD (Fig. 9), the trend is likewise monotonic, with a small residual gap at the highest tested rate. Overall, the three curves exhibit consistent shapes and variances, confirming that the storage–accuracy trade-off observed on Levir-CD transfers well to different scenes and annotation styles; the uncompressed performance can be almost fully achieved at moderate rates, while low-rate operation ($\leq$1 bpp) still yields competitive F1.

Qualitative examples in Fig. 10 underscore this behavior: even at extremely low bitrates (e.g., 0.07 bpp) predictions remain visually coherent and spatially consistent, with most of the gap to the *Light2Reg + TieCD-L* upper bound manifesting as slightly noisier, less sharp boundaries. As the rate increases, masks progressively sharpen and adhere more closely to object boundaries, with fewer spurious blobs and small omissions; the observed F1 gains thus stem mainly from cleaner, smoother edges rather than wholesale changes in the detected regions. Overall localization remains stable across all rates, while contour fidelity and confidence improve, confirming the robustness of CAD.

TABLE I: CAD computational complexity and throughput. Inference time and memory usage are computed on a Jetson Orin Nano 8 GB passing $3 \times 512 \times 512$ input tensors.

TieCD	Params	FLOPs	Inference (seconds)	Memory (MB)	Throughput (pixels/sec.)
Large	8.81M	121.47G	0.383 ± 0.001	5729 ± 9	685K
Small	8.22M	110.9G	0.341 ± 0.001	5391 ± 2.37	768K

Table I summarizes the overall computational profile of CAD framework. With 121 GFLOPs and 8.8 million parameters, CAD (using TieCD-L) achieves a throughput of 685 Kpixels/s when evaluated on the low-power Nvidia Jetson Orin Nano. With a small loss in terms of CD accuracy, as shown in Fig. 7, one can further speed (and lighten) up the pipeline by using the smaller CD network version, TieCD-S. This smaller variant increases throughput by about 80K pixels/s, and decreases memory usage by about 300 MB. Both Fig. 7 and Table I demonstrate that the framework comfortably meets the efficiency constraints of onboard deployment.

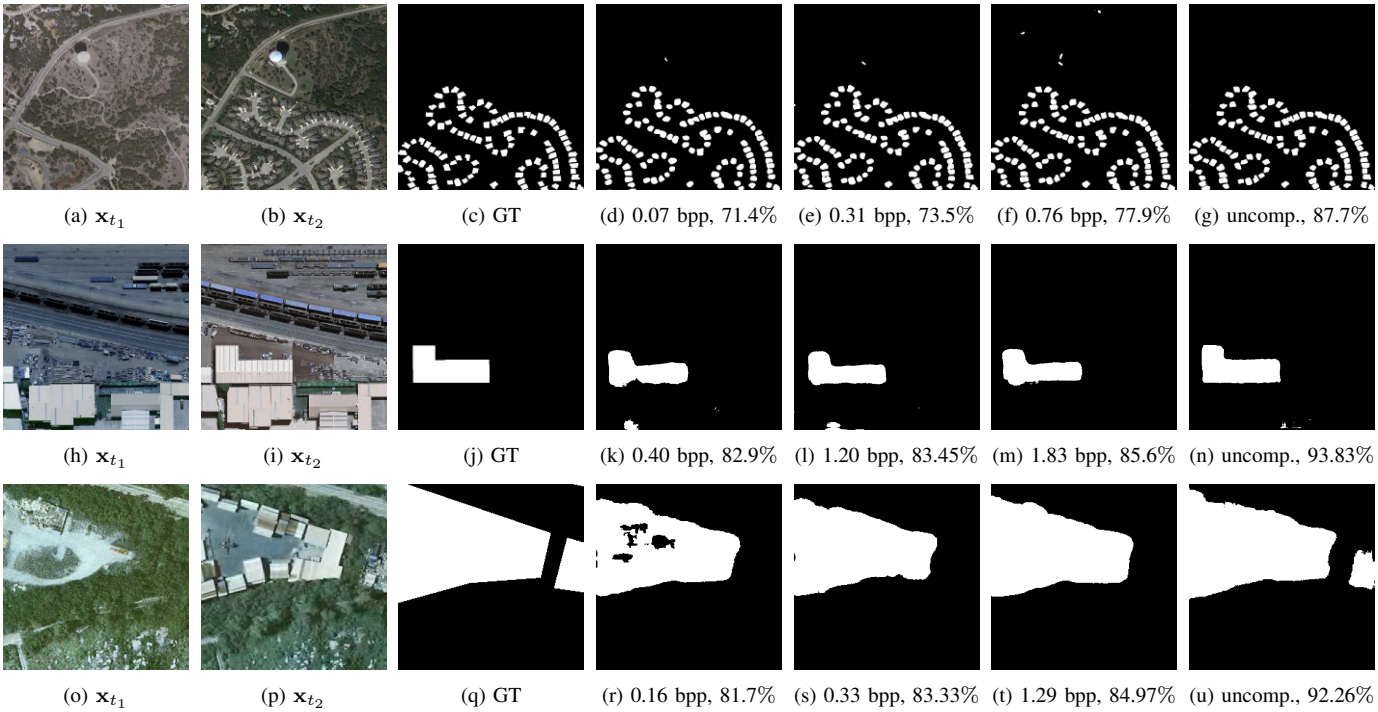
V. Analysis and Ablations

In this section we present the qualitative and quantitative results of the three modules of our CAD framework. Furthermore, we present some ablations to assess the impact of specific hyperparameters and adopted training strategies. FLOPs and number of parameters reported in this section are computed using the *calflops* [56] package with input images of size $3 \times 512 \times 512$.

