JGR | Machine Learning and Computation

RESEARCH ARTICLE
10.1029/2025JH001023

Comparing Solar Structure Detection Methods in SDO/AIA Observations and the Application to Raw Uncalibrated Data

P. Gonidakis[1], F. Carella[1], E. Dineva[1], H-J. Jeong[1,2], P. Antunes[3], A. Podobas[3], S. Raptis[4], V. Toy-Edens[4], M. Jin[5], S. Poedts[1,6], J. Magdalenic[1,7], and G. Miloshevich[1]

[1]Centre for mathematical Plasma Astrophysics (CmPA), KULeuven, Leuven, Belgium, [2]School of Space Research, Kyung Hee University, Yongin, Republic of Korea, [3]KTH Royal Institute of Technology, Stockholm, Sweden, [4]Johns Hopkins Applied Physics Laboratory, Laurel, MD, USA, [5]Lockheed Martin Solar and Astrophysics Lab, Palo Alto, CA, USA, [6]Institute of Physics, University of Maria Curie-Skłodowska, Lublin, Poland, [7]Solar-Terrestrial Centre of Excellence, Royal Observatory of Belgium, Uccle, Belgium

Abstract Recent advances in solar physics increasingly rely on automated identification of coronal structures using machine learning. Yet most studies emphasize scientific performance without evaluating feasibility for onboard deployment to prioritize downlink observations. This work investigates the automated identification of active regions and coronal holes by applying segmentation and detection techniques to Solar Dynamics Observatory (SDO) data. We compare three approaches: SCSS-Net, a deep learning model for semantic segmentation; YOLOv8n, a lightweight object detector; and a traditional pipeline based on basic computer vision operations (BCVO). Each method is assessed for its scientific accuracy and its suitability for deployment in future resource-limited missions. While no direct hardware benchmarking is performed in this study, we assess the feasibility of onboard implementation based on the associated number of trainable parameters, architecture and hardware requirements. Training and evaluation are first conducted on well-calibrated SDO images. We then extend the evaluation to raw and uncalibrated SDO images affected by instrumental artifacts. Performance is measured using the Intersection over Union (IoU) and Dice score. Results show that while SCSS-Net achieves the highest segmentation quality, YOLOv8n offers a strong balance between accuracy and efficiency. The BCVO pipeline remains viable under strict hardware limitations. Interestingly, our models retain compatibility on Level-0 observations. This is the first study comparing these widely used methods from the perspective of onboard deployment. Our findings provide a foundation for designing frameworks tailored to onboard hardware configurations.

Plain Language Summary Solar physicists are increasingly using machine learning to automatically identify features on the Sun. Most studies, however, only measure accuracy and do not consider whether these methods could run on spacecraft with limited computing power. In this work, we investigate the detection of two important solar features: active regions and coronal holes. We test three approaches: a deep learning model (SCSS-Net), a lightweight object detector (YOLOv8n), and a simpler method based on basic computer vision operations (BCVO). We compare them in terms of accuracy, efficiency, and suitability for future missions with strict hardware limits, such as Field-Programmable Gate Array-based systems. Training and evaluation are first performed on well-processed Solar Dynamics Observatory (SDO) data (Level-2). We then extend testing to raw, uncalibrated images (Level-0) that include real instrument artifacts. Results show that SCSS-Net achieves the highest accuracy, YOLOv8n provides a good balance between accuracy and speed, and BCVO remains useful under the tightest hardware constraints. Importantly, models trained on Level-2 data also perform well on Level-0 observations. While no direct hardware benchmarking is performed, this is the first study directly comparing these methods for onboard use providing a foundation for designing frameworks tailored to onboard hardware configurations.

Key Points:

- Onboard solar coronal structure detection enables spacecraft to flag, reduce, or prioritize key observations for downlink under restrictions
- We find that SCSS-Net leads in accuracy, YOLOv8n balances speed and quality, and basic computer vision operations remains effective under strict hardware limits
- Models trained on Level 2 Solar Dynamics Observatory (SDO) data also work on Level 0 SDO, showing potential for autonomous use in future missions

Correspondence to:
P. Gonidakis,
panagiotis.gonidakis@kuleuven.be

Citation:
Gonidakis, P., Carella, F., Dineva, E., Jeong, H-J., Antunes, P., Podobas, A., et al. (2026). Comparing solar structure detection methods in SDO/AIA observations and the application to raw uncalibrated data. *Journal of Geophysical Research: Machine Learning and Computation*, *3*, e2025JH001023. https://doi.org/10.1029/2025JH001023

Received 19 SEP 2025
Accepted 10 APR 2026

Author Contributions:
Conceptualization: P. Gonidakis, G. Miloshevich
Data curation: P. Gonidakis, S. Raptis, V. Toy-Edens, M. Jin
Formal analysis: P. Gonidakis
Funding acquisition: A. Podobas, S. Poedts, J. Magdalenic
Investigation: P. Gonidakis, P. Antunes
Methodology: P. Gonidakis, H-J. Jeong
Project administration: G. Miloshevich
Resources: P. Gonidakis, G. Miloshevich
Software: P. Gonidakis

1. Introduction

Spacecraft, both human-crewed and unmanned, play a central role in the exploration and utilization of space. Unmanned missions, ranging from satellites and probes to robotic landers, are built to operate autonomously or be remotely controlled, often enduring harsh environments for extended periods without direct human involvement. The hardware and software onboard these systems are constrained by space restrictions, including limited communication bandwidth, a limited power supply, and radiation-induced errors.

Supervision: G. Miloshevich
Validation: P. Gonidakis
Visualization: P. Gonidakis
Writing – original draft: P. Gonidakis
Writing – review & editing: F. Carella, E. Dineva, H-J. Jeong, P. Antunes, S. Raptis, V. Toy-Edens, M. Jin, S. Poedts, J. Magdalenic, G. Miloshevich

The adoption of Artificial Intelligence (AI) in spacecraft onboard computing presents a promising solution to challenges posed by rapid technological evolution and the vast, ever-growing volumes of data generated by spaceborne sensors. AI can enable autonomous onboard decision-making, such as processing imagery to filter out non-essential data. A practical example is CloudSatNet-1 (Pitonak et al., 2022), which reduces data transmission load by automatically discarding images containing clouds, thereby improving the efficiency of satellite-to-ground communication. AI can also support real-time diagnostics and responses to issues with spacecraft systems (Gallon et al., 2025; Ramachandran et al., 2020). Instruments can be intelligently activated thanks to context-aware AI driver algorithms.

A step forward in autonomous spacecraft operations is the ability to closely monitor solar activity. Phenomena such as solar flares and coronal mass ejections (CMEs) can pose significant risks to space missions (Schrijver & Siscoe, 2010; Webb & Howard, 2012). Segmenting or detecting coronal structures onboard can help address these challenges. Active regions (ARs), identified by enhanced extreme ultraviolet (EUV), X-ray emission and often identified in magnetograms, are the primary sites for flare initiation due to their strong and complex magnetic fields (Benz, 2017; Toriumi & Wang, 2019). Coronal holes (CHs), which appear as dark regions in EUV images due to lower plasma density and temperature, are recognized as sources of high-speed solar wind streams that contribute to recurrent geomagnetic disturbances (Cranmer, 2002). Both active regions and coronal holes are phenomena occurring in the solar atmosphere, spanning through its layers (photosphere, chromosphere and corona). Since they are both very prominent in the solar corona, for this study, we adopt the term 'solar coronal structures'. The accurate segmentation and tracking of these structures allows us to monitor their evolution and assess the likelihood of eruptive events.

Beyond scientific motivations, there are also practical motivations for integrating such capabilities. Onboard data storage and downlink bandwidth limitations often necessitate the selective discarding of data. Intelligent algorithms could prioritize samples based on their scientific relevance. Moreover, due to power constraints and operational considerations, high-resolution telescopes, such as the Extreme Ultraviolet Imager (EUI)/High Resolution Imager (HRI) onboard the Solar Orbiter mission (Auchère et al., 2020), must often remain idle or point away from target regions. In contrast with the SDO mission, where continuous data downlink is always possible, the Solar Orbiter mission could benefit from such capabilities. Enabling autonomous activation and targeting of such instruments is therefore of great importance.

Tracking filaments is also important due to their connection with CMEs and solar eruptions (Schmieder et al., 2013). However, for simplicity in this work, we only focus on active regions and coronal holes since filament identification typically requires information from different wavelengths than ARs and CHs (304 Å).

Numerous automated approaches for segmenting active regions and coronal holes have been proposed to detect solar activity, with significant advancements over time. Early methods relied on predefined rule-based systems (Henney & Harvey, 2005; Krista & Gallagher, 2009; Pérez-Suárez et al., 2013), followed by more sophisticated mathematical algorithms (Verbeeck et al., 2014). These were later complemented by computer vision techniques and traditional machine learning models (Delouille et al., 2018; Reiss et al., 2015), and more recently, by the adoption of large deep learning models with convolutional architectures comprising millions of parameters (Mackovjak et al., 2021). Various methods are also developed by extracting features from Active Region patches from magnetograms (Bobra et al., 2014, 2021). In general, Asensio Ramos et al. (2023) provides an extensive review of recent developments of machine learning in solar physics, including coronal hole segmentation.

It is important to note that large deep learning architectures typically require substantial amounts of data. A variety of high-quality data sets are available for the observation and analysis of solar coronal structures, such as active regions, coronal holes, and bright points, with observations from the Solar Dynamics Observatory (SDO) at the forefront (Pesnell et al., 2012). Additional resources include legacy data sets from missions, such as SOHO/EIT (Delaboudiniere et al., 1995; Domingo et al., 1995), which offer long-term coronal imaging from 1996 to 2010, and STEREO/EUVI (Howard et al., 2008), which provide stereoscopic views of the Sun for 3D analysis of coronal structures. More recently, Solar Orbiter's EUI (Müller et al., 2020; Rochus et al., 2020) has begun delivering extremely high-resolution EUV images from unique vantage points in the inner heliosphere, ideal for focused studies of fine-scale coronal dynamics.

A key challenge arises when considering the execution of such algorithms onboard spacecraft systems, where the available input is raw data, which is typically noisy and uncalibrated, often referred to as Level-0 data. In contrast,

machine learning models are usually trained on Level-2 data, which is fully calibrated, ready for scientific analysis, and provides a standardized, high-quality representation of the phenomena under investigation (Boerner et al., 2012). In principle, Level-0 data consist of raw pixel values with no corrections applied, while Level-1 data includes basic corrections for instrument effects but may still lack full alignment or calibration. The transformation from Level-0 to Level-2 can be computationally expensive, often requiring the data to be processed in chunks and handled differently depending on the specific characteristics of each sample. Despite these limitations and challenges, several prototype systems have already demonstrated the potential of onboard AI, particularly for tasks such as image encoding and compression (Guerrisi et al., 2023; Ržička et al., 2023).

Deploying AI in space is challenging due to high computational demands and the unsuitability of conventional processors, such as CPUs and GPUs, which are not designed for radiation-prone environments (Ghiglione & Serra, 2022). Field-Programmable Gate Arrays (FPGAs) offer a compelling alternative. As integrated circuits that can be programmed and reprogrammed after manufacture, FPGAs enable adaptable and efficient AI acceleration. This reconfigurability is particularly valuable for space missions, allowing for in-flight updates, corrections, or even complete hardware logic changes as mission requirements evolve. In addition to FPGAs, embedded edge-AI platforms such as NVIDIA Jetson (NVIDIA Corporation, 2023) and Intel Movidius (Myriad) (Intel Corporation, 2023) processors have been widely adopted for terrestrial and near-space applications, offering a balance between computational performance, power efficiency, and software maturity for deep learning inference. However, their use in long-duration space missions remains limited by radiation tolerance and qualification constraints, making FPGAs and radiation-hardened accelerators more suitable for fully onboard deployment.

This flexibility gives FPGAs a distinct advantage over Application-Specific Integrated Circuits (ASICs), which are fixed after production and cannot be modified once a satellite is launched. The design and production of an ASIC is also more time-consuming and expensive, making it less suitable for space applications that rarely require mass production and often have tight development schedules (Boada Gardenyes, 2011). Although FPGAs are generally more complex to program, their unique benefits have led vendors like Xilinx, Microchip, and NanoXplore to offer radiation-hardened FPGAs specifically designed for the rigors of space. In this work, we discuss the feasibility of implementing our AI algorithms on these reconfigurable hardware systems.

Existing studies consistently emphasize three critical metrics: power efficiency, FPGA resource demands, and inference speed (Antunes & Podobas, 2025). Reiter et al. (2020) proposed an FPGA-based implementation of a Binarized Neural Network for cloud detection. The results showed that the FPGA implementation achieved lower power consumption and faster inference compared to its CPU and GPU counterparts, highlighting its superior energy efficiency —a critical consideration for computing systems in space applications. Cosmas and Kenichi (2020) conducted a comparison over three distinct methodologies for satellite-reliable pose estimation. The first method involved direct regression on the whole image to identify key points, employing a ResNet-50 (Koonce, 2021) model architecture. The second method utilized heatmap-based detection of key points across the entire image using a U-Net model architecture, an encoder-decoder architecture typically employed for segmentation, often applied to medical imaging (Ronneberger et al., 2015). Lastly, a method for spacecraft detection, cropping, and subsequent heatmap-based detection of key points is proposed, utilizing YOLOv3 (You Only Look Once), a widely used object detection method in computer vision applications (Redmon & Farhadi, 2018). They concluded that the ResNet34-U-Net model was the best compromise between performance, inference time, and power consumption.