A. Change detection module results

We now assess the individual performance of our proposed TieCD module in order to understand its positioning compared to state-of-the-art techniques. In this experiment, images are perfectly registered and no compression is applied. We remark that our design goal is to strike a balance between complexity and detection accuracy. Table II reports CD performance of TieCD (in both its small and large variants) and other state-of-the-art CD methods on three benchmarks (Levir-CD, WHU-CD, SYSU), explicitly listing both the standard and the swapped temporal order and clustering methods by their robustness to it. TieCD is the only method whose predictions remain exactly the same when images are swapped, thanks to the strict temporal invariance enforced in the design.

A few dataset-specific trends emerge. Levir-CD exhibits strong order sensitivity for many architectures: models such as FC-Siam-Conc, ChangeFormer, HANet and BAN suffer



(a) $\mathbf{x}_{t_1}$ (b) $\mathbf{x}_{t_2}$ (c) GT (d) 0.07 bpp, 71.4% (e) 0.31 bpp, 73.5% (f) 0.76 bpp, 77.9% (g) uncomp., 87.7%

(h) $\mathbf{x}_{t_1}$ (i) $\mathbf{x}_{t_2}$ (j) GT (k) 0.40 bpp, 82.9% (l) 1.20 bpp, 83.45% (m) 1.83 bpp, 85.6% (n) uncomp., 93.83%

(o) $\mathbf{x}_{t_1}$ (p) $\mathbf{x}_{t_2}$ (q) GT (r) 0.16 bpp, 81.7% (s) 0.33 bpp, 83.33% (t) 1.29 bpp, 84.97% (u) uncomp., 92.26%

Fig. 10: Qualitative results. Top row: Levir-CD. Middle row: WHU-CD. Bottom row: SYSU. In each row, (a)–(b) / (h)–(i) / (o)–(p) show the images at time t_1 and t_2; (c) / (j) / (q) show the Ground Truth; the remaining subfigures report CAD outputs at different bitrates and the *Reg-CD* upper bound without compression. For the last four subfigures of each row, values in the caption represent bitrate and F1 score.

TABLE II: F1-score across Levir-CD, WHU-CD, and SYSU for both input orders. CD methods are clustered into two groups: those that are robust to temporal order swapping and those that are not.

			Levir-CD		**WHU-CD**		**SYSU**	
CD Method	**Params**	**FLOPs**	**F1**$(\mathbf{x}_{t_1},\mathbf{x}_{t_2})$	**F1**$(\mathbf{x}_{t_2},\mathbf{x}_{t_1})$	**F1**$(\mathbf{x}_{t_1},\mathbf{x}_{t_2})$	**F1**$(\mathbf{x}_{t_2},\mathbf{x}_{t_1})$	**F1**$(\mathbf{x}_{t_1},\mathbf{x}_{t_2})$	**F1**$(\mathbf{x}_{t_2},\mathbf{x}_{t_1})$
TieCD-S (ours)	0.20M	15.25G	91.04	91.04	87.30	87.30	76.46	76.46
TieCD-L (ours)	0.81M	58.98G	91.61	91.61	88.68	88.68	78.22	78.22
ChangerEx (R18) [16]	11.40M	47.46G	91.81	91.83	88.53	87.87	79.56	79.55
ChangeStar [13]	17.00M	153.5G	91.26	91.03	88.90	88.88	77.00	77.12
TinyCD [10]	0.29M	11.20G	90.33	56.74	91.74	2.68	78.42	76.85
FC-Siam-Conc [41]	1.55M	39.93G	87.14	1.42	83.12	0.66	75.12	70.07
LightCDNet-L [8]	2.82M	50.85G	91.15	58.00	86.38	80.76	77.69	72.27
ChangeFormer (B1) [43]	13.90M	52.84G	91.22	5.08	87.56	3.87	78.69	70.23
HANet [63]	3.03M	135.8G	90.53	8.30	88.16	15.01	76.42	67.95
BAN (ViT-L14-b1) [46]	16.00M	574G	92.10	4.70	92.50	-	-	-
TTP [64]	6.21M	1.72T	91.93	86.38	-	-	82.72	-

dramatic collapses when the input order is reversed, while ChangerEx and ChangeStar remain robust (the latter two promote temporal symmetry during training). Visual inspection of Fig. 11 shows these trends: our TieCD-L produces exactly the same, pixel-wise identical change maps for both temporal orders, as expected from its design-enforced invariance. In contrast, BAN and TinyCD suffer severe degradations when the order is swapped, often yielding maps that are barely usable. ChangerEx remains broadly consistent, yet the two outputs are visibly not the same, indicating that it is not strictly invariant.

WHU-CD amplifies this effect even further: TinyCD and FC-Siam-Conc perform competitively in the standard order but degrade almost completely when swapped, whereas ChangeStar and (to a slightly lesser extent) ChangerEx remain stable even if their outputs are not exactly the same when swapping the input permutations (i.e., they are not invariant). By contrast, SYSU appears less susceptible to order bias: several baselines still incur a drop, but it is typically moderate (e.g., TinyCD, FC-Siam-Conc, ChangeFormer), and the overall ranking is closer across methods. This suggests that, as already mentioned, temporal-order robustness is dataset-dependent, and likely influenced by scene diversity and the prevalence of directional change patterns; designing models that are temporally invariant by construction avoids such sensitivity.