Moreover, several variants of the YOLO architecture have been occasionally applied and evaluated for identifying coronal structures, particularly coronal holes. Using SDO data, they demonstrate the effectiveness and relevance of this approach for space weather monitoring, as well as its scalability to large data sets (Nandi et al., 2024). Specifically, the nano, medium, and extra-large (x) versions of YOLOv8 were evaluated using metrics such as F1 score, mean Average Precision (mAP), and True Positive Rate (TPR), with the nano model offering the most lightweight solution and the extra-large model being the most computationally demanding.

Building upon the previously discussed studies, we compare three distinct methodologies: segmentation using SCSS-Net (U-Net variant), detection using one of the lightest YOLO variants (YOLOv8n), and segmentation based on basic computer vision operations (BCVO). Each approach presents unique strengths and limitations. Our goal is to determine the best trade-off between accuracy, power efficiency, and inference time. In parallel, we investigate whether models trained on Level-2 SDO data can still perform well when applied to Level-0 data, a crucial step toward making these frameworks operate autonomously onboard a spacecraft. This is the first effort in

the space weather domain to compare such methods from multiple perspectives, including scientific performance, implementation simplicity, qualitative advantages and limitations, and suitability for onboard deployment using FPGAs.

This paper is structured as follows: Section 2 describes the data sets, algorithms, and evaluation metrics employed in this study. Section 3 presents the experimental results, while Section 4 discusses their interpretation, implications, and potential directions for future research toward on-board deployment of AI methods.

2. Data and Methods

2.1. The Data Set

2.1.1. SDO Machine Learning Ready Data Set

The SDO is a NASA mission launched in 2010 to observe the Sun continuously and at high resolution. It carries three instruments on board, the Atmospheric Imaging Assembly (AIA; Lemen et al., 2012), the Helioseismic and Magnetic Imager (HMI; Scherrer et al., 2012), and the Extreme Ultraviolet Variability Experiment (EVE; Woods et al., 2012)], which together provide a rich multi-modal data set for studying solar activity. AIA captures full-disk EUV/UV images of the Sun at 7 EUV channels (centered at 94, 131, 171, 193, 211, 304, and 335 Å) and 2 UV channels (centered at 1,600 and 1,700 Å), each corresponding to plasma emission from different temperature regimes in the solar atmosphere. Meanwhile, HMI provides measurements of the photospheric magnetic field as well as continuum intensity and Doppler velocity observations, enabling comprehensive studies of magnetic and dynamic processes on the solar surface.

Recognizing the need for structured and accessible data for AI research, Galvez et al. (2019) introduced the SDO Machine learning (SDOML) data set, a carefully curated and preprocessed machine learning-ready data set derived from SDO observations. Later, the SDOMLv2 data set was released, providing the data in the cloud-friendly Zarr format. By standardizing data preprocessing (e.g., alignment, normalization, cropping, and resizing), SDOMLv2 enables researchers to focus on model development and evaluation rather than handling raw data.

Although future missions are expected to employ more modern and efficient hardware, such as advanced CMOS sensors, developing and validating algorithms on SDO data remains highly relevant. SDO provides a long, consistent, and well-annotated observational record, which serves as a valuable testbed for building and benchmarking algorithms intended for future deployment.

2.1.2. Level-0 SDO

In most current research, machine learning models for solar image analysis are trained on well-preprocessed data sets, often described as Level-2 products from the SDO, or on derived data sets built upon them, such as the aforementioned SDOMLv2 data set. More specifically, these data products have undergone extensive preprocessing, including calibration, alignment, despiking, and correction for instrumental effects such as instrument degradation and other systematic errors (Boerner et al., 2012; Kasapis et al., 2023). As a result, Level-2 data provide clean, stable inputs ideal for supervised learning and model benchmarking. However, a major challenge arises when considering the onboard deployment of such models for future space missions: spacecraft typically have access only to Level-0 data, which are raw, uncalibrated telemetry streams directly from the instrument. These data include various instrumental artifacts, noise, and format differences that are absent from Level-2 products. Therefore, models trained on high-quality Level-2 data may fail or underperform when exposed to raw Level 0 inputs in operational settings. To bridge this gap, either onboard preprocessing pipelines must be developed to approximate Level-2 calibration, or models must be fine-tuned to become robust or be inherently generalizable to the noise and variability present in Level-0 data. Addressing this domain mismatch is essential for ensuring the reliability and autonomy of AI-powered systems deployed in space.

Calibrated level-1 scientific data of SDO/AIA are accessible from the Joint Science Operations Center (JSOC) at Stanford University (Hapgood et al., 1997). However, Level-0 data is currently not publicly available. Due to the large data volume, the JSOC retains only the most recent 30 days of Level-0 data on local disk storage, while older data is archived on tape. For this study, we obtained two Level-0 data sets directly from the instrument team. The first one consists of SDO/AIA 131 Å images from the following dates in December 2023: 11, 15, 20, and 31, with

10.1029/2025JH001023

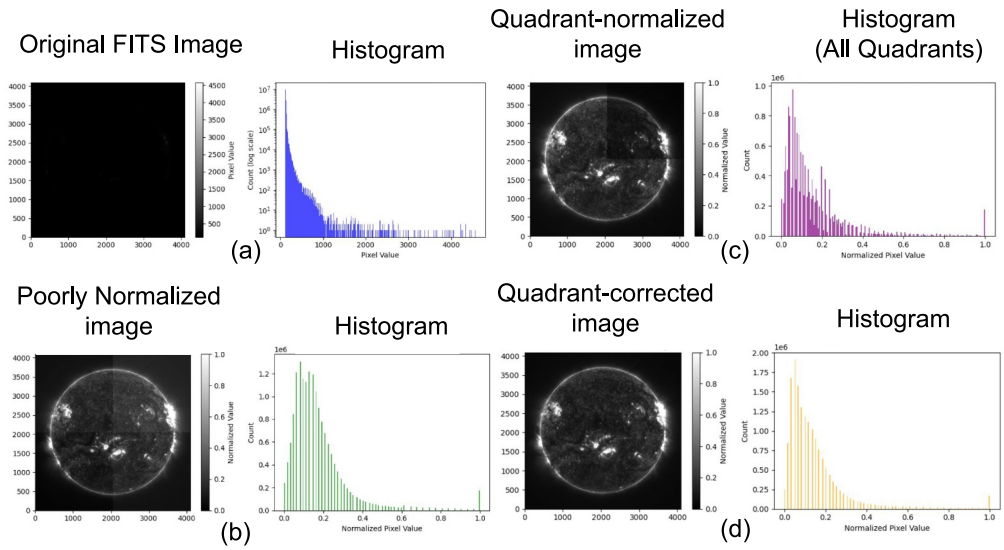

Figure 1. Illustrative examples of preprocessing a Level 0 SDO sample along the corresponding histogram: (a) Original raw sample, (b) preprocessing based on the full-image, where all four quadrant discontinuities are clearly visible, (c) per-quadrant preprocessing by clipping pixel values at the 1st and 99th percentiles, resulting in one remaining discontinuity, and (d) per-quadrant preprocessing by shifting each quadrant's median to match the global median, followed by percentile based normalization.

a cadence of 12 min. The second subset comprises 441 files from all SDO wavelengths, spanning a 2-week period with a 6-hr cadence, from 5 June 2025 to 19 June 2025.

Due to the sparsity of Level-0 data, we opt to use these samples exclusively for testing, without allocating any of them for training or fine-tuning. On the one hand, the limited amount of data is insufficient to support meaningful training or fine-tuning. On the other hand, preserving the entire data set for testing ensures a statistically representative sample, which is essential for a reliable and robust evaluation of model performance.

2.2. Preprocessing Level 0 SDO

Level-0 samples require preprocessing before being fed to the studied frameworks. This step is a critical part of any machine learning or deep learning pipeline as it transforms raw image data into a format that improves model performance and ensures consistency across the data set. Common preprocessing steps include resizing images to a fixed input size compatible with neural network architectures and normalizing pixel values—often scaling them to a range such as [0, 1] or $[-1, 1]$ to stabilize training. In parallel, to ensure compatibility with onboard deployment constraints, we deliberately keep preprocessing steps minimal.

In our case, firstly, the intensity pixel values need to be preprocessed and clipped to [0, 1]. The original Level-0 SDO images are stored with 14-bit depth, meaning that pixel values span the range from 0 to 16,383. When applying a global normalization to the [0, 1] range, distinct quadrant discontinuities become apparent. These artifacts arise due to the independent acquisition and compression of each quadrant of the CCD sensor. Even after applying full-image normalization combined with percentile-based clipping (1st and 99th percentiles), these discontinuities remain visible. To address this, we experimented with applying normalization separately to each quadrant. While this approach occasionally reduces the artifacts, in many cases, residual discontinuities persist in one or more quadrants. As a final correction, we shift the median intensity of each quadrant to match the global median before performing normalization. This method consistently suppresses the quadrant boundaries, resulting in visually continuous images across the whole frame. Figure 1 illustrates how the above preprocessing steps progressively address the quadrant discontinuities present in Level 0 SDO data.

The second preprocessing step involves resizing the images to match the input dimensions required by each framework. Both SCSS-Net and YOLOv8n are deep learning-based models that operate on fixed-size inputs. Accordingly, the original images are rescaled to 256 × 256 for SCSS-Net and 512 × 512 for YOLOv8n.

2.3. Solar Coronal Structure Masks

To enable the use of supervised machine learning methods for the segmentation or detection of solar coronal structures, it is crucial to acquire a significant and consistent set of labeled data.

Mackovjak et al. (2021) released a data set of binary, pixel-wise annotation masks for SDO/AIA observations, consisting of one sample per day from 2012 to 2018 and covering both active regions and coronal holes. To reduce annotation bias, masks were generated using multiple methods, including SPoCA (Verbeeck et al., 2014) (for both active regions and coronal holes), CHIMERA (Garton et al., 2018) and Region Growth (Tlatov et al., 2014) (for coronal holes), and were subsequently reviewed and supplemented with custom annotations by the authors.

2.3.1. SPoCA

One of the most practical and widely adopted approaches for generating such labels is through the use of the Spatial Possibilistic Clustering Algorithm (SPoCA; Verbeeck et al., 2014). SPoCA is a state-of-the-art segmentation method that operates on EUV images, most notably from the SDO/AIA 193 Å channel, and produces masks for key solar structures such as coronal holes and active regions. By combining fuzzy and possibilistic clustering, SPoCA offers robust performance across a wide range of solar conditions. Its output masks have been extensively used and validated in the solar physics community, often serving as a reference for benchmarking newly developed methods (Delouille et al., 2018). SPoCA's algorithm includes a square root transform, akin to an Anscombe transform (Anscombe, 1948), which has the property of approximately converting Poisson noise into Gaussian noise and a correction of the limb brightening effect inspired by Barra et al. (2009) and adapted by Verbeeck et al. (2014). Then, a segmentation is performed, resulting in three main classes: active regions (bright), quiet Sun (intermediate), and coronal holes (dark). The output is a set of probabilistic and binary masks indicating the location and extent of each class. Most importantly, SPoCA masks are publicly available through repositories such as the Heliophysics Event Knowledgebase (HEK; Hurlburt et al., 2012), making them an accessible and reliable source of training labels for data-driven approaches.

Undeniably, the SPoCA algorithm enables the automatic and efficient generation of large data sets of active regions (ARs) and coronal holes (CHs). However, this approach has several limitations that can potentially lead to biased data sets. For instance, under certain conditions SPoCA may underestimate the extent of a coronal hole, while in other cases it may erroneously include filament structures within coronal-hole regions (Jarolim et al., 2021; Reiss et al., 2015, 2021). For this reason, incorporating multiple mask sources is necessary to obtain a less biased data set.

2.3.2. CHIMERA

In CHIMERA (Coronal Hole Identification via Multi-thermal Emission Recognition Algorithm) (Garton et al., 2018), coronal-hole (CH) masks are produced by combining multi-thermal EUV intensity segmentation with a magnetic unipolarity filter. The method starts from co-temporal SDO/AIA 171, 193, and 211 Å images (and an SDO/HMI LOS magnetogram) and exploits the fact that CH pixels cluster in intensity–intensity space because of their characteristic temperature/density (relatively brighter in 171 Å and dim in 193/211 Å). For each pair of passbands (193 vs. 171, 211 vs. 171, 211 vs. 193), CHIMERA builds 2D histograms of on-disk pixel intensities, identifies the valley separating the quiet-Sun cluster from the low-emission (CH-like) cluster, and fits an "optimal" separation in log-space (a line in log-space that becomes a curve in intensity space), with a normalization step that adapts the curve to the day's mean on-disk intensities. Pixels lying below each of the three segmentation curves are flagged as CH candidates, yielding three candidate maps; a logical conjunction (AND) of these maps removes features that do not persist across all three thermal constraints. The resulting binary map is then cleaned by rejecting small regions (e.g., <1,000 arcsec^2) and by using HMI to enforce the unipolar nature of CHs: candidates whose mean radial field averages to 0 are discarded, with an area-dependent threshold (more stringent for small regions, looser for large ones). The remaining regions define the final CHIMERA coronal-hole mask.