In terms of accuracy–efficiency trade-offs, TieCD-L attains solid performance across datasets while remaining lightweight (0.81 M params, 58.98 GFLOPs). Foundation-model baselines (BAN, TTP) reach high scores in one order, but either lack results for the swapped case or show non-negligible drops,

underscoring the benefit of enforcing temporal invariance at the architectural level rather than relying on data augmentation alone.

B. Registration module results

In this section, we analyze the effectiveness of the registration module in terms of its impact on the performance of the CD task, the improvement due to the joint training strategy and computational complexity against alternative state-of-the-art approaches.

Table III reports the CD F1-score without any compression when the registration and CD module are used under two evaluation settings. In the *Solo* setting, each model is simply pretrained (as detailed in Sec. III-E) on its own objective and then frozen at test time, allowing us to measure the raw alignment quality of the registration method without further adaptation. In the *Reg–CD* setting, the registration and CD modules are jointly finetuned, providing insight into how much each task benefits from shared optimization.

We also compare our Light2Reg co-registration network with the state of the art for DL-based keypoint matching: LightGlue [29]. Unlike Light2Reg, which directly regresses a single homography matrix, LightGlue requires first extracting keypoint descriptors and then iteratively matching them. Based on the most accurate results of the LightGlue paper, we opt for SuperPoint [48] as a DL-based method to extract keypoints. Notice that Table III reports parameter count including both SuperPoint and LightGlue models, while the number of FLOPs is variable due to scene complexity, since iterative algorithms are used, and thus we report a lower bound. This lower bound accounts for the two inference passes by SuperPoint (178.5 GFLOPs each), excluding the matching iterations, which we observed ranging from an additional 40 GFLOPs for trivial image pairs to more than 150 GFLOPs for challenging cases.

The experimental results show that unregistered images degrade the CD F1-score, even when state-of-the-art techniques like LightGlue are used. However, we notice how the proposed Light2Reg in the joint *Reg-CD* training setup has very close performance, despite requiring about one order of magnitude fewer FLOPs. We also notice how the joint *Reg-CD* training setup significantly improves over *Solo* training, which is only able to provide coarse alignment that limits downstream CD accuracy. This is to be expected since the *Solo* training procedure relies on regressing the homography applied to a single image, thus having no content variation in the image pair. However, when used for CD, there are significant content differences. Hence, supplementing the training objective with the CD ground truth provides better alignment, as it trains the model to be more robust to scene variations. Lastly, we observe that also the pipeline using LightGlue benefits from the joint training with the CD model. A qualitative example of the impact of joint *Reg-CD* training is shown in Fig. 12.

C. Compression module results

In this section, we perform a comparative evaluation of DL image compression models in order to select one that strikes a balance between rate–distortion performance and low complexity. Fig. 13 compares the rate–distortion curves of four DL image compression models trained and evaluated on the Levir-CD train split. We compared two well-known methods [21], namely the Factorized Prior model and a custom reduced variant of the Scale Hyperprior model with only 150 latent channels, as well as the newer state-of-the-art method ELIC [24] and its smaller variant. The Factorized Prior consistently lags behind the others at both low and high bitrates, yielding unsatisfactory reconstruction quality and was therefore excluded from further consideration. ELIC achieves the highest PSNR among the four models, but at the expense of a large model size and FLOPs count (see Table IV). The smaller version of ELIC reduces computational complexity, yet still requires more than twice the FLOPs of the reduced variant of the Scale Hyperprior. Finally, the reduced variant of the Scale Hyperprior delivers a competitive rate–distortion curve, closely matching ELIC small and approaching standard ELIC, while using only 62 GFLOPs and 7M parameters, making it the best choice for our onboard compression module.

TABLE III: F1-score on unregistered Levir-CD dataset. Two configurations are proposed: *Reg-CD*, where registration and CD models are jointly finetuned after their independent pretrainings, and *Solo*, where registration and CD models are independently pretrained and tested without further finetuning.

Registration method	Params	FLOPs	F1-score
Light2Reg (Solo)	1.06M	45.18G	55.87 (± 1.59)
Light2Reg (Reg–CD)	1.06M	45.18G	70.30 (± 1.23)
SuperPoint + LightGlue (Solo)	13.15M	>357G	70.90 (± 0.91)
SuperPoint + LightGlue (Reg–CD)	13.15M	>357G	71.20 (± 1.07)

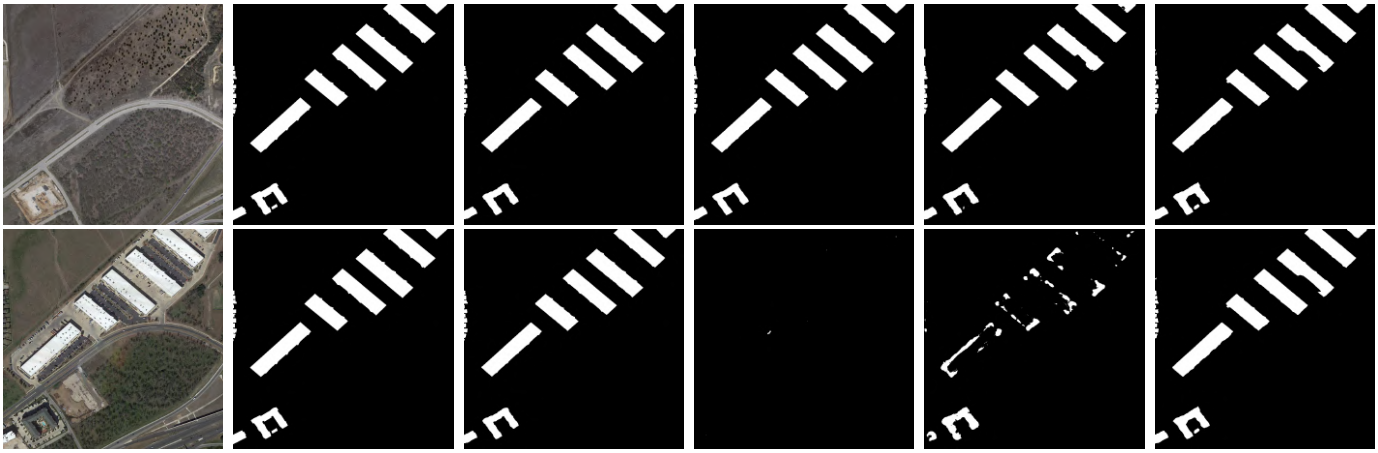
TABLE IV: Complexity of compression models.

Compression method	Params	FLOPs
Custom Scale Hyperprior [21]	6.96M	62.23G
Factorized Prior [21]	3.00M	44.28G
ELIC (small) [24]	26.63M	168.82G
ELIC [24]	35.42M	314.56G

D. CAD complexity and ablations on training strategies

Fig. 14 and Table V analyze the trade-offs between different variants of the CAD framework. The black curve (CAD-Features) corresponds to the standard configuration (with TieCD-L) already presented in the main results in Fig. 7.

As described in Sec. III-A, CAD operates primarily in the feature space, with the sole exception of the registration module, which receives the full-resolution images to regress the homography. The CAD-Pixels (red curve) design presents a variant in which the CD module uses an input in the full-resolution pixel space. This is more expensive as it requires extra layers and operations, which are not needed when directly passing deep features from the compression module, resulting in a considerable increase in computational complexity of over 45 GFLOPs and nearly twice the inference time. However, as seen in Fig. 14, the gain in F1-score is negligible, with both curves nearly overlapping across all rates.



(a) $(\mathbf{x}_{t_1},\mathbf{x}_{t_2})$ (b) Ground Truth (c) TieCD-L (d) BAN (e) TinyCD (f) ChangerEx

Fig. 11: Qualitative comparison across different CD methods and temporal orders on Levir-CD dataset. Top row: forward $(\mathbf{x}_{t_1}, \mathbf{x}_{t_2})$. Bottom row: forward $(\mathbf{x}_{t_2},\mathbf{x}_{t_1})$.

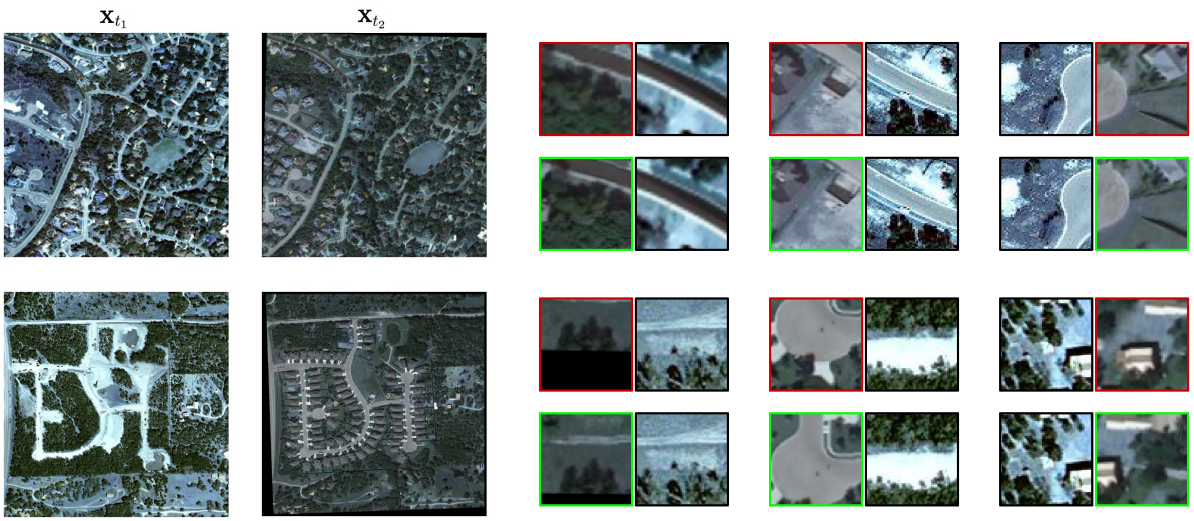


Fig. 12: Qualitative co-registration results on two misaligned Levir-CD image pairs. For each pair, the **black border** patches are cropped from the original reference frame $\mathbf{x}_{t_1}$. The **red border** patches show the result of warping transformed image $\mathbf{x}_{t_2}$ using our registration network after independent pretraining (*Solo* setup), while the **green border** patches show the warp obtained after joint train with the CD module (*Reg–CD* setup). Joint training yields noticeably better alignment, especially around building and road edges, demonstrating how the downstream CD objective refines the geometric estimates.

TABLE V: CAD computational complexity in pixel and feature spaces. Inference time and memory usage are computed on a Jetson Orin Nano 8GB. All the metrics refer to inference using the entire CAD framework, including all three submodules.