2.3.3. Region Growth

The Region Growth coronal-hole detection algorithm (Tlatov et al., 2014) identifies coronal holes by iteratively expanding seed regions of low EUV emission while enforcing physical and morphological constraints. The method is applied to full-disk SDO/AIA EUV images (typically 193 Å or 211 Å), where candidate seed pixels are first selected using an intensity threshold relative to the quiet Sun, ensuring that only persistently dark regions are initialized. From these seeds, regions are grown by adding neighboring pixels whose intensities remain below an adaptive threshold and whose local properties (e.g., gradient and contrast) are consistent with coronal-hole boundaries, allowing the region to expand until a stopping criterion is met. To suppress contamination from filaments and transient dark features, the grown regions are cross-checked against line-of-sight magnetic field maps from SDO/HMI, retaining only regions that exhibit a dominant magnetic polarity, a defining property of coronal holes. Finally, morphological filtering and minimum-area constraints are applied to remove small or fragmented structures, yielding a cleaned binary coronal-hole mask suitable for long-term synoptic analysis and solar-cycle studies.

2.3.4. Coronal Structure Annotation Data Set

Thanks to these sources, together with the custom masks provided by Mackovjak et al. (2021), a large and diverse set of ground-truth annotations is available, enabling the training of supervised machine-learning models. While SPoCA and the aforementioned algorithms are widely used as reliable sources of coronal-hole and active-region annotations and therefore serve as ground truth in this study, their direct deployment in an on-board context is currently impractical. In particular, SPoCA relies on a multi-stage processing pipeline that includes clustering in high-dimensional feature space, iterative region labeling, and the use of calibrated Level-2 EUV images, which together entail non-trivial computational cost, memory requirements, and algorithmic complexity. Moreover, SPoCA is designed for ground-based processing with access to full-disk, well-calibrated data and does not explicitly target real-time or resource-constrained environments. Similar limitations apply to other physics-based methods such as CHIMERA and Region Growth. Consequently, these algorithms are used here solely as reference annotation sources, while the focus of this work is on evaluating lightweight learning-based alternatives that are more suitable for potential future on-board deployment on space-qualified, low-power hardware such as FPGAs or embedded processors.

2.4. Segmentation Using SCSS-Net

The SCSS-Net (Solar Coronal Structure Segmentation Network; Mackovjak et al., 2021) is a U-Net-like (Ronneberger et al., 2015) deep learning model developed specifically to segment solar coronal structures, mainly active regions (ARs), coronal holes (CHs), and quiet Sun (QS), from extreme ultraviolet (EUV) images of the Sun, particularly those captured by the SDO/AIA instrument. It consists of an encoder-decoder structure. The encoder captures context through downsampling (via convolutions and pooling), and the decoder performs upsampling to recover spatial resolution for segmentation masks. Moreover, it includes skip connections between corresponding encoder and decoder layers, which help preserve fine-grained spatial details.

The encoder part of the network consists of three contracting blocks. Each block includes two sequential 2D convolutional layers with a kernel size of 3 $\times$ 3, padding set to "same," and a stride of 1. These are followed by batch normalization and ReLU activations, enhancing stability and non-linearity. Max-pooling layers with a 2$\times$ 2 window are applied after each block to downsample the feature maps, allowing the network to learn hierarchical representations of solar structures at multiple scales. The decoder mirrors the encoder and is composed of three expanding blocks. Each block begins with a transpose convolution (also called deconvolution) to upsample the feature maps, followed by concatenation with the corresponding feature maps from the encoder via skip connections. This helps retain spatial detail that might otherwise be lost during downsampling. The upsampled features are processed through two 3 $\times$ 3 convolutional layers, each followed by batch normalization and ReLU activations. A final 1 $\times$ 1 convolutional layer reduces the output to three channels corresponding to the target classes, and a softmax activation is applied to produce a per-pixel class probability map (Figure 3).

SCSS-Net is optimized using Adam with a learning rate of 10^{-4}. The objective function that needs to be minimized during training is commonly referred to as the loss function (Goodfellow et al., 2016). In our case, which involves binary classification at the pixel level, determining whether a pixel belongs to the target class or not, we employ binary cross-entropy (BCE). It is defined as:

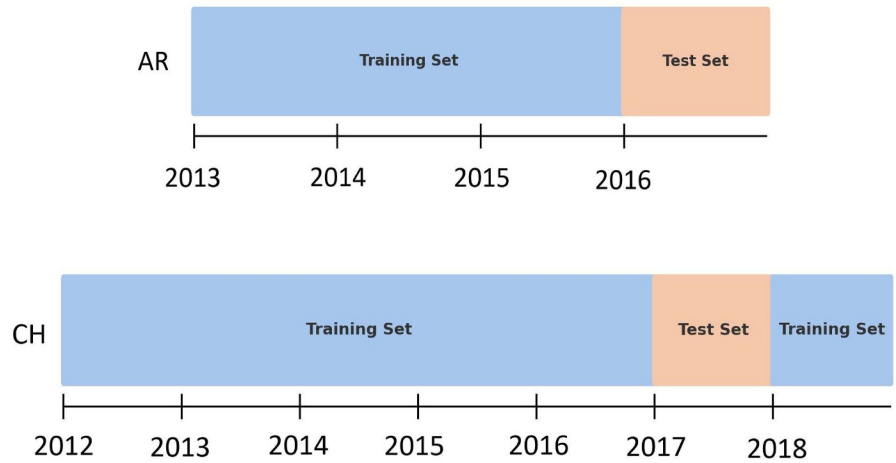

Figure 2. Data split for SCSS-Net and YOLOv8n for the active region (AR) and coronal hole (CH) as proposed by Mackovjak et al. (2021).

$$\mathrm{BCE} = -(y \log(\hat{y}) + (1 - y)\log(1 - \hat{y})), \tag{1}$$

where $y \in \{0, 1\}$ denotes the ground truth label, and $\hat{y} \in [0, 1]$ represents the predicted probability for the positive class.

The network contains approximately 7 million trainable parameters, which makes it sufficiently lightweight for efficient training and inference without compromising accuracy.

Each input is resized to a 256×256 pixel image. To enhance generalization and robustness, various data augmentation techniques are employed, including horizontal and vertical flips, random rotations, brightness and contrast adjustments, and random cropping. Following the proposed augmentation protocol, each transformation is applied with a probability of 50%. The applied augmentations include: horizontal flipping (reflection about the y-axis), vertical flipping (reflection about the x-axis), random rotation by an angle uniformly sampled from the range $(-45°, 45°)$, random gamma correction with a gamma range of $(100, 150)$, and random brightness and contrast adjustments with brightness and contrast ranges of $(0, 0.2)$ and $(0, 0.4)$, respectively. Training is performed with a batch size of 16 images over 50 to 100 epochs, using a 10%–20% validation split. Training and validation loss were carefully monitored to avoid overfitting. Based on this analysis, 100 epochs were selected for the final models.

We train two distinct segmentation models, one dedicated to active regions (ARs) and the other to coronal holes (CHs), following the approach of Mackovjak et al. (2021). For each model, we adopt their proposed data split (see Figure 2), as it ensures a fair partitioning without introducing biases while also enabling direct comparison and benchmarking. Adopting a different data split would have made a direct comparison with the original SCSS-Net work less appropriate.

We note that the employed data split, together with the use of the SDOML data set (Galvez et al., 2019), does not facilitate assessing model generalizability across different solar cycles and is therefore beyond the scope of this work. Furthermore, instrument degradation has already been accounted for during the construction of the SDOML data set, and our models do not explicitly address this aspect either.

2.5. Object Detection Using YOLO

YOLO (You Only Look Once; Jiang et al., 2022; Jocher & Qiu, 2024; Redmon & Farhadi, 2018) is a real-time object detection framework that reframes object detection as a single regression problem, directly predicting bounding boxes and class probabilities from full images in one evaluation. YOLO offers several advantages over its counterparts, making it useful to a wide range of applications. It can provide real-time speed (often exceeding 30 FPS on a good GPU), and it is also possible to deploy it on a general-use CPU or FPGA (Ni et al., 2023; Shao et al., 2024). Its lightweight versions (e.g., YOLOv5n, YOLOv8n) are exceptionally compact.

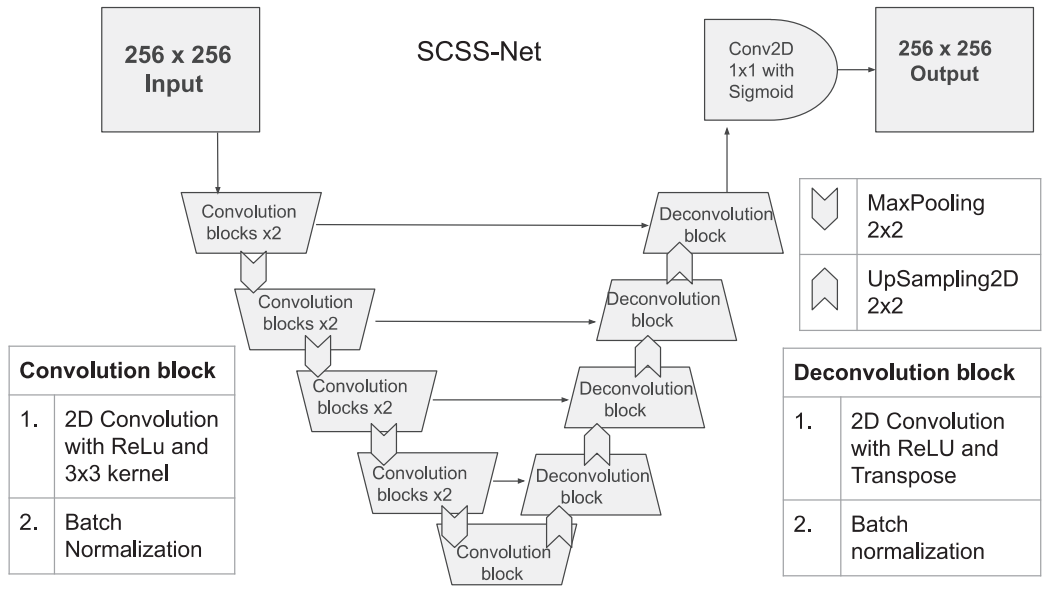

Figure 3. The architecture of SCSS-Net.

Since its first appearance, various versions of YOLO have been presented. Jiang et al. (2022) reviewed many of them, and one can choose the appropriate edition. In our case, a very light yet powerful model is needed. YOLOv8n was selected as the most suitable model, comprising 3.2 million trainable parameters and achieving an mAP (50–95) of 37.3, making it the lightest model proposed at the time this work was conducted. This model consists of three parts: the backbone, the neck, and the head (Figure 4).

The Sigmoid Linear Unit (SiLU) activation function is preferred, as it improves gradient flow and convergence, offering better efficiency and performance compared to ReLU. The C2f module (short for Cross-Stage Partial with 2 fused convolutions) is a widely used custom-designed building block that consists of a 1×1 convolution followed by a 3×3 convolution with batch normalization and SiLU. The Spatial Pyramid Pooling - Fast (SPFF) performs a sequence of max-pooling operations with the same kernel size (typically 5×5) all to the same input. It concatenates the original input with the pooled outputs, enhancing the receptive field without significantly increasing computational cost.

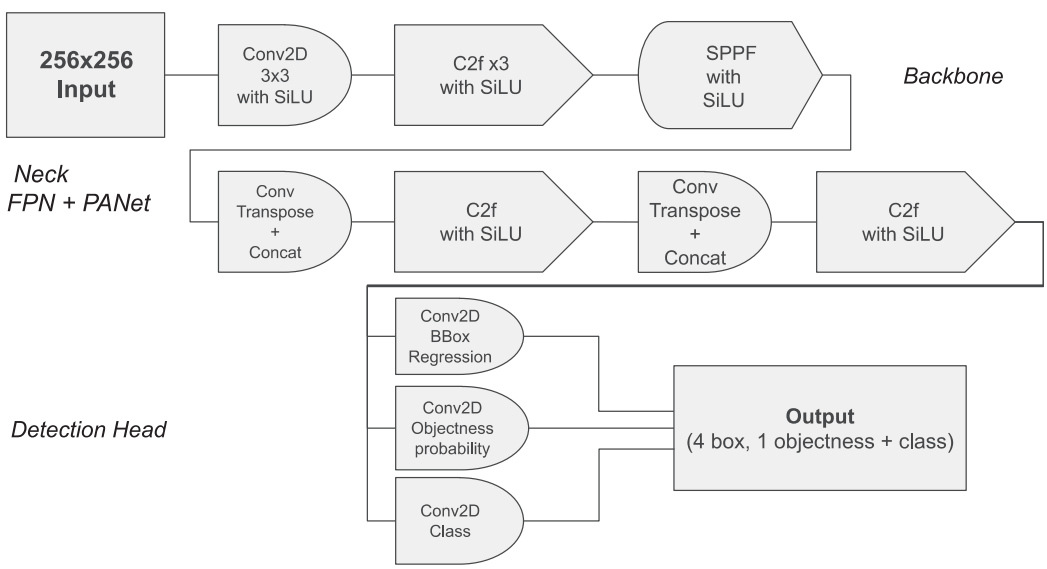

Figure 4. The architecture of YOLOv8n.