Domain	Params	FLOPs	Inference time (seconds)	Memory usage (MB)
Pixel	8.83M	166.39G	0.734 ± 0.001	6829 ± 8
Feature	8.81M	121.47G	0.383 ± 0.001	5729 ± 9

To quantify the specific benefit of the coding-for-machines compression, we additionally report CAD-Fixed Compression (blue curve), where the compression model is pretrained with a standard rate–distortion objective and then kept frozen during the subsequent end-to-end CAD optimization. Compared to CAD-Features, this baseline isolates the effect of jointly optimizing the compressor for CD accuracy: end-to-end task-driven compression yields a consistent improvement in F1-score, showing that the learned bit allocation better preserves information useful for change detection rather than solely maximizing visual reconstruction quality.

Additionally, we analyze the performance when no end-to-end finetuning is performed, i.e., we rely on independent modules pretrained for the compression, registration and change detection tasks. This is reported in Fig. 14 as the CAD-No Joint Training variant (green curve). It can be noticed that this approach leads to significantly degraded performance, with low F1-scores even at high rates. This highlights the importance of joint training: simply combining independently

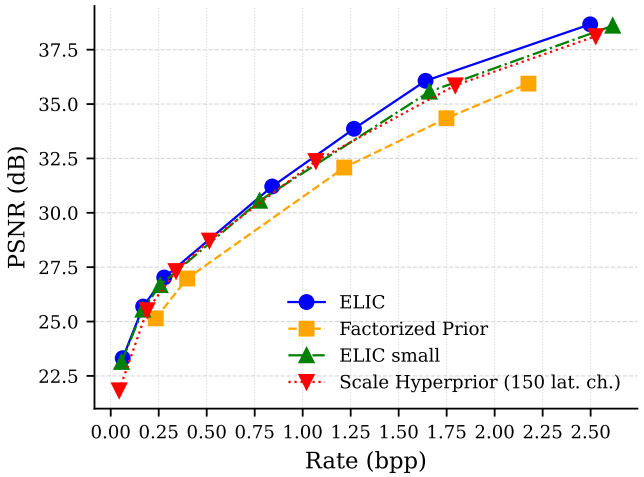


Fig. 13: Rate–distortion curves of compression models trained and tested on Levir-CD dataset.

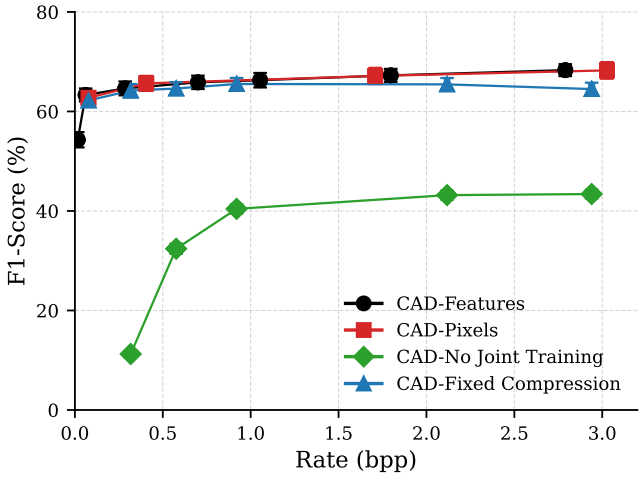


Fig. 14: Rate–F1 curves for CAD framework operating in feature space (CAD-Features), a variant working in pixel space (CAD-Pixels), a baseline where all modules are independently trained without end-to-end finetuning (CAD-No Joint Training), and a baseline with *fixed* rate–distortion compression (CAD-Fixed Compression), where the compressor is pretrained with a rate–distortion objective and then frozen during Reg–CD / end-to-end training, on Levir-CD dataset.

optimized modules is insufficient to achieve effective onboard CD.

Furthermore, to evaluate the impact of our proposed pretraining strategy (Sec. III), we compare two training regimes: (1) pretraining each module individually before joint finetuning of the whole CAD framework, and (2) training the entire CAD framework from random initialization using only the combined loss of Eq. (28). As shown in Fig. 15, the pretrained variant consistently outperforms the scratch baseline at all bitrates, achieving higher F1-scores for a given bpp. In addition, the curve from the pretraining baseline is smoother, indicating stable behavior. In contrast, without any pretraining

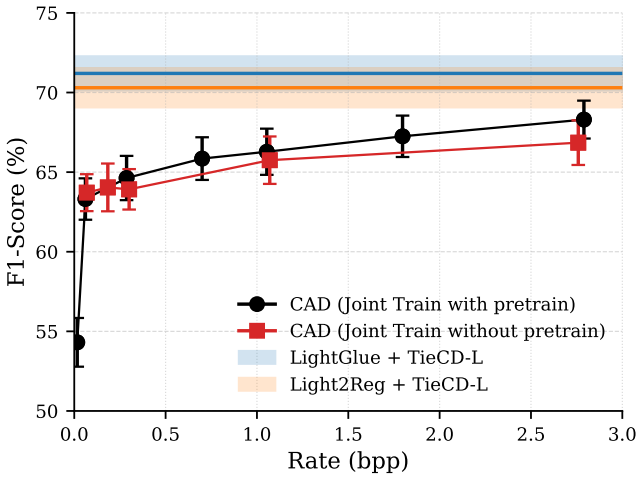


Fig. 15: Rate–F1 curves comparing CAD with initial module-wise pretraining (black) against the same framework trained end-to-end from scratch (red), on Levir-CD dataset.

TABLE VI: α sweep at fixed rate (3bpp) on Levir-CD.

α	**F1**
0.1	61.28 ($\pm$ 1.4)
0.3	68.30 ($\pm$ 1.2)
0.5	68.25 ($\pm$ 1.6)
0.7	65.78 ($\pm$ 1.2)
0.9	64.15 ($\pm$ 1.9)

strategy CAD shows larger fluctuations and slower, less reliable convergence, underlining the practical importance of our modular pretraining approach.

E. α parameter ablations

One of the most impactful parameters of the training procedure is α, which balances CD loss and co-registration loss in Eq. (28). In this section, we evaluate the sensitivity of CAD to different values of α at a fixed rate. In particular, we sweep $\alpha \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$ at a fixed rate of approximately 3 bpp. Results are presented in Table VI in terms of the final CD F1-score. They show that the F1-score peaks for $\alpha \in [0.3, 0.5]$, confirming that this range best balances the CD loss and the registration loss in Eq. (28). Very small values (e.g., $\alpha = 0.1$) prioritize the registration term with respect to the change detection loss, resulting in a significantly less discriminative CD module. Conversely, large values (e.g., $\alpha \geq 0.7$) under-constrain the registration module, leaving residual misalignment that degrades boundaries and reduces CD quality. We adopt $\alpha = 0.3$ in all our experiments since it provides a slightly higher F1-score and a smaller standard deviation.

VI. Conclusions

We presented a novel framework that for the first time addresses the problem of change detection directly onboard satellites. We showed that issues like the lack of image registration and the limited storage available onboard require the

development of an integrated framework that jointly considers compression, registration and CD. By developing a single lightweight neural network model for the entire framework and optimizing it end-to-end, we showed that compelling CD performance at significant compression ratios can be achieved. Importantly, we have found that the joint training allows employing a low-complexity co-registration module, allowing to obtain a low-complexity design that is suitable for fast inference on low-power devices and has been experimentally tested on one such device. On the other hand, we also note some limitations and possible future directions. First, due to the lack of publicly available, truly non-orthorectified benchmarks, our evaluation relies on strong projective distortions; correspondingly, the global homography used for alignment is an approximation that does not model terrain relief or more complex distortions. Second, hardware constraints may be tighter in orbit than in our 15 W embedded evaluation, calling for further porting, quantization, and scheduling on even lower-power processors. Third, our experiments focus on RGB imagery; extending CAD framework to multispectral/hyperspectral and SAR will likely require adapting the compression, registration, and CD modules (and possibly moving beyond a single homography, e.g., to piecewise or depth-aware models).

We believe that with this work we have taken a first big step in the direction of onboard real-time processing. We strongly encourage the scientific community to release raw non-orthorectified image datasets, to further evolve this area of research.

REFERENCES

[1] G. Giuffrida, L. Fanucci, G. Meoni, M. Batič, L. Buckley, A. Dunne, C. van Dijk, M. Esposito, J. Hefele, N. Vercruyssen, G. Furano, M. Pastena, and J. Aschbacher, "The phi-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, 2022.

[2] G. Meoni, R. Del Prete, F. Serva, A. De Beusscher, O. Colin, and N. 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*, 2024.

[3] A. Marin, C. Coelho, F. Deconinck, I. Babkina, N. Longepe, and M. Pastena, "Phi-sat-2: Onboard ai apps for earth observation," *Proc. Space Artif. Intell*, vol. 6, 2021.

[4] 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.

[5] 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.

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

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

[8] Y. Xing, J. Jiang, J. Xiang, E. Yan, Y. Song, and D. Mo, "Lightcdnet: Lightweight change detection network based on vhr images," *IEEE Geoscience and Remote Sensing Letters*, vol. 20, pp. 1–5, 2023.

[9] Z. Li, C. Tang, X. Liu, W. Zhang, J. Dou, L. Wang, and A. Y. Zomaya, "Lightweight remote sensing change detection with progressive feature aggregation and supervised attention," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 61, pp. 1–12, 2023.

[10] A. Codegoni, G. Lombardi, and A. Ferrari, "Tinycd: A (not so) deep learning model for change detection," *Neural Computing and Applications*, vol. 35, no. 11, pp. 8471–8486, 2023.

[11] G. Zhao, L. Shan, and W. Wang, "End-to-end remote sensing change detection of unregistered bi-temporal images for natural disasters," in *International Conference on Artificial Neural Networks*. Springer, 2023, pp. 259–270.

[12] W. Jing, K. Chi, Q. Li, and Q. Wang, "Changerd: A registration-integrated change detection framework for unaligned remote sensing images," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 220, pp. 64–74, 2025.

[13] Z. Zheng, A. Ma, L. Zhang, and Y. Zhong, "Change is everywhere: Single-temporal supervised object change detection in remote sensing imagery," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 15 193–15 202.

[14] Z. Zheng, Y. Zhong, S. Tian, A. Ma, and L. Zhang, "Changemask: Deep multi-task encoder-transformer-decoder architecture for semantic change detection," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 183, pp. 228–239, 2022.

[15] Z. Zheng, Y. Zhong, L. Zhang, and S. Ermon, "Segment any change," *arXiv preprint arXiv:2402.01188*, 2024.

[16] S. Fang, K. Li, and Z. Li, "Changer: Feature interaction is what you need for change detection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 61, pp. 1–11, 2023.

[17] G. Inzerillo, D. Valsesia, E. Magli, and A. Fiengo, "Towards storage-aware onboard change detection," in *IGARSS 2025-2025 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2025.