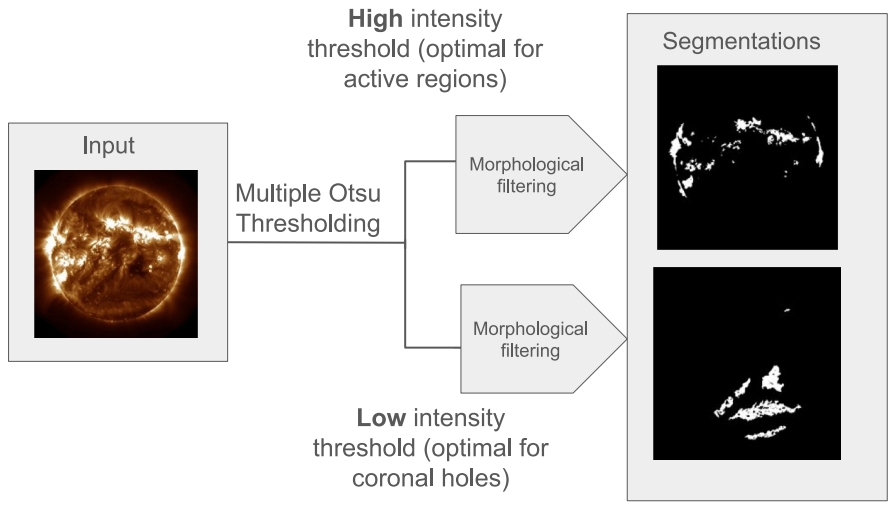

Figure 5. The architecture of the BCVO framework.

The SDOmlv2 data set mentioned above is appropriately adapted for object detection by converting the segmentation masks into axis-aligned bounding boxes. Following the same chronological data split used for the SCSS-Net models (see Figure 2), we train two separate detection models: one dedicated to active regions and another to coronal holes.

Similar to SCSS-Net, after exploring various input sizes (512×512, 256×256 and 126×126), an input size of 256×256 was selected as the best compromise between network complexity and performance. Finally, both AR and CH models were trained for 100 epochs, which was sufficient to achieve convergence without overfitting.

2.6. Segmentation Using BCVO

In this work, we propose a lighter computational-wise framework consisting of a set of basic computer vision operations and traditional machine learning (BCVO). Its main pipeline is visualized in Figure 5.

Initially, we produce a binary mask from a grayscale SDO AIA image using simple automatic thresholding operations widely used in computer vision. Thresholding converts grayscale images into binary masks, effectively separating objects of interest from the background. Otsu's method (Otsu, 1975) is one of the most popular techniques for automatic threshold selection. This algorithm assumes the image has a bimodal histogram. It calculates the optimal global threshold by maximizing the between-class variance, which is a statistical measure of how well the foreground and background are separated. Multiple Otsu (Arora et al., 2008; Liao et al., 2001), an extension of the original method, evaluates all possible combinations of thresholds and selects the set that best separates the image into distinct classes by maximizing the between-class variance based on pixel intensity distributions. Otsu's method is fully automatic and computationally efficient, making it especially suitable for consistent lighting conditions. To address scenarios involving more than two distinct intensity regions, such as in solar coronal images, multiple Otsu thresholding extends the classical method by selecting multiple thresholds that divide the image into three or more classes. This is particularly useful in the context of solar coronal structure segmentation, where images often exhibit complex intensity distributions that represent different physical phenomena, such as active regions, coronal holes, and quiet Sun areas.

Subsequently, we use morphological filtering to post-process the resulting binary masks. Thanks to morphological filtering, one can refine binary segmentation masks by exploiting the spatial structure of objects. Such operations can clean noise, close gaps, and smooth boundaries. Basic operations include erosion, which shrinks foreground regions to eliminate small artifacts, and dilation, which expands regions to fill gaps. Their combinations, such as opening (erosion followed by dilation) and closing (dilation followed by erosion), are particularly effective for removing isolated noise and sealing small holes. In this work, after exploring various combinations of such operations, we chose to perform opening and then closing, which removes small segmentations representing noise and connects large segmented areas that represent the same coronal structure.

Table 1
Performance Comparison of SCSS-Net, YOLO, and BCVO for Active Region (AR) and Coronal Hole (CH) Segmentation and Detection for Level-2 SDO Test Samples

	Dice		IoU		Precision		Recall		mAP@0.5	
Method	AR	CH	AR	CH	AR	CH	AR	CH	AR	CH
SCSS-Net	0.61 ± 0.08	0.86 ± 0.08	0.51 ± 0.05	0.78 ± 0.07	0.76 ± 0.06	0.79 ± 0.10	0.64 ± 0.08	0.91 ± 0.09	0.77 ± 0.1	0.92 ± 0.1
YOLO	0.57 ± 0.03	0.80 ± 0.05	0.4 ± 0.03	0.69 ± 0.05	0.71 ± 0.11	0.75 ± 0.09	0.59 ± 0.11	0.85 ± 0.08	0.68 ± 0.08	0.86 ± 0.07
BCVO	0.58 ± 0.03	0.78 ± 0.05	0.37 ± 0.04	0.6 ± 0.05	0.69 ± 0.12	0.72 ± 0.10	0.56 ± 0.09	0.82 ± 0.09	0.64 ± 0.09	0.81 ± 0.08

2.6.1. Characterization of Segments

Following SCSS-Net or BCVO, we design a simple unsupervised machine learning algorithms further characterize the detected structures. This is achieved by extracting hand-crafted features, which are manually designed descriptors that quantify aspects such as shape, area, intensity distribution, texture, edge sharpness, or spatial context. These features form a compact representation of each region, enabling further analysis beyond simple pixel-level segmentation. We initially defined a set of 27 features (Table A1) characterizing texture, shape, and contour, partially inspired by the feature set proposed by Delouille et al. (2018). After carefully examining their mutual correlations through the covariance matrix, this set was reduced to 21 features.

To group and interpret the segmented regions in a meaningful way, unsupervised learning techniques can be applied. K-means clustering (Lloyd, 1982; MacQueen, 1967) partitions the feature space into clusters, where each instance is assigned to the cluster with the nearest centroid based on a distance metric, typically the Euclidean distance. The centroids get iteratively updated to minimize the within-cluster variance. A common challenge in clustering is choosing the appropriate number of clusters. To address this, the Kneedle method (Satopaa et al., 2011) is used to automatically detect the "elbow point" in a plot of within-cluster variance (or inertia) versus the number of clusters. This point represents the optimal trade-off between model complexity and clustering performance, where adding more clusters yields diminishing returns.

In parallel, the feature space is effectively visualized by projecting it into a 2D or 3D space using t-distributed Stochastic Neighbor Embedding (t-SNE). Principal Component Analysis (PCA) is typically applied internally to reduce noise and dimensionality. Subsequently, K-means clustering is performed on the resulting representation to identify groups within the data (see Figure 6).

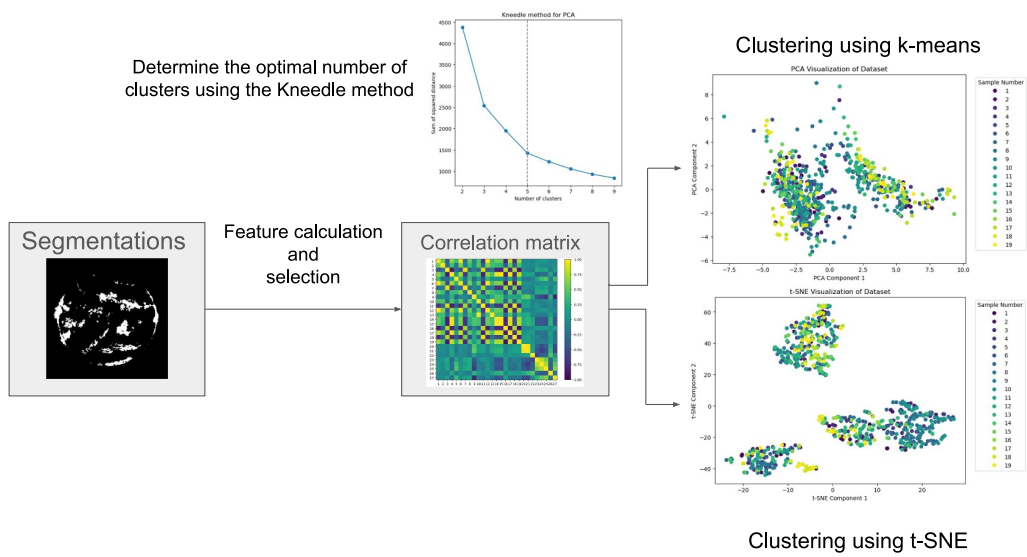

Figure 6. Characterization of the segmentations.

2.7. Evaluation Metrics

In image segmentation, the objective is to classify each pixel into a specific class or region, whereas in object detection the task involves identifying and localizing object instances using bounding boxes. Common evaluation metrics across these tasks include precision, recall, F1-score, pixel accuracy, and mean Average Precision (mAP). Metrics such as Intersection over Union (IoU) and the Dice coefficient are particularly informative when the goal is to evaluate the spatial overlap between predicted and ground-truth structures.

In this study, we therefore adopt IoU as the primary metric to evaluate the spatial agreement between predictions and reference annotations. To confirm the observed performance trends, we additionally report the Dice score. Subsequently, we further include precision, recall, and mean Average Precision at an IoU threshold of 0.5 (mAP@0.5) being in line with standard practices in object detection and segmentation literature. These additional metrics provide complementary insights by quantifying potential over-detection (false positives), under-detection (missed structures), and overall detection quality.

IoU, also called the Jaccard Index, is a standard metric used to quantify the overlap between two regions: the predicted segmentation and the ground truth. It is defined as the area of the intersection divided by the area of the union of the two regions:

$$\text{IoU} = \frac{|A \cap B|}{|A \cup B|}, \tag{2}$$

where A is the predicted set of pixels (or bounding box) and B is the ground truth. The value of IoU ranges from 0 to 1, with 1 indicating perfect overlap. IoU is widely used in both semantic segmentation and object detection to evaluate the spatial precision of predictions. In detection tasks, a prediction is often considered correct if its IoU with the ground truth exceeds a predefined threshold (e.g., 0.5).

The Dice score, also known as the Sørensen–Dice coefficient, is a widely used metric for evaluating image segmentation performance by measuring the overlap between the predicted segmentation and the ground-truth mask. It is particularly well suited for imbalanced problems, as it directly quantifies the agreement between foreground regions. The Dice score ranges from 0 to 1, where 1 indicates perfect overlap and 0 denotes no overlap.

$$\text{Dice} = \frac{2\,|A \cap B|}{|A| + |B|} \tag{3}$$

where A and B denote the predicted and ground-truth pixel sets, respectively.

Precision and recall further quantify the reliability of detections. Precision measures the proportion of predicted positive detections that are correct, whereas recall measures the proportion of ground-truth objects that are successfully detected:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{4}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{5}$$

where TP, FP, and FN denote true positives, false positives, and false negatives, respectively. In our study, recall indicates the ability of a method to detect all existing structures, while precision reflects the tendency of the model to avoid spurious detections.

Finally, we report the mean Average Precision at an IoU threshold of 0.5 (mAP@0.5), a widely used metric in object detection benchmarks. For each class, the precision–recall curve is constructed by varying the detection confidence threshold, and the Average Precision (AP) is computed as the area under this curve. The mean of these values across classes yields the mAP score. The mAP@0.5 metric considers a prediction correct when its IoU

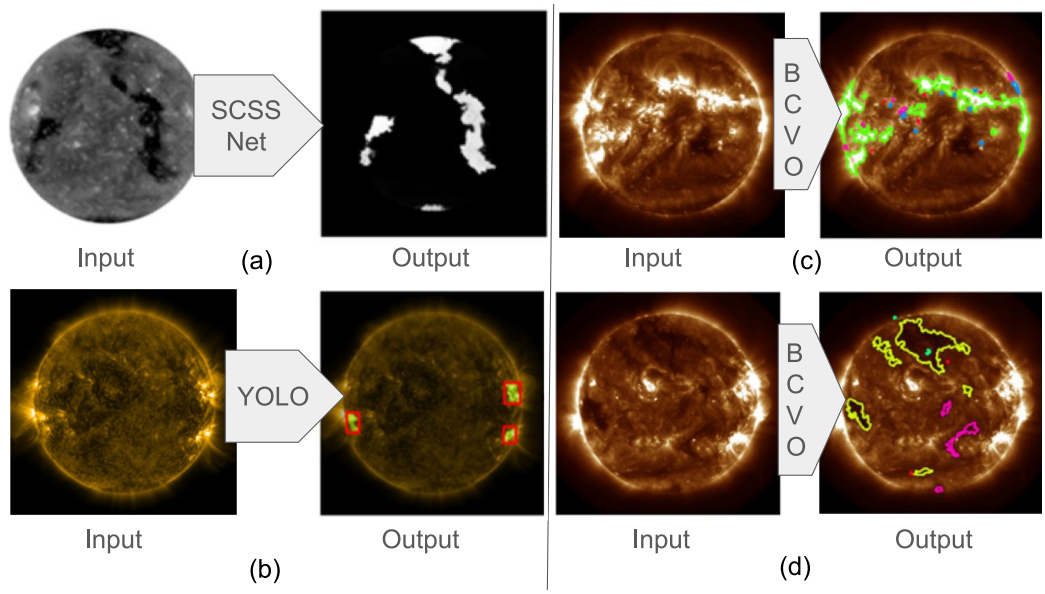

Figure 7. Illustrative examples of outputs produced by the three frameworks when applied to Level 2 SDO AIA data: (a) CH segmentation using SCSS-Net, (b) AR detection using YOLOv8n, (c) active region and bright spot segmentation using BCVO and segmentation characterization, and (d) coronal hole segmentation using BCVO and segmentation characterization.

with the ground truth exceeds 0.5, thereby providing a thresholded measure of detection quality that complements the spatial overlap metrics described above.

The authors acknowledge the use of generative AI tools to assist with refining grammar and vocabulary. All AIgenerated content was critically reviewed to ensure accuracy and appropriateness.

3. Results

This section provides both a qualitative and quantitative evaluation of the methods under investigation. For all trainable methods, namely SCSS-Net and YOLOv8n, we train two separate models: one dedicated to active regions (ARs) and another to coronal holes (CHs). The AR models use SDO/AIA images at 171 Å as input, whereas the CH models are trained on SDO/AIA images at 193 Å.

Figure 2 illustrates the chronological split into data training and test sets per year for the AR model and CH model, respectively. We adopt the data split proposed by Mackovjak et al. (2021), which enables a more reliable and direct comparison with their results. Considerations related to solar-cycle dependence and instrument degradation are beyond the scope of this study and are therefore not addressed in the chosen split.

When hyperparameter tuning is necessary, a subset of the training set is reserved for validation purposes and is later merged with the training set. For the BCVO method, we apply multiple Otsu thresholding, which can automatically determine a threshold for high-intensity values associated with active regions and low-intensity values corresponding to CHs and other structures. In parallel, the following characterization theme follows the same data split for consistency.

In Figure 7 we illustrate some representative outputs of the three selected frameworks when applied to level 2 SDO AIA samples.

3.1. Evaluating on Level-2 SDO

All metrics described in Section 2.7 are computed to provide a comprehensive and multi-perspective evaluation of the examined models. The corresponding results are summarized in Table 1.

Furthermore, to facilitate the interpretation of the results, we use the IoU metric for visual representation of the performance of the model. In Figure 8 (top), we plot the segmentation and detection performance, measured by

10.1029/2025JH001023

Table 2
Performance Comparison of SCSS-Net, YOLO, and BCVO for Active Region (AR) and Coronal Hole (CH) Detection for Level-0 SDO Test Samples

	Dice		IoU		Precision		Recall		mAP@0.5	
Method	AR	CH	AR	CH	AR	CH	AR	CH	AR	CH
SCSS-Net	0.61 ± 0.08	0.86 ± 0.08	0.42 ± 0.03	0.64 ± 0.02	0.76 ± 0.06	0.79 ± 0.10	0.64 ± 0.1	0.91 ± 0.07	0.77 ± 0.1	0.92 ± 0.1
YOLO	0.57 ± 0.03	0.80 ± 0.05	0.4 ± 0.02	0.69 ± 0.03	0.71 ± 0.11	0.76 ± 0.06	0.58 ± 0.08	0.84 ± 0.08	0.69 ± 0.05	0.86 ± 0.04
BCVO	0.58 ± 0.03	0.78 ± 0.05	0.36 ± 0.04	0.57 ± 0.02	0.68 ± 0.12	0.71 ± 0.09	0.55 ± 0.11	0.80 ± 0.08	0.63 ± 0.05	0.79 ± 0.04

IoU, for the three methods (SCSS-Net, YOLO, and BCVO) applied to ARs and CHs using the corresponding trained models. Across both categories, SCSS-Net consistently outperforms the other methods, achieving the highest IoU values. YOLO achieves moderate performance, while BCVO produces the lowest IoU scores of the three, yet still relatively high.

The error bars, representing the standard deviation (ranging from ±0.03 to ±0.07), indicate the variability in the performance of each method. Although some overlap is observed (e.g., between YOLO and BCVO for AR detection), the overall differences remain substantial. This observation is supported by statistical analysis. More specifically, a Friedman test was conducted to compare the three methods across all evaluated metrics. In this way, we check whether the three methods differ significantly. Friedman test is appropriate in this case since we are comparing multiple algorithms across multiple metrics and data sets without assuming normality. The test revealed statistically significant differences between the models for both AR and CH models ($p < 0.05$).

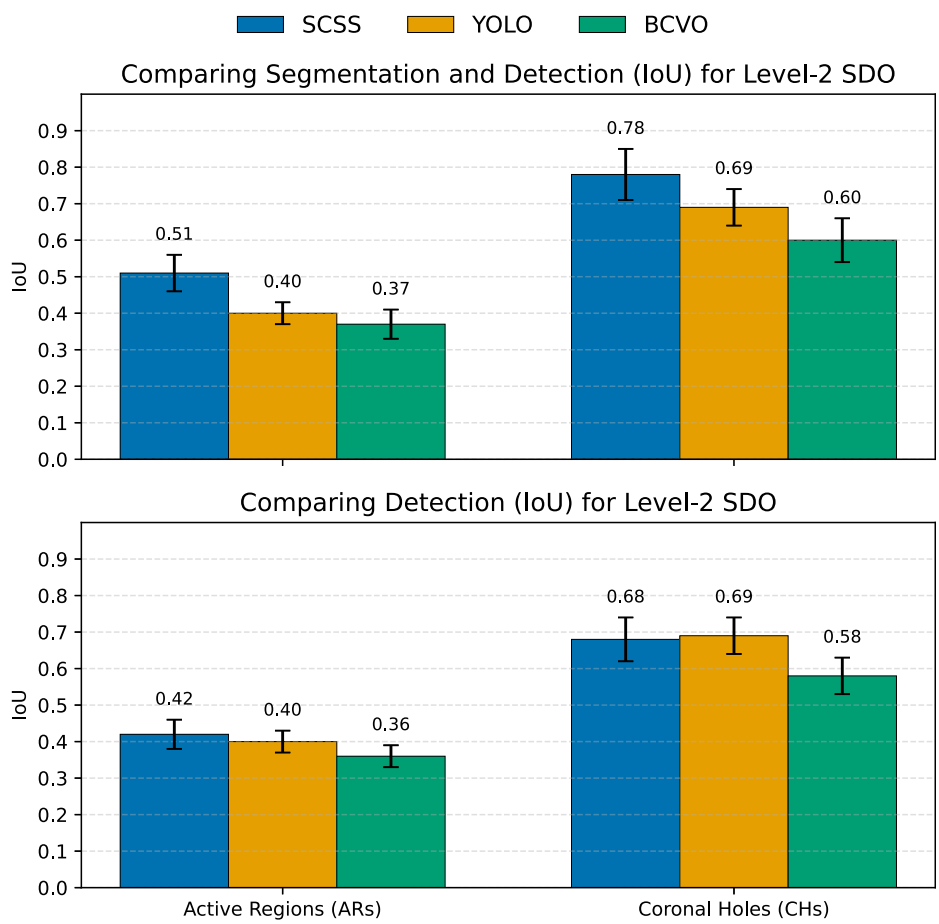

Figure 8. Comparing the three methods for AR and CH segmentation/detection.

Table 3
Comparing Key Characteristics of SCSS-Net, Yolov8n, and Basic Computer Vision Operations (BCVO)

SCSS-Net	Yolov8n	BCVO
Segmentation	Detection	Segmentation
U-Net	CNN	Automatic thresholding using computer vision and morphological filters
78M trainable parameters	3.2M trainable parameters	No training needed
Inference: GPU (CPU possible but slow)	Inference: CPU/GPU	Inference: CPU
		Not optimal for FPGA integration (these computations cannot be easily parallelized)

A comparable performance trend is confirmed by the Dice score across the three frameworks for both active-region (AR) and coronal-hole (CH) segmentation. Specifically, the AR models achieve Dice scores of 0.61 for SCSS-Net, 0.57 for YOLO, and 0.58 for BCVO, while the CH models obtain scores of 0.86, 0.81, and 0.78, respectively.

Since SCSS-Net and BCVO perform segmentation, whereas YOLO detection one can convert and adapt segmentation masks to bounding boxes for both ground truth and model output. This way, all the studied frameworks are evaluated more fairly for detection and YOLO's predictions can be assessed without directly comparing bounding boxes to pixel-wise segmentation maps. Figure 8 (below) represents IoU for all three frameworks performing detection. In this second performance comparison, we observe that SCSS-Net performs similarly to YOLO. A similar pattern is also reflected in the Dice score. For active-region (AR) segmentation, the Dice scores of SCSS-Net and BCVO decrease to 0.58, comparable to the score achieved by YOLO (0.57). For coronal-hole (CH) segmentation, the Dice scores of SCSS-Net and BCVO decrease to 0.83 and 0.77, respectively.

3.2. Evaluating on Level-0 SDO

We evaluate our models trained on Level 2 data both qualitatively and quantitatively on Level 0 data sets. The full set of computed metrics is summarized in Table 2 while Figure 9 presents the IoU scores computed across all Level-0 samples. Compared to the evaluation on Level-2 data, the AR models maintain their performance, while the CH models appear slightly more sensitive to the lower data quality. Though they still achieve relatively high IoU values. For CHs, the IoU scores remain in the range of 0.6–0.7, with YOLOv8n demonstrating the highest accuracy. SCSS-Net is again evaluated under stricter conditions, as its performance is assessed within a detection framework, despite being inherently a segmentation-based model.

We find that applying median correction followed by per-quadrant pixel value normalization effectively eliminates artifacts in Level-0 SDO data, including the prominent quadrant discontinuities (Figure 1). In contrast,

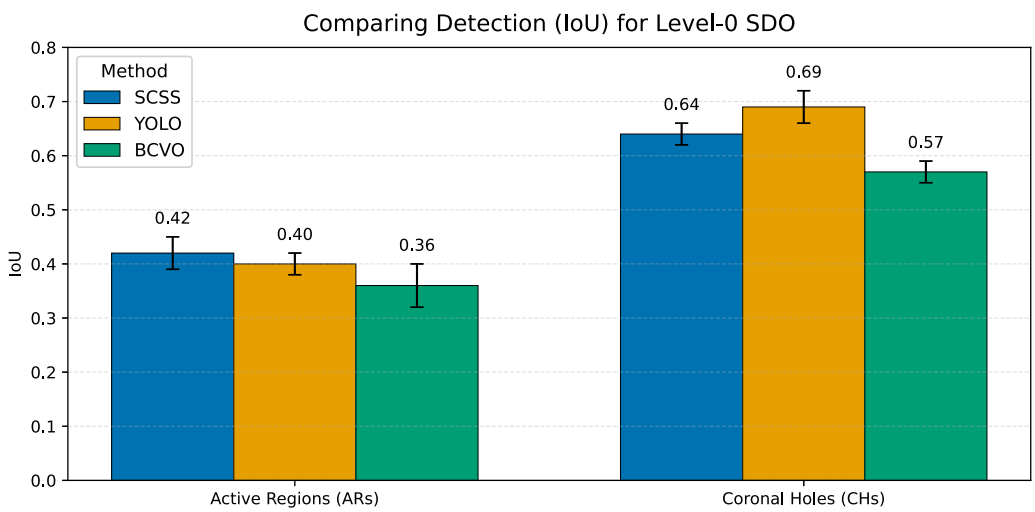

Figure 9. Comparing the three methods performing detection on level 0 SDO AIA samples for ARs and CHs.

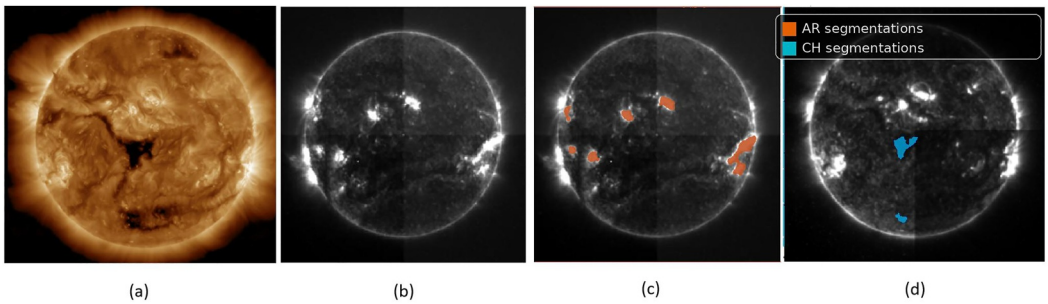

Figure 10. Evaluation of SCSS-Net on Level-0 SDO/AIA data after simpler than optimal preprocessing. (a) Reference Level-2 SDO/AIA 193 Å sample (11/12/2023). (b) Poorly preprocessed Level-0 SDO/AIA 131 Å sample (15/12/2023), rotated by 180°, where quadrant discontinuities are clearly visible. (c) Same image as in panel (b) with the predicted Active Region (AR) segmentation overlay. (d) Coronal Hole (CH) prediction overlay for the Level-0 sample (11/12/2023). The results suggest that the AR SCSS-Net model is more tolerant to minimal preprocessing and visible quadrant discontinuities compared with the CH SCSS-Net model.

lighter preprocessing steps result in artifacts, as illustrated in Figure 10, where we show examples of semi-preprocessed, Level-0 SDO samples passed to SCSS-Net. As previously discussed, correcting quadrant discontinuities is the most critical step. Despite this, the SCSS-Net model for ARs remains relatively robust and performs well even with partial correction. In contrast, the CH model is significantly more sensitive to these artifacts and fails to generalize effectively unless the discontinuities are fully addressed.

In Figure 10, we show that active regions and coronal holes are effectively segmented by the SCSS-Net models. The AR model appears more robust, while the performance of the CH model is compromised, likely due to inaccurate background noise correction. As shown in the figure, IoU for AR remains nearly identical to that achieved on the Level 2 test set, whereas the CH model's IoU score decreases by approximately 12%.

4. Discussion

In this work, we investigate a representative use case in which artificial intelligence could be incorporated onboard a space mission, focusing on the detection of solar coronal structures, with particular emphasis on the accurate localisation of active regions and coronal holes. Such capabilities could enable missions to autonomously perform tasks such as data-sample prioritization for downlink, automatic instrument triggering and pointing, or the activation of adaptive observing modes, thereby reducing the need for continuous ground-based intervention. At the same time, when required, the system could request and support human decision-making by flagging events of interest and providing contextual information.

We evaluate three methods for solar structure segmentation or detection, and investigate their suitability for onboard integration in future spacecraft. We assess each method in terms of its performance, that is, its ability to meet scientific requirements, and efficiency, that is, its compliance with technical and engineering constraints.

Segmentation provides pixel-level delineation of structures, offering detailed spatial information about the shape and boundaries of regions of interest. In contrast, detection focuses on localizing objects by generating bounding boxes, without requiring fine-grained detail. While segmentation is generally more accurate and informative and can extract hand-crafted features, it is also considerably more computationally demanding, as demonstrated in the case of SCSS-Net. For onboard processing in space missions, where computational and power resources are limited, detection presents a more practical alternative. In contrast, detection enables efficient identification and prioritization of solar regions of interest, mainly active regions and coronal holes, without the overhead associated with full-resolution segmentation. In an onboard setting, this can facilitate optimized data downlink, ensuring that the most scientifically relevant observations are transmitted to Earth. Furthermore, detection outputs can be used to guide high-resolution instruments for targeted follow-up observations, effectively balancing onboard efficiency with scientific value. This is particularly relevant for missions like Solar Orbiter (Müller et al., 2020), which hosts both low- and high-resolution telescopes. In this context, the high-resolution instrument must be selectively pointed toward regions of scientific interest.

When comparing the performance of the studied frameworks, it is important to note that while the IoU metric is a widely accepted metric for evaluating segmentation accuracy, its application in comparing segmentation and detection tasks should be interpreted with caution. In segmentation, IoU measures the pixel-wise overlap between the predicted mask and the ground truth, providing a fine-grained evaluation of spatial accuracy. In contrast, detection methods often produce coarse bounding boxes or sparse region proposals that approximate the object location without capturing its precise shape or boundary. As a result, applying IoU uniformly across both types of tasks may favor segmentation methods by design, since they inherently optimize for pixel-level overlap. This can lead to an underestimation of detection performance, especially in scenarios where detection outputs are not intended to delineate exact boundaries but rather to localize regions of interest. Therefore, while IoU can provide a useful comparison baseline, it is important to interpret the results within the context of each task's objective and to complement IoU with other task-specific metrics where appropriate.

This is why, at a later stage, we reformulate the SCSS-Net approach as a detection problem by converting both the ground truth and the predicted segmentation masks into bounding boxes. For this conversion, we define axis-aligned bounding boxes, which enclose the segmented regions within the smallest upright rectangles aligned to the image axes. While this may include background pixels around rotated or irregularly shaped structures, axis-aligned boxes are the standard format required by object detection frameworks such as YOLO. In contrast, rotated bounding boxes can provide a tighter fit by more closely following the segmentation's orientation, but can generally be incompatible with standard detection pipelines. In our work, we adopt axis-aligned bounding boxes for both adapting SCSS-Net to a detection-based formulation and for preparing the training and testing data sets used with YOLO.

Mackovjak et al. (2021) report IoU scores of 0.78 for active regions (evaluated using 2016 data) and 0.57 for coronal holes (evaluated using 2017 data), utilizing ground-truth masks derived from SPoCA. Our results are in agreement with these findings. Furthermore, based on qualitative inspection, we observe that SPoCA masks often underestimate the true spatial extent of coronal hole regions, particularly for coronal holes, a phenomenon often reported in the literature (Linker et al., 2021).

Since this work does not solely focus on comparing the three methods based on their segmentation performance, but also evaluates their potential for onboard deployment in future spacecraft missions, additional characteristics are considered. Table 3 summarizes the key implementation details of each method. It is evident that SCSS-Net is too resource-intensive to be suitable for such constrained environments. In contrast, selecting the nano version of YOLO offers a substantial reduction in model size, up to 95% fewer parameters, making it far more suitable for embedded applications. Although BCVO represents a lightweight framework, its reliance on traditional computer vision and machine learning techniques poses limitations. Such approaches are generally not easily parallelizable and, therefore, cannot exploit the energy and computational efficiencies typically required for deployment on specialized hardware beyond a standard general-purpose CPU.

Although the YOLO nano variant is considered lightweight, it may still be considered too demanding for systems with strict hardware and power constraints, as highlighted in the literature (Antunes & Podobas, 2025). A model exceeding 1 million parameters can be characterized as heavy when one aims to embed such a method in a low-energy consumption setting. Furthermore, complicated activation functions like SiLu do not allow easy translation to a hardware language. To address these, future work will explore further simplification of the YOLO architecture or the development of a custom, lightweight CNN designed from scratch, with a target size of approximately 1 million parameters. A key question remains whether such a simplified model can maintain comparable performance, and more importantly, whether its accuracy is sufficient to meet the scientific objectives of detecting solar coronal structure. Jocher and Qiu (2024) recently released a new version of YOLO, incorporating a nano version consisting of 2.6 million parameters, and will definitely be evaluated in the short-term future. However, a promising fact is that using YOLO, one can proceed with reducing input image resolution and feature map bit quantization, leading to very limited power consumption, even below 5 W (Wang et al., 2020).

An important aspect of assessing onboard implementation is evaluating the performance of the studied methods on raw, uncalibrated data, both quantitatively and qualitatively. When comparing Level-0 and Level-2 SDO data, we observe no significant noise or artifacts; the most noticeable is the visible quadrant tiling. This effect occurs because each quadrant corresponds to a separate portion of the CCD and is often compressed and transmitted independently, resulting in slight discontinuities. These are corrected during the Level-0 to Level-1 calibration process. In future missions, this issue is expected to diminish, as CMOS sensors will likely be favored since they can offer lower

power consumption, higher frame rates, and on-chip processing, as demonstrated by the CMOS-APS sensors aboard the Solar Orbiter (Howard et al., 2013). In our study, we address this issue by introducing a dedicated preprocessing step consisting of median correction followed by percentile-based normalization of pixel intensities within each quadrant. This approach scales the pixel values to the range [0, 1], making them suitable for machine learning methods while effectively removing any visual artifacts such as quadrant discontinuities. Lighter preprocessing fails to fully correct these artifacts, which particularly impacts the performance of coronal hole models that are more sensitive to such inconsistencies. Nevertheless, we demonstrate that machine learning models trained on level-2 SDO data remain compatible with Level-0 inputs, still offering scientifically satisfactory results.

Within the scope of this study, we intentionally include artifacts related to data acquisition and instrumental effects in order to assess model robustness under degraded and less controlled input conditions. However, handling additional anomalies such as invalid or corrupted frames (e.g., missing data, telemetry dropouts, or Quality $\neq 0$ flags) through explicit quality-control logic is beyond the scope of the present work and would be required in an operational setting.

One possible strategy to address the impact of such anomalies in practice is to enforce temporal consistency in the input data, for example, by using differences or short sequences of consecutive acquisitions, ensuring that decisions are not made based on a single potentially invalid frame. Alternative approaches could include incorporating quality flags when available or introducing lightweight anomaly-detection or sanity-check mechanisms upstream of the inference pipeline. While these safeguards are not implemented here, they represent important extensions for future operational deployment and are acknowledged accordingly. More generally, when decisions rely on the output of AI models, it is advisable to base them on a sequence of observations rather than on a single sample, which may be corrupted and lead to misleading conclusions. Using temporal or multi-sample information enables the system to better tolerate isolated anomalies and transient artifacts.

Lastly, yet of significant importance, expanding training data toward a more universal model and quantization trials for greater power efficiency represent key directions for future work.

4.1. Assessment of FPGA Suitability

FPGAs offer a compelling platform for accelerating computational algorithms due to their reconfigurable architecture, enabling significant reductions in power consumption and inference time (Antunes & Podobas, 2025). However, carefully considering algorithmic characteristics is crucial, as only some of the methods discussed in this paper are well-suited for efficient FPGA implementation.

U-Net-like algorithms have previously been successfully implemented on FPGAs (Bahl et al., 2019; Kim et al., 2024). Nevertheless, a network with 78 million parameters, such as the SCSS-Net, presents a significant challenge. Such a large model is unsuitable for embedded FPGA devices designed for edge cases, such as space use cases, due to its memory requirements. These requirements hinder inference time and increase the power consumption of the accelerator. We can take several steps to adapt such a network, including layer and weight pruning to reduce model size and memory footprint, and strategically modifying or removing skip connections. Skip connections require storing intermediate feature maps, which introduces data dependencies and memory transfers that complicate efficient FPGA pipelining.

In contrast, the YOLOv8n algorithm is the most compatible with FPGA implementation among those discussed. Its compact size and inherent dataflow architecture, where computations can be arranged in a highly parallel pipeline, make it an excellent candidate for hardware acceleration. This network has already been successfully accelerated in an embedded FPGA by Danilowicz and Kryjak (2025).

However, we need to conduct a more extensive study regarding its power consumption and inference time for our specific use case, similar to the feasibility study of various machine learning methods conducted by Antunes et al. (2025). A possible adaptation for FPGA deployment is to replace the SiLU activation layers with ReLUs. SiLU functions, while effective in software, involve more complex operations, such as the sigmoid. In contrast, ReLU's simpler piecewise linear nature is more efficient and resource-friendly for direct hardware implementation on FPGAs, often with negligible impact on performance.

Furthermore, we can also efficiently accelerate parts of the BCVO framework on FPGAs. Specifically, the Otsu thresholding algorithm has been shown to be suitable for FPGA acceleration by Barros et al. (2021) and the KNN

algorithm by Djosic et al. (2024). However, a comprehensive study must verify that memory access patterns or inherently sequential parts do not bottleneck the overall BCVO algorithm.

At present, deploying such AI frameworks fully on board spacecraft remains technically infeasible, primarily due to hardware and power constraints. Indicatively, the Solar Dynamics Observatory (SDO) operates with an available power of approximately 160 W and is equipped with a BAE RAD750 on-board computer (132 MHz CPU, 128 MB SDRAM, 256 KB start-up ROM). As a mission launched more than 15 years ago, SDO does not reflect the memory and computational capabilities of more recent spacecraft. Solar Orbiter, a newer mission, provides a comparable power budget of approximately 180 W, varying with mission phase and solar distance, and is equipped with a main on-board computer based on ESA's radiation-hardened ERC-32S SPARC architecture, operating at around 25 MHz and delivering approximately 20 MIPS (Marirrodriga et al., 2021). Despite being more recent, these specifications still impose severe limitations on the deployment of contemporary AI models.

More recently, FPGA-based prototypes operating at clock frequencies of around 100 MHz have demonstrated the feasibility of running lightweight AI inference in space-relevant environments, indicating a promising direction for future on-board intelligence (Gizzi et al., 2025). When developing such prototypes, achieving an implementation that can operate within a power budget of approximately 5 W would constitute an important first step. This would require model pruning and quantization, as well as a careful selection of input and output data cadence. These aspects are beyond the scope of the present work and are left for future investigation.

5. Conclusion

We compared three methods for detecting or segmenting solar coronal structures, focusing on both their performance and their suitability for onboard deployment in future spacecraft missions. Raw uncalibrated data have been used as an input to these types of methods.

After thoroughly evaluating their accuracy against each other and against state-of-the-art benchmarks, we focused on their computational efficiency and hardware compatibility. In particular, we discuss their suitability for FPGA-oriented implementations by analyzing their number of trainable parameters, network architecture and indicative general hardware requirements, while leaving detailed hardware emulation, acceleration strategies, and energy-efficiency benchmarking to future work. Among the evaluated approaches, the YOLO nano version, which is custom-trained for active region and coronal hole detection, demonstrated a promising balance between accuracy and model compactness. It remains to be further investigated whether its resource requirements are sufficiently low to enable reliable and efficient onboard embedding under the strict power and hardware constraints of space environments.

Appendix A: Coupling SCSS-Net/BCVO Segmentation With the Characterization Method

One can integrate the SCSS-Net or the more traditional BCVO method, with our proposed characterization approach described in Section 2.6.1. Figure A1 presents an output of this pipeline. Additionally, we demonstrate how a large number of segmentations are projected and then clustered using t-SNE.

Table A1
The List of 27 Features Extracted for the Detected Structure Characterization

Index	Feature name
1.	Mean
2.	Variance
3.	Energy
4.	Entropy
5.	Minimal gray level
6.	Maximal gray level
7.	ASM

Table A1
Continued

Index	Feature name
8.	Contrast
9.	Correlation
10.	Sum of squares variance
11.	Inverse difference moment
12.	Sum average
13.	Sum variance
14.	Sum entropy
15.	Entropy (GLCM)
16.	Difference variance
17.	Difference entropy
18.	Randomness
19.	Dependency
20.	Area
21.	Perimeter
22.	Eccentricity
23.	Extent
24.	Solidity
25.	Roundness
26.	Elongation
27.	Ratio perimeter and area

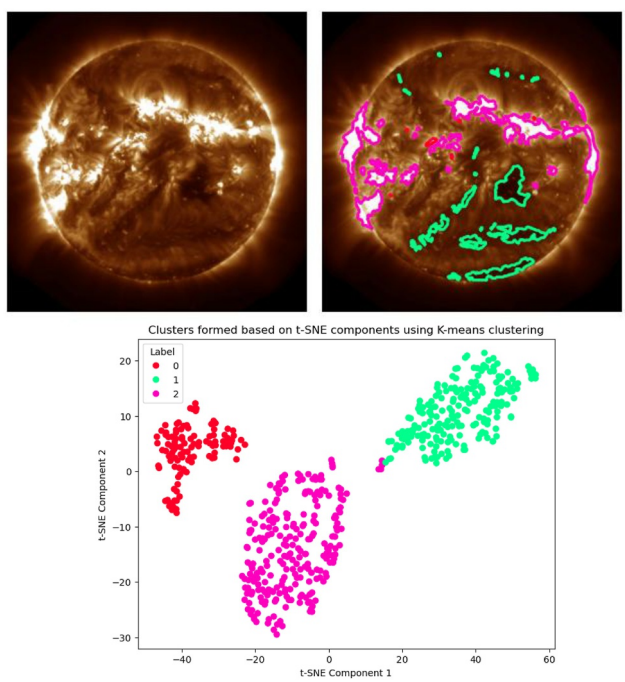

Figure A1. Illustrative example of the BCVO pipeline followed by the characterization method applied to Level-2 SDO/AIA data. Cluster 0 corresponds to bright points, cluster 1 to coronal holes, and cluster 2 to active regions.

This characterization relies on representing pixel values using carefully chosen features. More specifically, in Table A1, we present the set of extracted hand-crafted features used to represent and characterize the segmented regions. These features capture texture, shape, and contour complexity, providing unsupervised methods with richer descriptive information than raw pixel values alone. This set is subsequently reduced to 22 features by removing highly correlated ones, thereby lowering computational cost and facilitating more effective characterization.

Appendix B: Active Region and Coronal Hole Segmentation and Detection Using Level-0 SDO Data

Figures B1 and B2 provide a closer look at the SCSS-Net outputs when applied successfully to the AR model and unsuccessfully to the CH one on Level-0 SDO data. Once again, it is evident that the coronal hole model is particularly sensitive to preprocessing. Without careful correction, any quadrant discontinuities severely degrade performance.

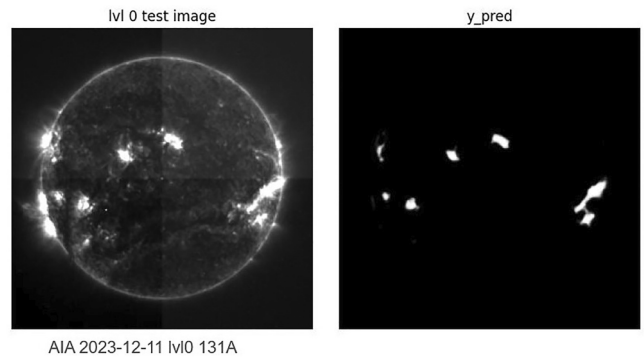

Figure B1. Applying SCSS-Net AR model to semi-processed level-0 data.

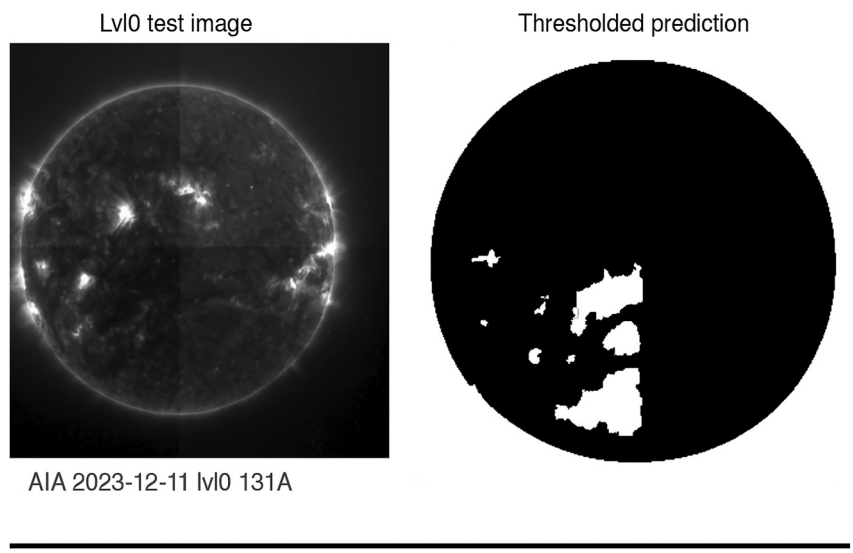

Figure B2. Unsuccessfully applying SCSS-Net CH model to semi-processed level-0 data.

Conflict of Interest

The authors declare no conflicts of interest relevant to this study.

Availability Statement

This work makes use of the publicly available machine-learning data set based on Solar Dynamics Observatory (SDO) observations (Galvez et al., 2019). The data set is publicly available through the AWS Marketplace (Galvez et al., 2020), while detailed documentation can be found at https://github.com/SDOML/sdoml.github.io. The source code developed for this study is also openly accessible to ensure full reproducibility of the results presented in this work. A static archived version is hosted on Zenodo (Gonidakis, 2026) (https://doi.org/10.5281/zenodo.18173333), while the actively maintained repository is available at https://github.com/pangonidakis/Solar-Structure-Detection-Methods-in-SDO-AIA-observations.

Acknowledgments

We gratefully acknowledge Angelos Vourlidas from Johns Hopkins University for his valuable insights. The research that led to these results was subsidized by the European Union. ASAP has received funding from the European Union's HORIZON Research and Innovation Action under the Grant Agreement No 101082633. Views and opinions expressed are, however, those of the author(s) only and do not necessarily reflect those of the European Union. This study was additionally supported by the Belgian Defence – Royal Higher Institute for Defence (RHID) through contract no 22DEFRA006. We thank the Belgian Federal Science Policy Office (BELSPO) for providing financial support within the framework of the Brain-be program, under contract number B2/202/P1/DELPHI. G. M. has received funding from the European Union's Horizon research and innovation program under the Marie Skłodowska-Curie grant agreement No 101148539 and the Research Foundation – Flanders (FWO), HELIOSKILL project grant (G0B9923N). F.C. and G.L. have received funding from Horizon Europe ERC-2022-ADG Ref. 101095310. S.R. acknowledges funding from the Johns Hopkins University Applied Physics Laboratory independent R&D. S.P. and H.-J.J. are funded by the European Union (ERC, Open SESAME, 101141362). Views and opinions expressed are, however, those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them. S.P. is also supported by the projects C16/24/010 (C1 project Internal Funds KU Leuven), G0B5823N and G002523N (WEAVE) (FWO-Vlaanderen), and 4000145223 (SIDC Data Exploitation (SIDEX2), ESA Prodex). H.-J.J. also acknowledges support from the BK21 FOUR program of the Graduate School, Kyung Hee University (GS-1-JO-NON-20242364 and GS-1-JO-NON-20253142). The resources and services used in this work were provided by the VSC (Flemish Supercomputer Center), funded by the FWO and the Flemish Government. Finally, the authors declare that there are no conflicts of interest.

References

Anscombe, F. J. (1948). The transformation of Poisson, binomial and negative-binomial data. *Biometrika*, *35*(3/4), 246–254. https://doi.org/10.2307/2332343

Antunes, P., Al Hafiz, M. I., Ekelund, J., Dineva, E., Miloshevich, G., Gonidakis, P., & Podobas, A. (2025). Evaluating four FPGA-accelerated space use cases based on neural network algorithms for on-board inference. In *2025 IEEE 18th international symposium on embedded Multicore/Many-core Systems-on-Chip (MCSoC)* (pp. 804–812). https://doi.org/10.1109/MCSoC67473.2025.00126

Antunes, P., & Podobas, A. (2025). FPGA-based neural network accelerators for space applications: A survey. Retrieved from https://arxiv.org/abs/2504.16173

Arora, S., Acharya, J., Verma, A., & Panigrahi, P. K. (2008). Multilevel thresholding for image segmentation through a fast statistical recursive algorithm. *Pattern Recognition Letters*, *29*(2), 119–125. https://doi.org/10.1016/j.patrec.2007.09.005

Asensio Ramos, A., Cheung, M. C., Chifu, I., & Gafeira, R. (2023). Machine learning in solar physics. *Living Reviews in Solar Physics*, *20*(1), 4. https://doi.org/10.1007/s41116-023-00038-x

Auchère, F., Andretta, V., Antonucci, E., Bach, N., Battaglia, M., Bemporad, A., et al. (2020). Coordination within the remote sensing payload on the solar orbiter mission. *Astronomy & Astrophysics*, *642*, A6. https://doi.org/10.1051/0004-6361/201937032

Bahl, G., Daniel, L., Moretti, M., & Lafarge, F. (2019). Low-power neural networks for semantic segmentation of satellite images. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)* (pp. 2469–2476). https://doi.org/10.1109/ICCVW.2019.00302

Barra, V., Delouille, V., Kretzschmar, M., & Hochedez, J.-F. (2009). Fast and robust segmentation of solar EUV images: Algorithm and results for solar cycle 23. *Astronomy & Astrophysics*, *505*(1), 361–371. https://doi.org/10.1051/0004-6361/200811416

Barros, W. K. P., Dias, L. A., & Fernandes, M. A. C. (2021). Fully parallel implementation of Otsu automatic image thresholding algorithm on FPGA. *Sensors*, *21*(12), 4151. https://doi.org/10.3390/s21124151

Benz, A. O. (2017). Flare observations. *Living Reviews in Solar Physics*, *14*(1), 2. https://doi.org/10.1007/s41116-016-0004-3

Boada Gardenyes, R. (2011). Trends and patterns in ASIC and FPGA use in space missions and impact in technology roadmaps of the European Space Agency.

Bobra, M. G., Sun, X., Hoeksema, J. T., Turmon, M., Liu, Y., Hayashi, K., et al. (2014). The Helioseismic and Magnetic Imager (HMI) vector magnetic field pipeline: SHARPs–Space-Weather HMI active region patches. *Solar Physics*, *289*(9), 3549–3578. https://doi.org/10.1007/s11207-014-0529-3

Bobra, M. G., Wright, P. J., Sun, X., & Turmon, M. J. (2021). SMARPs and SHARPs: Two solar cycles of active region data. *The Astrophysical Journal Supplement Series*, *256*(2), 26. https://doi.org/10.3847/1538-4365/ac1f1d

Boerner, P., Edwards, C., Lemen, J., Rausch, A., Schrijver, C., Shine, R., et al. (2012). Initial calibration of the atmospheric imaging assembly (AIA) on the solar dynamics observatory (SDO). *Solar Physics*, *275*(1), 41–66. https://doi.org/10.1007/s11207-011-9804-8

Cosmas, K., & Kenichi, A. (2020). Utilization of FPGA for onboard inference of landmark localization in CNN-based spacecraft pose estimation. *Aerospace*, *7*(11), 159. https://doi.org/10.3390/aerospace7110159

Cranmer, S. R. (2002). Coronal holes and the high-speed solar wind. *Space Science Reviews*, *101*(3), 229–294. https://doi.org/10.1023/a:1020840004535

Danilowicz, M., & Kryjak, T. (2025). Real-time multi-object tracking using YOLOv8 and SORT on a SoC FPGA. Retrieved from https://arxiv.org/abs/2503.13023

Delaboudiniere, J.-P., Artzner, G., Brunaud, J., Gabriel, A. H., Hochedez, J.-F., Millier, F., et al. (1995). EIT: Extreme-ultraviolet imaging telescope for the SOHO mission. *Solar Physics*, *162*(1), 291–312.

Delouille, V., Hofmeister, S. J., Reiss, M. A., Mampaey, B., Temmer, M., & Veronig, A. (2018). Coronal holes detection using supervised classification. In *Machine learning techniques for space weather* (pp. 365–395). Elsevier.

Djosic, S., Jovanovic, M., & Lj. Djordjevic, G. (2024). Optimized k-Nearest neighbors search implementation on resource-constrained FPGA platforms. *Microprocessors and Microsystems*, *109*, 105089. https://doi.org/10.1016/j.micpro.2024.105089

Domingo, V., Fleck, B., & Poland, A. I. (1995). The SOHO mission: An overview. *Solar Physics*, *162*(1), 1–37.

Gallon, R., Schiemenz, F., Menicucci, A., & Gill, E. (2025). Convolutional neural network design and evaluation for real-time multivariate time series fault detection in spacecraft attitude sensors. *Advances in Space Research*, *76*(5), 2960–2976. https://doi.org/10.1016/j.asr.2025.06.068

Galvez, R., Fouhey, D. F., Jin, M., Szenicer, A., Muñoz-Jaramillo, A., Cheung, M. C., et al. (2019). A machine-learning data set prepared from the NASA solar dynamics observatory mission. *The Astrophysical Journal Supplement Series*, *242*(1), 7. https://doi.org/10.3847/1538-4365/ab1005

Galvez, R., Fouhey, D. F., Jin, M., Szenicer, A., Muñoz-Jaramillo, A., Cheung, M. C., et al. (2020). Solar dynamics observatory (SDO) machine learning dataset [Dataset]. *AWS*. https://doi.org/10.3847/1538-4365/ab1005

Garton, T. M., Gallagher, P. T., & Murray, S. A. (2018). Automated coronal hole identification via multi-thermal intensity segmentation. *Journal of Space Weather and Space Climate*, *8*, A02. https://doi.org/10.1051/swsc/2017039

Ghiglione, M., & Serra, V. (2022). Opportunities and challenges of AI on satellite processing units. In *Proceedings of the 19th ACM international conference on computing frontiers* (pp. 221–224).

Gizzi, E., Chase Jr., T., Cassamajor-Paul, C., Chertok, R., Clough, L., Firth, C., et al. (2025). OnAIR: Applications of the NASA on-board artificial intelligence research platform. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 39, pp. 28893–28899).

Gonidakis, P. (2026). Solar-Structure-Detection-Methods-in-SDO-AIA-observations: Release v1 for zenodo [Software]. *Zenodo*. https://doi.org/10.5281/zenodo.18173334

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Convolutional networks. *Deep learning*, *1*(2).
Guerrisi, G., Del Frate, F., & Schiavon, G. (2023). Artificial intelligence based on-board image compression for the Φ-Sat-2 mission. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *16*, 8063–8075. https://doi.org/10.1109/jstars.2023.3296485
Hapgood, M., Dimbylow, T., Sutcliffe, D., Chaizy, P., Ferron, P., Hill, P., & Tiratay, X. (1997). The joint science operations centre. *Space Science Reviews*, *79*(1), 487–525. https://doi.org/10.1023/A:1004954504114
Henney, C., & Harvey, J. (2005). Large-scale structures and their role in solar activity. In K. Sankarasubramanian, M. Penn, & A. Pevtsov (Eds.), *ASP Conference Series* (Vol. 346, p. 261).
Howard, R. A., Moses, J., Vourlidas, A., Newmark, J., Socker, D. G., Plunkett, S. P., et al. (2008). Sun Earth connection coronal and heliospheric investigation (SECCHI). *Space Science Reviews*, *136*(1), 67–115. https://doi.org/10.1007/s11214-008-9341-4
Howard, R. A., Vourlidas, A., Korendyke, C. M., Plunkett, S. P., Carter, M. T., Wang, D., et al. (2013). The solar and heliospheric imager (SoloHI) instrument for the solar orbiter mission. In *Solar physics and space weather instrumentation* (Vol. 8862, pp. 155–167).
Hurlburt, N., Cheung, M., Schrijver, C., Chang, L., Freeland, S., Green, S., et al. (2012). Heliophysics event knowledgebase for the solar dynamics observatory (SDO) and beyond. In *The solar dynamics observatory* (pp. 67–78). Springer.
Intel Corporation. (2023). Intel Movidius Myriad vision processing units. Retrieved from https://www.intel.com/content/www/us/en/developer/tools/openvino-toolkit/overview.html
Jarolim, R., Veronig, A., Hofmeister, S., Heinemann, S., Temmer, M., Podladchikova, T., & Dissauer, K. (2021). Multi-channel coronal hole detection with convolutional neural networks. *Astronomy & Astrophysics*, *652*, A13. https://doi.org/10.1051/0004-6361/202140640
Jiang, P., Ergu, D., Liu, F., Cai, Y., & Ma, B. (2022). A review of yolo algorithm developments. *Procedia Computer Science*, *199*, 1066–1073. https://doi.org/10.1016/j.procs.2022.01.135
Jocher, G., & Qiu, J. (2024). Ultralytics YOLO11. Retrieved from https://github.com/ultralytics/ultralytics
Kasapis, S., Thompson, B. J., Rodriguez, J. V., Attie, R., Cucho-Padin, G., Da Silva, D., et al. (2023). Turning noise into data: Characterization of the Van Allen radiation belt using SDO spikes data. *Space Weather*, *21*(3), e2022SW003310. https://doi.org/10.1029/2022sw003310
Kim, J.-H., Kim, Y., Cho, D.-H., & Kim, S.-M. (2024). On-Orbit AI: Cloud detection technique for resource-limited nanosatellite. *International Journal of Aeronautical and Space Sciences*, *26*(4), 1975–1988. https://doi.org/10.1007/s42405-024-00865-8
Koonce, B. (2021). ResNet 50. In *Convolutional neural networks with swift for tensorflow: Image recognition and dataset categorization* (pp. 63–72). Springer.
Krista, L. D., & Gallagher, P. T. (2009). Automated coronal hole detection using local intensity thresholding techniques. *Solar Physics*, *256*(1–2), 87–100. https://doi.org/10.1007/s11207-009-9357-2
Lemen, J. R., Title, A. M., Akin, D. J., Boerner, P. F., Chou, C., Drake, J. F., et al. (2012). The atmospheric imaging assembly (AIA) on the solar dynamics observatory (SDO). *Solar Physics*, *275*(1), 17–40.
Liao, P.-S., Chen, T.-S., & Chung, P.-C. (2001). A fast algorithm for multilevel thresholding. *Journal of Information Science and Engineering*, *17*(5), 713–727.
Linker, J. A., Heinemann, S. G., Temmer, M., Owens, M. J., Caplan, R. M., Arge, C. N., et al. (2021). Coronal hole detection and open magnetic flux. *The Astrophysical Journal*, *918*(1), 21. https://doi.org/10.3847/1538-4357/ac090a
Lloyd, S. (1982). Least squares quantization in PCM. *IEEE Transactions on Information Theory*, *28*(2), 129–137. https://doi.org/10.1109/tit.1982.1056489
Mackovjak, Š., Harman, M., Maslej-Krešňáková, V., & Butka, P. (2021). SCSS-Net: Solar corona structures segmentation by deep learning. *Monthly Notices of the Royal Astronomical Society*, *508*(3), 3111–3124. https://doi.org/10.1093/mnras/stab2536
MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. In *Berkeley symposium on mathematical statistics and probability* (pp. 281–297).
Marirrodriga, C. G., Pacros, A., Strandmoe, S., Arcioni, M., Arts, A., Ashcroft, C., et al. (2021). Solar Orbiter: Mission and spacecraft design. *Astronomy & Astrophysics*, *646*, A121. https://doi.org/10.1051/0004-6361/202038519
Müller, D., Cyr, O. S., Zouganelis, I., Gilbert, H. R., Marsden, R., Nieves-Chinchilla, T., et al. (2020). The solar orbiter mission-science overview. *Astronomy & Astrophysics*, *642*, A1.
Nandi, D., Anupam, A., Roy, S., Prasad, A., & Patra, S. N. (2024). Enhanced precision in automatic identification of Coronal Hole regions in solar images using the proposed Supervised Intensity Thresholding with Distance Transform Clustering and Connected Component Labeling (SITDTCCCL) method on data collected by the Solar Dynamic Observatory between January 2019 and July 2023. *Research Square*.
Ni, S., Wei, X., Zhang, N., & Chen, H. (2023). Algorithm–hardware co-optimization and deployment method for field-programmable gate-array-based convolutional neural network remote sensing image processing. *Remote Sensing*, *15*(24), 5784. https://doi.org/10.3390/rs15245784
NVIDIA Corporation. (2023). Nvidia Jetson platform for edge AI. Retrieved from https://developer.nvidia.com/embedded-computing
Otsu, N. (1975). A threshold selection method from gray-level histograms. *Automatica*, *11*(285–296), 23–27.
Pérez-Suárez, D., Higgins, P. A., Bloomfield, D. S., McAteer, R. J., Krista, L. D., Byrne, J. P., & Gallagher, P. T. (2013). Automated solar feature detection for space weather applications. In *Image processing: Concepts, methodologies, tools, and applications* (pp. 979–997).
Pesnell, W. D., Thompson, B. J., & Chamberlin, P. (2012). The solar dynamics observatory (SDO). In *The solar dynamics observatory* (pp. 3–15). Springer.
Pitonak, R., Mucha, J., Dobis, L., Javorka, M., & Marusin, M. (2022). CloudSatNet-1: FPGA-based hardware-accelerated quantized CNN for satellite on-board cloud coverage classification. *Remote Sensing*, *14*(13), 3180. https://doi.org/10.3390/rs14133180
Ramachandran, S., Rosengarten, M., & Belardi, C. (2020). Semi-supervised machine learning for spacecraft anomaly detection diagnosis. In *2020 IEEE aerospace conference* (pp. 1–10). https://doi.org/10.1109/AERO47225.2020.9172454
Redmon, J., & Farhadi, A. (2018). YOLOv3: An incremental improvement. arXiv preprint arXiv:1804.02767.
Reiss, M. A., Hofmeister, S. J., De Visscher, R., Temmer, M., Veronig, A. M., Delouille, V., et al. (2015). Improvements on coronal hole detection in SDO/AIA images using supervised classification. *Journal of Space Weather and Space Climate*, *5*, A23. https://doi.org/10.1051/swsc/20155025
Reiss, M. A., Muglach, K., Möstl, C., Arge, C. N., Bailey, R., Delouille, V., et al. (2021). The observational uncertainty of coronal hole boundaries in automated detection schemes. *The Astrophysical Journal*, *913*(1), 28. https://doi.org/10.3847/1538-4357/abf2c8
Reiter, P., Karagiannakis, P., Ireland, M., Greenland, S., & Crockett, L. (2020). FPGA acceleration of a quantized neural network for remote-sensed cloud detection. In *7th international workshop on On-Board payload data compression*.
Rochus, P., Auchere, F., Berghmans, D., Harra, L., Schmutz, W., Schühle, U., et al. (2020). The solar orbiter EUI instrument: The extreme ultraviolet imager. *Astronomy & Astrophysics*, *642*, A8. https://doi.org/10.1051/0004-6361/201936663
Ronneberger, O., Fischer, P., & Brox, T. (2015). 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* (Vol. 18, pp. 234–241).

Ržička, V., Mateo-García, G., Bridges, C., Brunskill, C., Purcell, C., Longépé, N., & Markham, A. (2023). Fast model inference and training on-board of satellites. In *IGARSS 2023-2023 IEEE international geoscience and remote sensing symposium* (pp. 2002–2005).

Satopaa, V., Albrecht, J., Irwin, D., & Raghavan, B. (2011). Finding a "kneedle" in a haystack: Detecting knee points in system behavior. In *2011 31st international conference on distributed computing systems workshops* (pp. 166–171). https://doi.org/10.1109/ICDCSW.2011.20

Scherrer, P. H., Schou, J., Bush, R., Kosovichev, A., Bogart, R., Hoeksema, J., et al. (2012). The helioseismic and magnetic imager (HMI) investigation for the solar dynamics observatory (SDO). *Solar Physics*, *275*(1), 207–227.

Schmieder, B., Démoulin, P., & Aulanier, G. (2013). Solar filament eruptions and their physical role in triggering coronal mass ejections. *Advances in Space Research*, *51*(11), 1967–1980. https://doi.org/10.1016/j.asr.2012.12.026

Schrijver, C. J., & Siscoe, G. L. (2010). *Heliophysics: Space storms and radiation: Causes and effects*. Cambridge University Press.

Shao, Y., Shang, J., Li, Y., Ding, Y., Zhang, M., Ren, K., & Liu, Y. (2024). A configurable accelerator for CNN-based remote sensing object detection on FPGAS. *IET Computers & Digital Techniques*, *2024*(1), 4415342. https://doi.org/10.1049/2024/4415342

Tlatov, A., Tavastsherna, K., & Vasil'eva, V. (2014). Coronal holes in solar cycles 21 to 23. *Solar Physics*, *289*(4), 1349–1358. https://doi.org/10.1007/s11207-013-0387-4

Toriumi, S., & Wang, H. (2019). Magnetic properties and evolution of solar active regions. *Living Reviews in Solar Physics*, *16*(1), 3. https://doi.org/10.1007/s41116-019-0019-7

Verbeeck, C., Delouille, V., Mampaey, B., & De Visscher, R. (2014). The SPoCA-suite: Software for extraction, characterization, and tracking of active regions and coronal holes on EUV images. *Astronomy & Astrophysics*, *561*, A29. https://doi.org/10.1051/0004-6361/201321243

Wang, Z., Xu, K., Wu, S., Liu, L., Liu, L., & Wang, D. (2020). Sparse-YOLO: Hardware/software co-design of an FPGA accelerator for YOLOv2. *IEEE Access*, *8*, 116569–116585. https://doi.org/10.1109/access.2020.3004198

Webb, D. F., & Howard, T. A. (2012). Coronal mass ejections: Observations. *Living Reviews in Solar Physics*, *9*(1), 1–83. https://doi.org/10.12942/lrsp-2012-3

Woods, T. N., Eparvier, F., Hock, R., Jones, A., Woodraska, D., Judge, D., et al. (2012). Extreme ultraviolet variability experiment (EVE) on the solar dynamics observatory (SDO): Overview of science objectives, instrument design, data products, and model developments. In *The solar dynamics observatory* (pp. 115–143).