[18] I. Blanes, E. Magli, and J. Serra-Sagrista, "A tutorial on image compression for optical space imaging systems," *IEEE Geoscience and Remote Sensing Magazine*, vol. 2, no. 3, pp. 8–26, 2014.

[19] M. Hernández-Cabronero, A. B. Kiely, M. Klimesh, I. Blanes, J. Ligo, E. Magli, and J. Serra-Sagristà, "The ccsds 123.0-b-2 "low-complexity lossless and near-lossless multispectral and hyperspectral image compression" standard: A comprehensive review," *IEEE Geoscience and Remote Sensing Magazine*, vol. 9, no. 4, pp. 102–119, 2021.

[20] J. Ballé, V. Laparra, and E. P. Simoncelli, "End-to-end optimized image compression," *arXiv preprint arXiv:1611.01704*, 2016.

[21] J. Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston, "Variational image compression with a scale hyperprior," *arXiv preprint arXiv:1802.01436*, 2018.

[22] Z. Cheng, H. Sun, M. Takeuchi, and J. Katto, "Learned image compression with discretized gaussian mixture likelihoods and attention modules," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 7939–7948.

[23] B. Bross, Y.-K. Wang, Y. Ye, S. Liu, J. Chen, G. J. Sullivan, and J.-R. Ohm, "Overview of the versatile video coding (vvc) standard and its applications," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 10, pp. 3736–3764, 2021.

[24] D. He, Z. Yang, W. Peng, R. Ma, H. Qin, and Y. Wang, "Elic: Efficient learned image compression with unevenly grouped space-channel contextual adaptive coding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5718–5727.

[25] G. Pilikos, M. Azcueta, R. Camarero, and N. Floury, "Raw sar data compression with deep learning," in *IGARSS 2024-*

2024 IEEE International Geoscience and Remote Sensing Symposium. IEEE, 2024, pp. 2546–2549.

[26] 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*, 2024.

[27] D. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, vol. 2, 1999, pp. 1150–1157 vol.2.

[28] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superglue: Learning feature matching with graph neural networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 4938–4947.

[29] P. Lindenberger, P.-E. Sarlin, and M. Pollefeys, "Lightglue: Local feature matching at light speed," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 17 627–17 638.

[30] Y. Ye, T. Tang, B. Zhu, C. Yang, B. Li, and S. Hao, "A multiscale framework with unsupervised learning for remote sensing image registration," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–15, 2022.

[31] R. Zhou, D. Quan, S. Wang, C. Lv, X. Cao, J. Chanussot, Y. Li, and L. Jiao, "A unified deep learning network for remote sensing image registration and change detection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 62, pp. 1–16, 2024.

[32] P. R. Coppin and M. E. Bauer, "Digital change detection in forest ecosystems with remote sensing imagery," *Remote sensing reviews*, vol. 13, no. 3-4, pp. 207–234, 1996.

[33] R. D. Johnson and E. Kasischke, "Change vector analysis: A technique for the multispectral monitoring of land cover and condition," *International journal of remote sensing*, vol. 19, no. 3, pp. 411–426, 1998.

[34] T. Celik, "Unsupervised change detection in satellite images using principal component analysis and k-means clustering," *IEEE geoscience and remote sensing letters*, vol. 6, no. 4, pp. 772–776, 2009.

[35] V. Walter, "Object-based classification of remote sensing data for change detection," *ISPRS Journal of photogrammetry and remote sensing*, vol. 58, no. 3-4, pp. 225–238, 2004.

[36] B. Desclée, P. Bogaert, and P. Defourny, "Forest change detection by statistical object-based method," *Remote sensing of environment*, vol. 102, no. 1-2, pp. 1–11, 2006.

[37] C. Huang, K. Song, S. Kim, J. R. Townshend, P. Davis, J. G. Masek, and S. N. Goward, "Use of a dark object concept and support vector machines to automate forest cover change analysis," *Remote sensing of environment*, vol. 112, no. 3, pp. 970–985, 2008.

[38] J. Im and J. R. Jensen, "A change detection model based on neighborhood correlation image analysis and decision tree classification," *Remote Sensing of Environment*, vol. 99, no. 3, pp. 326–340, 2005.

[39] J. Barber, "A generalized likelihood ratio test for coherent change detection in polarimetric sar," *IEEE Geoscience and Remote Sensing Letters*, vol. 12, no. 9, pp. 1873–1877, 2015.

[40] D. Ciuonzo, V. Carotenuto, and A. De Maio, "On multiple covariance equality testing with application to sar change detection," *IEEE Transactions on Signal Processing*, vol. 65, no. 19, pp. 5078–5091, 2017.

[41] R. C. Daudt, B. Le Saux, and A. Boulch, "Fully convolutional siamese networks for change detection," in *2018 25th IEEE international conference on image processing (ICIP)*. IEEE, 2018, pp. 4063–4067.

[42] H. Chen and Z. Shi, "A spatial-temporal attention-based method and a new dataset for remote sensing image change detection," *Remote Sensing*, vol. 12, no. 10, p. 1662, 2020.

[43] W. G. C. Bandara and V. M. Patel, "A transformer-based siamese network for change detection," in *IGARSS 2022-2022 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2022, pp. 207–210.

[44] Y. Liu, F. Liu, J. Liu, and L. Xiao, "Edge-object co-driven learning for remote sensing change detection," *IEEE Transactions on Geoscience and Remote Sensing*, 2025.

[45] D. Wang, J. Zhang, M. Xu, L. Liu, D. Wang, E. Gao, C. Han, H. Guo, B. Du, D. Tao *et al.*, "Mtp: Advancing remote sensing foundation model via multi-task pretraining," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2024.

[46] K. Li, X. Cao, and D. Meng, "A new learning paradigm for foundation model-based remote-sensing change detection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 62, pp. 1–12, 2024.

[47] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *2011 International Conference on Computer Vision*, 2011, pp. 2564–2571.

[48] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superpoint: Self-supervised interest point detection and description," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2018, pp. 224–236.

[49] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.

[50] B. Zhu, L. Zhou, S. Pu, J. Fan, and Y. Ye, "Advances and challenges in multimodal remote sensing image registration," *IEEE Journal on Miniaturization for Air and Space Systems*, vol. 4, no. 2, pp. 165–174, 2023.

[51] Y. Ye, L. Bruzzone, J. Shan, F. Bovolo, and Q. Zhu, "Fast and robust matching for multimodal remote sensing image registration," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 11, pp. 9059–9070, 2019.

[52] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*. Springer, 2015, pp. 234–241.

[53] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[54] D. Misra, "Mish: A self regularized non-monotonic activation function," *arXiv preprint arXiv:1908.08681*, 2019.

[55] H. Choi and I. V. Bajić, "Scalable image coding for humans and machines," *IEEE Transactions on Image Processing*, vol. 31, pp. 2739–2754, 2022.

[56] X. Ye. (2023) Calflops: a flops and params calculate tool for neural networks in pytorch framework. [Online]. Available: https://github.com/MrYxJ/calculate-flops.pytorch

[57] E. Aybar, L. Ysuhuaylas, J. Loja, K. Gonzales, F. Herrera, L. Bautista, R. Yali, A. Flores, L. Diaz, N. Cuenca *et al.*, "Cloudsen12, a global dataset for semantic understanding of cloud and cloud shadow in sentinel-2," *Scientific Data*, vol. 9, no. 1, p. 782, 2022.

[58] S. Ji, S. Wei, and M. Lu, "Fully convolutional networks for multisource building extraction from an open aerial and satellite imagery data set," *IEEE Transactions on geoscience and remote sensing*, vol. 57, no. 1, pp. 574–586, 2018.

[59] Q. Shi, M. Liu, S. Li, X. Liu, F. Wang, and L. Zhang, "A deeply supervised attention metric-based network and an open aerial image dataset for remote sensing change detection," *IEEE Transactions on Geoscience and Remote Sensing*, pp. 1–16, 2021.

[60] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014. [Online]. Available: https://api.semanticscholar.org/CorpusID:6628106

[61] J. Bégaint, F. Racapé, S. Feltman, and A. Pushparaja, "Compressai: a pytorch library and evaluation platform for end-to-end compression research," *arXiv preprint arXiv:2011.03029*, 2020.

[62] K. Li, J. Jiang, A. Codegoni, C. Han, Y. Deng, K. Chen, Z. Zheng, H. Chen, Z. Liu, Y. Gu, Z. Zou, Z. Shi, S. Fang,

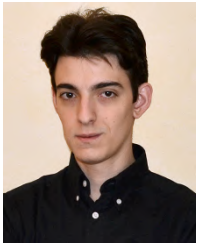
D. Meng, Z. Wang, and X. Cao, "Open-CD: A comprehensive toolbox for change detection," *arXiv preprint arXiv:2407.15317*, 2024.

[63] C. Han, C. Wu, H. Guo, M. Hu, and H. Chen, "Hanet: A hierarchical attention network for change detection with bitemporal very-high-resolution remote sensing images," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 16, pp. 3867–3878, 2023.

[64] K. Chen, C. Liu, W. Li, Z. Liu, H. Chen, H. Zhang, Z. Zou, and Z. Shi, "Time travelling pixels: Bitemporal features integration with foundation model for remote sensing image change detection," *arXiv preprint arXiv:2312.16202*, 2023.



Gabriele Inzerillo received the M.Sc. in Computer Engineering from the Politecnico di Torino in 2022. He is currently pursuing a Ph.D. on deep learning for remote sensing, as a result of a collaboration between Politecnico di Torino and the European Space Agency, within the OSIP Framework.



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.



Aniello Fiengo received a degree in Telecommunication Engineering from the University of Naples Federico II and a Ph.D. in Image and Signal Processing from Institut Mines-Télécom Télécom ParisTech, Paris, in 2016. Currently, at ESA-ESTEC, he oversees the definition and implementation of onboard processing algorithms for Earth Observation missions and serves as Technical Officer for multiple R&D activities.



Enrico Magli (S'97-M'01-SM'07-F'17)(IEEE Fellow) received the M.Sc. and Ph.D. degrees from the Politecnico di Torino, Italy, in 1997 and 2001, respectively. He is currently a Full Professor with Politecnico di Torino, Italy, where he leads the Image Processing and Learning group, performing research in the fields of deep learning for image and video processing, image compression and image forensic for multimedia and remote sensing applications. He is a Senior Associate Editor of IEEE Journal on Selected Topics in Signal Processing, and a former Associate Editor of the EURASIP Journal on Image and Video Processing, the IEEE TRANSACTIONS ON MULTIMEDIA and the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY. He is a Fellow of the IEEE, a Fellow of the ELLIS Society for the advancement of artificial intelligence in Europe, and has been an IEEE Distinguished Lecturer from 2015 to 2016. 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 (as senior author), the IEEE ICIP 2019 Best Paper Award, the IEEE Multimedia 2019 Best Paper Award, and the 2010 and 2014 Best Associate Editor Award of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY.