







# StreakMind: AI detection and analysis of satellite streaks in astronomical images with automated database integration

Rafael Carrillo<sup>1,2,3,\*</sup> , René Duffard<sup>3</sup> , Pablo García-Martín<sup>4</sup> , Javier Romero<sup>5</sup> ,  
Nicolás Morales<sup>3</sup> , and Luis Gonçalves<sup>6</sup> 

<sup>1</sup> Real Instituto y Observatorio de la Armada (ROA), Plaza de las Marinas s/n, 11100 San Fernando (Cádiz), Spain

<sup>2</sup> Universidad de Granada, Departamento de Física Teórica y del Cosmos, 18071 Granada, Spain

<sup>3</sup> Instituto de Astrofísica de Andalucía – CSIC, Apdo. 3004, 18080 Granada, Spain

<sup>4</sup> Safran, 171 Bd de Valmy, 92700 Colombes, France

<sup>5</sup> Universidad de Cádiz, Departamento de Matemáticas, 11510 Puerto Real (Cádiz), Spain

<sup>6</sup> CFisUC, Departamento de Física, Universidade de Coimbra, 3004-516 Coimbra, Portugal

Received 23 December 2025 / Accepted 4 March 2026

## ABSTRACT

**Context.** Artificial satellites and space debris are increasingly contaminating astronomical images, affecting scientific surveys and producing large volumes of streaked exposures. Manual inspection is no longer feasible at scale, and reliable identification and characterisation of streaks has become essential for both the quality control of data and the monitoring of objects in Earth orbit.

**Aims.** We present StreakMind, an automated pipeline designed to detect near-Earth objects (NEOs) and satellite streaks in astronomical images, characterise their geometry, and cross-identify them with known orbital objects. The system integrates all inference results into a structured database suitable for large surveys.

**Methods.** A YOLO-OBB model was trained on a hybrid manual–synthetic dataset of 2335 images and used to detect streaks in processed FITS frames. Geometric refinement, inter-frame association, satellite cross-identification, and Gaussian-based confidence scoring were then applied to produce final identifications, which were stored in a normalised relational database. In this work, images acquired at La Sagra Observatory (L98) with a Celestron C14+Fastar telescope were used to develop and test automated streak detection and characterisation methods.

**Results.** On the test set, the model achieved a precision of 94% and a recall of 97%. It reliably detected faint streaks, delivered consistent geometric reconstructions across the dataset, and performed robust satellite cross-identification. The Gaussian-based confidence scoring provided stable identification probabilities across consecutive frames.

**Conclusions.** StreakMind demonstrates strong potential for large-scale automated analyses of linear streaks produced by both NEOs and artificial satellites in ground-based astronomical images. The pipeline offers high detection reliability, robust geometric reconstruction, and reproducible satellite cross-identification within a fully integrated end-to-end framework.

**Key words.** methods: data analysis – astronomical databases: miscellaneous – minor planets, asteroids: general

## 1. Introduction

The advent of wide-field high-cadence surveys has revolutionised the detection of near-Earth objects (NEOs), defined as asteroids with an orbit within 1.3 AU from the Sun (Bottke et al. 2002), shifting NEO detection and Solar System survey science into an era of big data. Modern facilities now monitor the sky with unprecedented speed, generating a volume of imagery that renders manual inspection unfeasible. While the discovery of these Solar System objects remains a primary scientific goal and the cornerstone for planetary defense, this continuous surveillance is equally critical for space situational awareness, especially as it relates to the monitoring of the growing population of artificial satellites and debris essential for maintaining orbital operations. Consequently, detection efforts must now navigate a crowded dynamic environment and distinguish faint NEOs from the dense traffic of Earth’s growing in-orbit technological ecosystem.

Several methods can be used to find fast moving objects close to Earth in astronomical images. One approach uses a

type of algorithm that relies on the identification of linear features and is based on methods such as the Hough transform (Billot et al. 2024; Gonçalves et al. 2026) and the Radon transform (Stark et al. 2022). Alternatively, synthetic tracking has emerged as a powerful technique for discovering faint NEOs. By digitally shifting and stacking multiple short exposures along hypothetical velocity vectors, this method dramatically increases the signal-to-noise ratio (S/N) of moving targets that are otherwise undetectable in individual frames (Stănescu et al. 2025; Vaduvescu et al. 2025; Zhai et al. 2024). In this study, we describe a machine learning tool we designed to detect moving objects in astronomical images.

Some previous works have highlighted the advantages of applying machine learning techniques to transient detection in astronomical images. From space, these methods have been employed to detect streaks in Hubble Space Telescope observations (Kruk et al. 2022), in the CHEOPS mission archive (García-Martín et al. in prep.), and in simulated imagery for the Euclid mission (Pöntinen et al. 2023; Lieu et al. 2019). From the ground, the Zwicky Transient Facility has a long history of integrating deep learning to automate the identification of fast-moving objects. Duev et al. (2019) introduced a classifier

\* Corresponding author: raficarrillo02@gmail.com

operating on difference images, reducing human scanning time. Most recently, [Irueta-Goyena et al. \(2025\)](#) advanced this effort by implementing an object segmentation approach trained on a mix of real and synthetic data, demonstrating the ability to recover hundreds of valid streaks missed by human scanners.

With this paper we present StreakMind, a comprehensive AI-based data pipeline designed for the automated detection and characterisation of streaks in ground-based astronomical images. The detection stage is generic and applicable to any linear moving feature; however, the identification component developed in this work focuses on artificial satellites and space debris through cross-matching against external ephemerides. All detections are integrated into a centralised database that supports cross-referencing with catalogues of artificial objects. The extension of the identification layer to NEOs is planned as future work.

Automatically detecting and characterising faint linear streaks produced by artificial satellites and debris in large volumes of astronomical imaging data is a significant challenge, particularly when these artefacts must be distinguished from genuine Solar System objects and instrumental effects. We developed an end-to-end pipeline to (i) detect linear streaks in ground-based astronomical images, (ii) refine their geometry and associate detections across consecutive frames, (iii) standardise the resulting measurements into Minor Planet Center (MPC)-style records, and (iv) cross-identify candidate artificial objects using external ephemerides, integrating all outputs into a relational database suitable for large-scale analyses.

## 2. Methods and data

This section describes the data and methodological framework used to develop and evaluate the StreakMind pipeline. We first describe the detection pipeline and model architecture (Section 2.1). We then summarise the telescope instrumentation and the FITS dataset (Sections 2.2–2.3), followed by the generation of synthetic streaks (Section 2.4). The subsequent subsection details the labelling procedure and the FITS-to-PNG conversion including oriented bounding boxes (OBBs; Section 2.5). Finally, Section 2.6 outlines the stratified sampling strategy adopted for training, validation, and testing.

At the core of this workflow lies the automated detection of linear streaks in individual images. This task is addressed using a deep-learning-based object detector, whose architecture and configuration are described in the following subsection.

### 2.1. Pipeline architecture

You Only Look Once (YOLO) is a family of real-time object detection models that predict both the location and the category of objects in a single pass through a neural network. By combining object localisation and classification into a single-stage process, YOLO enables fast and efficient inference, making it suitable for time-critical applications. The YOLO11 version, introduced in 2024 ([Jocher & Qiu 2024](#)), incorporates several architectural improvements that enhance both detection accuracy and computational performance. The model maintains the traditional three-part structure consisting of a backbone, neck, and head. The backbone extracts visual features from the input image at different levels of detail. The neck then combines these features across multiple scales, allowing the model to detect both small and large objects more effectively. Finally, the head processes this information to produce the final predictions,

including object positions and class labels. Further details on the architectural components and multitask capabilities of YOLO11 can be found in [Khanam & Hussain \(2024\)](#).

For this work, we adopted a pretrained YOLO11 model configured for OBBs, pretrained on the DOTAv1.0 dataset ([Xia et al. 2018](#)). For the specific purpose of streak detection, the network was subsequently trained on an augmented dataset comprising both real and synthetically generated astronomical images, as described in detail in Section 2.4. Training was carried out on a cloud-based NVIDIA A100 GPU, using eight worker threads to accelerate data loading.

### 2.2. Telescope and instrumentation

Observations were conducted at the La Sagra Observatory, located at latitude 37.982402°N, longitude 2.565290°W, at an altitude of 1513.654841 metres (MPC code L98). The telescope used is a Celestron C14+Fastar f/2.1 reflector, designated internally as Tetra1, mounted on a German Paramount mount. The instrument has an aperture of 356 mm and a focal length of 712 mm. It is equipped with an SBIG ST-10 3 CCD camera, providing a pixel scale of 4.12 arcsec/pixel and a field of view of 74.9 × 50.5 arcminutes. The native pixel size of the detector is 6.8 μm, and images were acquired with 2 × 2 binning. This binning mode was adopted to reduce data volume and facilitate nightly data transfer, as the observations were conducted within the framework of an external collaboration requiring the download of the full datasets after each observing session. The substantial pixel scale results from the integration of Fastar optics at the primary focus, reducing the focal ratio from f/10 to f/2. While this configuration increases the image scale, it successfully provides the wider field of view required for the instrument's primary scientific applications. No physical photometric filter was employed during acquisition.

### 2.3. FITS dataset description

To train and evaluate the StreakMind pipeline, we relied on a combination of real astronomical images and synthetically generated data. This subsection describes the real FITS image dataset and motivates the need for the inclusion of synthetic streaks, whose generation and properties are detailed separately in Section 2.4.

The real dataset consists of a total of 2055 FITS images, originally acquired for astrometric and photometric studies of asteroids, with the target objects generally located near the centre of the field of view. Table 1 summarises the observing dates between April and June 2019, indicating the specific asteroid targeted each night and the number of images obtained.

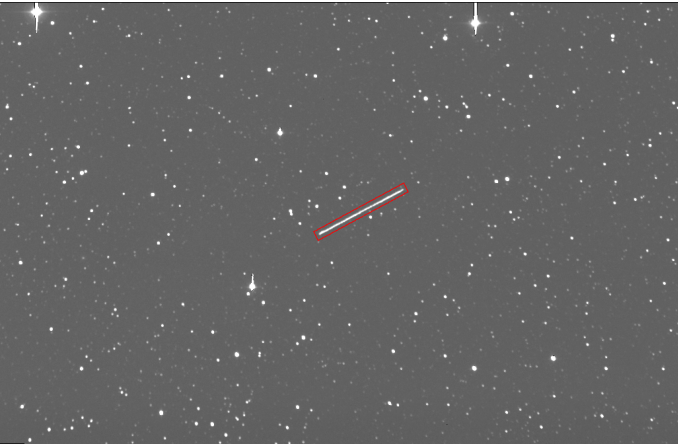
Each FITS image measures 1092 × 736 pixels and includes World Coordinate System (WCS) headers for astrometric calibration. Exposure times ranged from 8 to 120 seconds, achieving approximate limiting magnitudes between 19 and 20 under typical observing conditions. Observations were conducted under varying seeing conditions and sky brightness levels.

All 2055 images were calibrated using flat-field and dark frames matching the same exposure times. Subsequently, the images were manually inspected by multiple analysts to determine the presence of linear streaks. Across the dataset, a total of 765 streaks were identified, with measured lengths ranging from 8.5 pixels to 1161.7 pixels, and a mean length of 203.5 pixels.

The distribution of streak lengths in the real dataset is highly skewed, with a substantially smaller number of long streaks compared to short ones. To enable a controlled stratification of the

**Table 1.** Distribution of FITS images included in the dataset.

| Date       | Number of images | Image target                 |
|------------|------------------|------------------------------|
| 2019-04-11 | 77               | 00360 (Carlova)              |
| 2019-04-12 | 108              | 00360 (Carlova)              |
| 2019-04-13 | 77               | 00360 (Carlova)              |
| 2019-04-26 | 51               | 00145 (Adeona)               |
| 2019-04-27 | 262              | 00145 (Adeona)               |
| 2019-04-30 | 124              | 00068 (Leto)                 |
| 2019-05-10 | 337              | 00145 (Adeona)               |
| 2019-05-11 | 280              | 00145 (Adeona)               |
| 2019-05-12 | 122              | 00218 (Bianca); 00068 (Leto) |
| 2019-05-13 | 21               | 00068 (Leto)                 |
| 2019-05-14 | 80               | 00489 (Comacina)             |
| 2019-06-03 | 98               | 00031 (Euphrosyne)           |
| 2019-06-04 | 84               | 00031 (Euphrosyne)           |
| 2019-06-05 | 66               | 00031 (Euphrosyne)           |
| 2019-06-06 | 251              | 00218 (Bianca)               |
| 2019-06-07 | 17               | 00134 (Sophrosyne)           |

**Fig. 1.** Real image from the dataset obtained during the observation on the night of May 10, 2019, with the La Sagra telescope. An OBB has been overlaid to illustrate the resulting annotation as applied to actual observational data.

dataset (described in Section 2.6), the 75th percentile of the streak-length distribution, corresponding to 269.1 pixels, was adopted as the threshold separating short and long streaks. In cases where an image contained multiple streaks, the average streak length within that image was used, since the object of study is the image itself rather than individual streaks. Based on this criterion, the dataset comprises 1523 images without streaks, 412 images containing short streaks, and 120 images containing long streaks. Owing to the resulting imbalance between the short- and long-streak classes, synthetic trails were introduced to increase the representation of long streaks in the dataset. Figure 1 shows a representative example from the dataset, illustrating an OBB annotation overlaid on a real observational image containing a linear streak.

For each observing night, all images were aligned to a common reference frame in order to streamline the subsequent reduction process. This procedure was applied uniformly across the dataset. As a consequence of small pointing variations and tracking drifts of the telescope, only the region common to all aligned images was preserved, resulting in alignment-induced dead margins in the final images, as discussed in Sect. 3.6.

#### 2.4. Synthetic data generation

To address the limitations identified in the real FITS dataset, particularly the under-representation of long streaks, synthetic streaks were generated and incorporated into the training set as described below. Synthetic streaks were added to real images using custom Python scripts. For five empirically defined brightness levels, ranging from faint to bright, either full-crossing or partial linear features of random length were injected into the images. A second streak was added in 10% of the cases to simulate multiple-satellite passages. All synthetic streaks were placed at random locations within the image. A minimum synthetic length of 269 px – corresponding to the 75th percentile of the real streak-length distribution – was imposed to ensure an adequate representation of long streaks in the augmented dataset.

In contrast, the angular distribution of the synthetic streaks was designed to resemble the real dataset. Random orientations were drawn from a Gaussian distribution, with 70% of the simulated lines lying between  $-60^\circ$  and  $60^\circ$ , reflecting the dominant range observed in the real measurements.

To correctly simulate the sensor’s effect, the point spread function (PSF) was estimated for each image using PSFEx (PSF Extractor) (Bertin 2013). The resulting PSF was then convolved, via Fourier transforms, with a one-pixel-wide linear feature to simulate the appearance of streaks on the detector.

A total of 280 additional images containing synthetic streaks were incorporated into the dataset to improve the model performance in identifying long streaks. Among these images, approximately 69% (193) contain synthetic streaks that cross the entire image from side to side, while the remaining 31% (87) contain streaks that remain fully within the field of view.

#### 2.5. Labelling and FITS $\rightarrow$ PNG conversion

Having described both the real and synthetic data components, we now detail how the annotated information is prepared for use by the detection model. This subsection describes the manual labelling procedure, the conversion of FITS images into PNG format, and the construction of pixel-based OBBs used for training and evaluation. Annotations were performed manually using the Tycho Tracker software (Parrott 2020).

For each detected streak, a comprehensive set of parameters was meticulously measured and documented, reflecting both astrometric and photometric characteristics as well as geometric details of the streak. The following properties were individually determined for every streak segment:

- Date – Date of observation. (e.g. 2019 04 12.04618).
- Observatory – Name of the observatory (e.g. La Sagra – IAA).
- MPC\_code – MPC observatory code (e.g. L98).
- Telescope – Telescope used for the observation (e.g. TETRA 1 – Reflector f/2 – 0.3 5m).
- Sensor – Detector or camera model (e.g. SBIG ST-10 3 CCD Camera).
- Image Name – file name of the image, including date and additional identifiers such as object number, sequence, repetition, chip, and filter (e.g. 20190411\_00360-S003-R001-C001-NoFilt).
- Streak Multiplicity – indicates whether the streak is unique in the frame or if multiple streaks are present (e.g. Single Streak).
- Streak Completeness – indicates whether the streak is complete or incomplete.

- `Track_Id` – internal identifier assigned to the streak (e.g. ABC0021).
- `Central_RA` – right Ascension of the streak’s centre point (e.g. 16 48 47.16).
- `Central_DEC` – declination of the streak’s centre point (e.g. –07 47 35.8).
- `RA_Mark1` – right Ascension of the streak’s starting point (e.g. 16 49 42.20).
- `DEC_Mark1` – declination of the streak’s starting point (e.g. –07 49 37.2).
- `RA_Mark2` – right Ascension of the streak’s end point. (e.g. 16 47 51.82).
- `DEC_Mark2` – declination of the streak’s end point (e.g. –07 45 35.0).
- `Pixel_Mark1` – pixel coordinates ( $x, y$ ) of the streak’s starting point in the image frame – for example, (87.00, 670.50).
- `Pixel_Mark2` – pixel coordinates ( $x, y$ ) of the streak’s end point in the image frame – for example, (487.50, 709.00).
- `Central_Pixel` – pixel coordinates ( $x, y$ ) of the streak’s centre point – for example, (286, 689).
- `Angle` – orientation of the streak with respect to the image  $Y$ -axis, measured from  $0^\circ$  to  $180^\circ$  (e.g.  $98.5^\circ$ ).
- `Magnitude` – measured brightness of the streak. In the present implementation, no magnitude is computed. The images were acquired without a physical filter and no photometric transformation was applied. The magnitude field in the MPC line is therefore filled with a placeholder value (e.g. ‘XX.X R’) and is not used in the analysis. The designation ‘R’ is included only as a conventional label, since the unfiltered CCD response tends to be closer to the red part of the spectrum.
- `MPC Line` – standard MPC-formatted observation record to be reported, containing the object designation, observation date and time in UTC, right ascension (RA) and declination (Dec) of the streak, magnitude, and observatory code. This format allows for uniform reporting of minor planet and small-body observations to the MPC (e.g. ABC0021 C2019 04 12.04618 ...).

While the S/N is a fundamental parameter for assessing detectability, it was not explicitly logged for this first iteration of our code. Instead, we rely on the magnitude as the primary metric. This choice is justified by the uniformity of the observational data. Given the consistent imaging conditions, parameters and background levels across the dataset, the magnitude serves as an effective proxy for the relative visibility of the streaks. The development of a specific module to calculate and integrate explicit S/N values is ongoing work for future iterations of the pipeline (see Section 5.2).

Following the manual labelling process, it was essential to transform the extracted measurements into a format suitable for the training object detection machine learning model. Although YOLO and similar frameworks support a variety of standard image formats (e.g. PNG, JPEG, BMP), they do not natively accept FITS files, nor do they operate on astronomical coordinate systems. Therefore, it was necessary to convert the FITS images to a compatible format and translate the astrometric measurements into pixel-based annotations suitable for model training.

Consequently, a series of pre-processing steps was devised to convert the manually annotated data into structured inputs for the detection pipeline. This included transforming the FITS images into PNG format for compatibility with common computer vision tools and computing precise OBBs around each identified streak. These OBBs ensured that both the spatial extent

and the orientation of each streak were accurately preserved in the dataset.

The conversion from FITS to PNG is performed preserving the native orientation of FITS files. This ensures that the visual representation of each PNG matches the original astronomical frame, where the origin is located at the bottom-left and the vertical axis increases upward. Nonetheless, the saved PNG image adheres to the standard raster convention, with the origin at the top-left and the vertical axis increasing downward.

To maintain consistency with this coordinate system, a vertical flip is applied to the  $y$ -coordinates when loading PNG images for annotation. This adjustment ensures that all geometric computations – such as rotation, scaling, and vertex ordering – are carried out in the reference frame expected by object detection frameworks and image rendering tools.

Special consideration for image sources. For real astronomical images, the conversion from FITS to PNG is performed placing the first row of the array at the bottom of the PNG. Consequently, when generating OBBs from FITS-based coordinates (originally referenced with the origin at the top-left), a vertical correction is applied as

$$y_{\text{corr}} = H_{\text{img}} - y_{\text{original}},$$

where  $H_{\text{img}}$  denotes the height of the image in pixels, and  $y_{\text{original}}$  corresponds to the original FITS-based vertical coordinate. For boosted images, i.e. synthetically generated images introduced for dataset augmentation (hereafter referred to as the synthetic set, also denoted as the boosted set), the pre-processing pipeline directly places the first row at the bottom. Since this vertical inversion is already performed at this stage, the OBB generation step uses

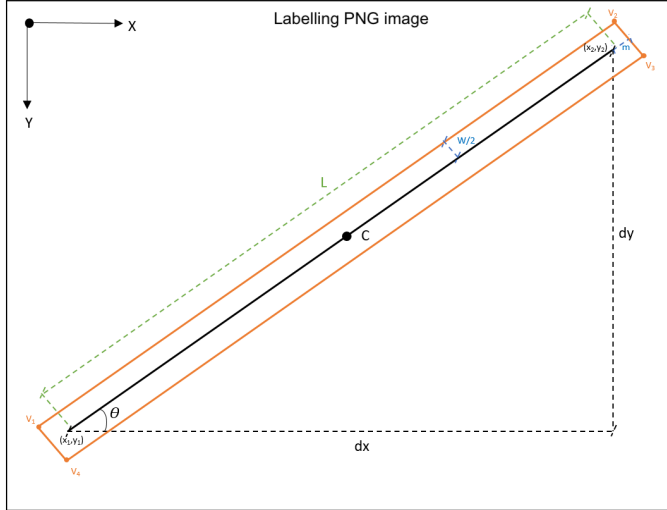
$$y_{\text{corr}} = y_{\text{original}},$$

thus avoiding a double flip and ensuring consistent spatial alignment between the image and its annotations.

Throughout the conversion process, the pixel grid remains unaltered, and a ZScale normalisation is applied using the Astropy implementation (Astropy Collaboration 2013), based on the original algorithm from the IRAF data reduction system (Tody 1986). This algorithm enhances the contrast of faint structures by determining the display limits ( $z_1, z_2$ ) through a linear fit to the central pixels of the sorted intensity distribution, thereby excluding extreme outliers. Consequently, all coordinates remain directly comparable between FITS and PNG formats, facilitating accurate streak localisation and annotation.

Bounding boxes for each detected streak are constructed using the pixel coordinates of its two endpoints, denoted as `Pixel_Mark1` ( $x_1, y_1$ ) and `Pixel_Mark2` ( $x_2, y_2$ ). The following procedure defines the OBB associated with each streak.

From the two measured streak endpoints, the displacement components  $dx$  and  $dy$  define the streak length  $L$  and its orientation  $\theta$  in the image plane. The midpoint of the streak defines the OBB centre  $C$ . A fixed margin  $m$  is applied along the streak axis to extend the box longitudinally and provide contextual background, while a constant width  $W$  defines its transverse extent. The four OBB corners  $V_k$  are then constructed in a local coordinate system aligned with the streak axis and centred at  $C$ , and subsequently expressed in the image reference frame. When required, the box is clipped to ensure that all vertices remain within the image boundaries, and the final vertex ordering is enforced to match the format expected by YOLO-based detection frameworks in the PNG reference frame. A complete mathematical description of this procedure, including clipping and vertex



**Fig. 2.** Schematic illustration of the key parameters involved in the bounding box definition.

ordering, is provided in Appendix A. The geometric parameters involved in this construction are schematically illustrated in Fig. 2, whereas Fig. 1 displays a real example from the dataset.

The resulting OBB annotations are thus fully defined in the PNG image reference frame and ready for use in the detection pipeline. These annotated data constitute the basis for the dataset partitioning strategy described in the following subsection.

## 2.6. Dataset stratification

This subsection describes the stratification strategy adopted for dataset partitioning and training, ensuring that the training, validation, and test subsets are representative of the same streak-length distribution, both for the initial split of the real images and for the final configuration after synthetic data augmentation.

Before implementing any stratification strategy, we initially trained the model using randomly sampled data. The results obtained from this preliminary setup were acceptable: the model performed as expected in validation and test sets, particularly for streaks that shared characteristics similar to those in the training set. However, this random sampling approach introduced a key limitation, the inability to control the distribution of streak lengths across the training, validation, and test subsets. This posed a risk of bias, especially if long or short streaks were underrepresented in any split.

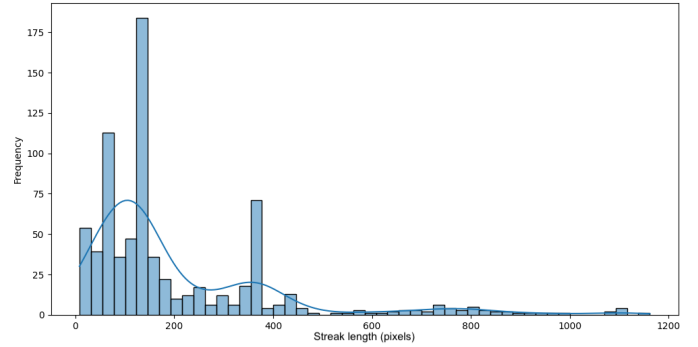
To support the stratified sampling strategy used later in the training pipeline, we first analysed the distribution of individual streak lengths. As previously described in Section 2.3, the dataset contains a total of 765 real streaks, with lengths covering a broad dynamic range. For convenience, Table 2 summarises the main descriptive statistics and percentiles of this distribution.

The 75th percentile (269.1 pixels) was adopted as the operational threshold separating short and long streaks, a choice motivated by the clear skew of the distribution (Figure 3). This threshold provides a practical balance between representativeness and class separability, enabling a controlled stratification of the dataset for training, validation, and test splits.

Such stratification is particularly relevant to mitigate potential overfitting to specific streak lengths and to enhance the generalisation capability of the model under varying observational conditions. The full distribution of streak lengths in the real dataset is illustrated in Figure 3, where a clear skew towards

**Table 2.** Length percentiles for all streaks in the dataset.

| Metric                | Value    |
|-----------------------|----------|
| 25th percentile (P25) | 75.2 px  |
| 50th percentile (P50) | 127.4 px |
| 75th percentile (P75) | 269.1 px |
| 85th percentile (P85) | 362.4 px |
| 90th percentile (P90) | 405.7 px |
| 95th percentile (P95) | 716.6 px |



**Fig. 3.** Histogram of the individual streak lengths across the dataset.

shorter streaks is visible, with a long tail representing a minority of very long detections.

After selecting the 75th percentile as the threshold for streak length, we proceeded to classify the images in the dataset into three categories based on the average length of the streaks they contain. Specifically, we identified 412 images classified as short-streak images, 120 as long-streak images, and 1523 images without streaks. For images containing both short and long streaks, we computed the mean streak length and applied the threshold criterion accordingly: if the average streak length exceeded 269.1 pixels, the image was classified as containing long streaks; otherwise, it was considered a short-streak image. It is important to note that the detection algorithm is able to detect multiple streaks in a single image; the use of the mean streak length per image is only introduced here as a convenient summary statistic for the stratification procedure and does not reflect any limitation on the number of streaks that can be detected.

To ensure a balanced representation of the three image-level classes across all dataset partitions, we applied a stratified splitting strategy using the image classification described above. The complete dataset was divided into training, validation and test subsets with proportions of 70%, 20%, and 10%, respectively.

The real image dataset split consists of 1438 images in the training set, 411 images in the validation set, and 206 images in the test set. Each subset preserves the original class distribution of short-streak, long-streak, and no-streak images, as ensured by the stratified splitting strategy. This balanced distribution across splits is essential to prevent training bias and to obtain reliable and consistent evaluation metrics during model validation and testing, as summarised in Table 3.

Although the stratification procedure described above ensured that each subset preserved the original proportions of short-streak, long-streak, and no-streak images contained in the real dataset, the resulting distribution remained imbalanced with respect to the relative frequency of long streak versus short streak images. Specifically, short-streak images account for roughly

**Table 3.** Number of images and class distribution in each subset of the dataset of real images.

| Subset     | Class     | Count (%)    |
|------------|-----------|--------------|
| Training   | Long      | 84 (5.8%)    |
|            | Short     | 288 (20.0%)  |
|            | No-streak | 1066 (74.2%) |
| Validation | Long      | 24 (5.8%)    |
|            | Short     | 83 (20.2%)   |
|            | No-streak | 304 (74.0%)  |
| Test       | Long      | 12 (5.8%)    |
|            | Short     | 41 (19.9%)   |
|            | No-streak | 153 (74.3%)  |

**Notes.** Percentages are relative to the corresponding subset.

**Table 4.** Number of images and class distribution in each subset of the final (real plus synthetic) dataset.

| Subset     | Class     | Count (%)    |
|------------|-----------|--------------|
| Training   | Long      | 262 (16.0%)  |
|            | Short     | 306 (18.7%)  |
|            | No-streak | 1066 (65.2%) |
| Validation | Long      | 75 (16.1%)   |
|            | Short     | 88 (18.8%)   |
|            | No-streak | 304 (65.1%)  |
| Test       | Long      | 37 (15.8%)   |
|            | Short     | 44 (18.8%)   |
|            | No-streak | 153 (65.4%)  |

**Notes.** The final dataset includes both real and synthetic images. Percentages are relative to the corresponding subset.

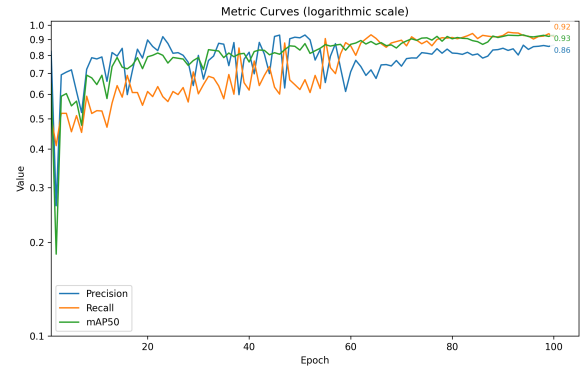
~20% of each subset, whereas long-streak images constitute only ~6%. This imbalance may hinder the model’s ability to learn robust features for underrepresented streak types, particularly long streaks and thus potentially limit its generalisation capability.

To mitigate this limitation, synthetic data generation was applied as described in Section 2.4. The resulting augmented dataset (real plus synthetic) achieved a more balanced representation of short-streak and long-streak images, while slightly reducing the proportion of no-streak images. The final distribution of the combined dataset is summarised in Table 4.

This final stratified and augmented dataset constitutes the input used for training, validation, and testing of the detection model, whose performance is analysed in the following section.

### 3. Results and integration into the detection database

The model’s performance was first evaluated under controlled conditions using the held-out test set, as described in Section 3.1. This evaluation, based on standard metrics such as precision, recall, and mean Average Precision (mAP50), provided a quantitative assessment of detection accuracy, complemented by qualitative visual inspection to verify practical effectiveness in detecting streaks under astronomical conditions.

**Fig. 4.** Evolution of performance metrics – precision, recall, and mAP50 – throughout the 100 training epochs. The values are plotted on a logarithmic scale to emphasise performance dynamics over time.

Beyond the controlled evaluation on the test set, the model was applied to real observational data, where additional artefacts and observational effects (e.g. diffraction spikes or saturated stars) required dedicated post-processing and database-driven filtering.

As a preliminary step, inference was run on a set of 273 images using the trained model with an input resolution of 640 px, a confidence threshold<sup>1</sup> of 0.25, and an IoU threshold of 0.45 for box matching. The complete workflow comprises: (i) bright-star filtering and extraction of temporal metadata (Section 3.2), (ii) photometric extension of OBBs to recover the full streak length (Section 3.3), (iii) estimation of streak endpoints from the refined corner sets (Section 3.4), (iv) geometric extrapolation and inter-frame association of detections belonging to the same physical object (Section 3.5), and (v) record standardisation and assessment of streak completeness (Section 3.6).

#### 3.1. Performance metrics

To assess the performance of the trained model, we performed both a quantitative evaluation on the test set and a qualitative visual inspection of selected predictions. This dual approach allowed us to not only measure the model’s accuracy using standard metrics but also verify its practical effectiveness in detecting streaks under real astronomical conditions.

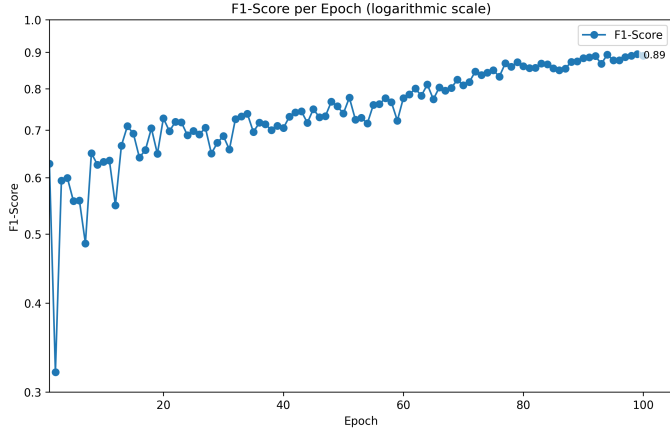
Model performance was evaluated using standard metrics such as precision<sup>2</sup>, recall<sup>3</sup>, and mean Average Precision at Intersection over Union (IoU<sup>4</sup>) 0.5 (mAP50) (Goodfellow et al. 2016; Everingham et al. 2010). Figure 4 shows the evolution of these three metrics across the training epochs.

<sup>1</sup> Minimum confidence required for a predicted object to be retained; lower values increase sensitivity but may introduce false positives.

<sup>2</sup> Precision is defined as  $TP / (TP + FP)$ , where TP denotes true positives and FP denotes false positives. It reflects the proportion of correct positive predictions among all positive predictions made by the model.

<sup>3</sup> Recall is defined as  $TP / (TP + FN)$ , where TP denotes true positives and FN denotes false negatives. It reflects the proportion of actual positives that were correctly identified by the model.

<sup>4</sup> The Intersection over Union (IoU) measures the overlap between the predicted bounding box and the ground truth box. It is defined as the area of their intersection divided by the area of their union, and it is commonly used to determine whether a detection is considered correct.



**Fig. 5.** Evolution of the F1-score per training epoch using a logarithmic scale on the vertical axis. The curve highlights the overall classification performance of the model across epochs.

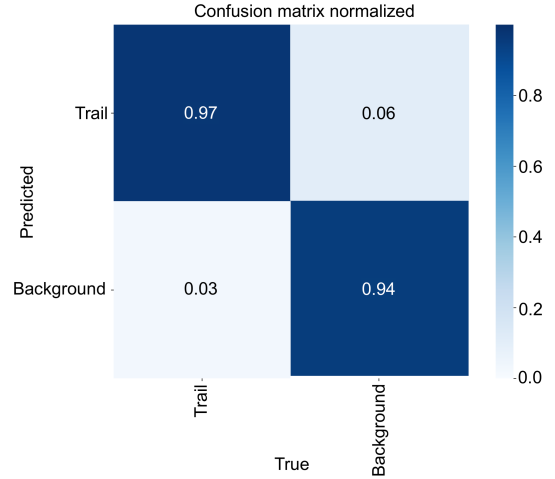
In addition to standard performance metrics, the F1-score<sup>5</sup> (Goodfellow et al. 2016) was computed to provide a balanced measure of classification performance. This metric combines both precision and recall into a single value, allowing for a more comprehensive evaluation of detection quality. Its progression across training epochs is illustrated in Figure 5.

In order to assess the classification behaviour of the trained model, a normalised confusion matrix was computed on the test set using a fixed IoU threshold of 0.8 to determine detection matches. The test set consists of held-out images strictly excluded from the training and validation process, serving as an unbiased benchmark to evaluate the model’s generalisation capabilities. This matrix summarises how the predictions are distributed across the two binary classes (streak/no streak). The left column is expressed in number of individual trails, while the right column refers to number of images, because background frames do not contain bounding-box annotations and must therefore be evaluated per image.

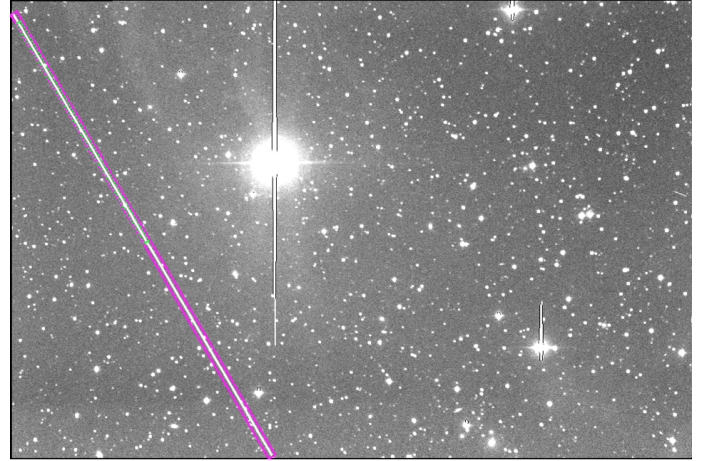
This evaluation uses a hybrid definition reflecting the nature of the available labels. On the trail side (left column), each physical streak is counted as detected if at least one predicted OBB overlaps the ground-truth annotation with  $\text{IoU} \geq 0.8$ . Under this criterion, the model correctly identifies 107 of 110 real streaks (TP = 107, FN = 3).

On the image side (right column), false positives are counted per image: a background frame is considered a false positive if the model produces a streak in an image where no real streak is present. All false-positive detections were verified by visual inspection, and in every case the spurious prediction appears in a different background frame. This one-to-one relation between false positives and images enables a consistent estimate of the number of true-negative frames in the test set.

Taking this into account, and in order to provide a fully populated mixed-unit confusion matrix for reference, the number of true-negative images is obtained directly from the composition of the test set. The background subset contains 153 images, and every false positive appears in a different background frame. Therefore, subtracting the six false-positive images yields a total of 147 true-negative images (Figure 6). Using the strict



**Fig. 6.** Normalised confusion matrix evaluated on the test set using an IoU threshold of 0.8. The left column is expressed in trails, the right column in images, as background frames lack bounding-box annotations and were evaluated per image.



**Fig. 7.** Example of the photometric elongation procedure applied to a single detection. The original OBB returned by the model is shown in green, while the photometrically extended OBB is shown in magenta.

IoU-based definition as the benchmark, the model achieves a precision of 94% and a recall of 97% on the test set. These results demonstrate that the detection model achieves high accuracy under controlled evaluation conditions.

### 3.2. Pre-processing: Stellar filtering and temporal metadata extraction

Under realistic observing conditions, and particularly in frames containing very bright stars, the baseline detection model occasionally misclassified horizontal stellar diffraction spikes as streaks, whereas vertical spikes were largely unaffected by this behaviour. An example is shown in Fig. 7, where a high-confidence prediction would be erroneously assigned to a horizontal spike emerging from a saturated star. To suppress these false positives, a catalogue-driven filtering stage was implemented.

A preliminary survey of the images was used to determine the magnitude threshold above which saturated stars produce detectable diffraction spikes. Based on this, a catalogue-driven filtering procedure was implemented: for each

<sup>5</sup> The F1-score is defined as  $2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$ . It represents the harmonic mean of precision and recall, and penalises models that disproportionately favour one over the other.

image zone, a cone search against the Gaia DR3 catalogue (Gaia Collaboration 2023) is performed via ADQL queries, retrieving all stars brighter than the adopted threshold within a padded region. The resulting stellar catalogue is then cross-matched with the candidate detections using the `search_around_sky` function from Astropy (Astropy Collaboration 2013). Any predicted box located within a predefined angular distance of such stars is flagged as contaminated and removed from further processing. Catalogue queries are parallelised across zones and the filtering is performed in a fully vectorised manner, ensuring that nearly all spike-induced false positives are eliminated while genuine streaks are preserved.

After applying the bright-star filter, the number of spurious detections is substantially reduced. Out of the 66 spike-induced false positives initially present in the inference set (129 detections in total), the filter removes 51, leaving only 15 residual cases. This corresponds to a filter efficiency of 77%, which is sufficient to prevent these artefacts from significantly impacting the subsequent processing stages.

In addition to suppressing spike-induced false positives, this pre-processing stage also extracts the temporal metadata required for subsequent inter-frame analysis. FITS headers are parsed for all observing sequences to extract instrumental and temporal metadata. From these headers, three products are constructed: the image dimensions, the exposure time, and the table of inter-frame intervals computed from successive entries<sup>6</sup>. These intervals provide the temporal baseline required for the endpoint-propagation analysis discussed in Sect. 3.5.

Spike-induced false positives are largely suppressed during the filtering stage. The remaining detections are then geometrically refined to improve the representation of genuine streaks before inter-frame association.

### 3.3. OBB longitudinal extension via photometric pre-analysis

With the spurious detections removed, the next step focuses on refining the geometry of the remaining streak candidates prior to inter-frame association. The OBBs returned by the model frequently underestimate the true extent of linear streaks, as YOLO-based regressors are not optimised for objects that span a significant fraction of the field of view. To address this limitation, an additional post-processing stage performs a photometric pre-analysis along the OBB's major axis and extends the box longitudinally while ensuring geometric consistency.

For each selected detection, the corresponding PNG frame is loaded and transformed into a photometrically enhanced image via greyscale conversion, Contrast Limited Adaptive Histogram Equalization (CLAHE), Gaussian smoothing, and normalisation. This pre-processing increases the contrast of faint streak wings and stabilises the subsequent background estimation.

From the original OBB, the main direction of the trail, its transverse direction, its width, and its two endpoints are recovered. A one-dimensional flux profile  $I(s)$  is then sampled along the axis using bilinear interpolation. Instead of measuring a single line, the profile is computed as the mean over a small bundle of parallel strips spanning the transverse width of the streak, which improves robustness against pixel noise, PSF asymmetries, and local defects.

<sup>6</sup> These intervals represent the elapsed time between the start of consecutive exposures and include both the exposure time and the inter-exposure idle time.

The background level  $I_{\text{bg}}$  and the noise  $\sigma$  are estimated from two sidebands parallel to the streak, displaced outwards by a fixed margin and sampled across a transverse band to obtain a representative distribution of background intensities. The elongation proceeds while the measured flux remains above a dynamic threshold,

$$I(s) > I_{\text{thr}} = I_{\text{bg}} + k\sigma,$$

where the factor  $k$  is adapted to the local signal-to-noise ratio measured in short inner segments located immediately inside each end of the streak. These segments are evaluated using the same multi-strip approach in order to stabilise the S/N estimate. Low-S/N extremities yield higher values of  $k$ , preventing over-extension driven by isolated fluctuations, while allowing genuinely faint parts of the trail to be recovered.

The endpoints are then advanced iteratively in the main direction, one pixel at a time. The propagation continues as long as the intensity remains above threshold and several safeguards are satisfied: a minimum number of consecutive sub-threshold samples, a relative cut based on a fraction of the profile's peak intensity, and a maximum elongation defined as a multiple of the original box length. A final post-check suppresses noise-driven excursions, and elongations that do not exceed a minimal geometric gain are discarded.

The resulting extended axis defines a new OBB. A deep copy of the detection dictionary is updated accordingly: only the geometric fields of each entry are modified, and a flag is stored indicating whether elongation occurred. An illustrative example of this procedure is shown in Fig. 7, where the original OBB (green) and the photometrically extended OBB (magenta) are displayed for a representative detection. The updated dictionary is then saved and propagated to the subsequent stages of the analysis pipeline.

Following the longitudinal extension of the OBBs, the streak endpoints are determined. These endpoints are used to define the streak centre and to enable subsequent temporal propagation and inter-frame association.

It should be emphasised that this pre-analysis is used exclusively for geometric refinement of the OBBs and for determining reliable streak endpoints. No calibrated photometry of the streaks is performed in this work, and the Gaia catalogue is employed solely for astrometric purposes. Consequently, no total calibrated magnitudes are derived from the streak flux measurements. A dedicated photometric analysis is planned for future developments of StreakMind (see Sect. 5.2).

### 3.4. Endpoint estimation from corner clusters

The model outputs OBBs defined by four corner points. These four vertices, updated after the photometry process described in Sect. 3.3, are partitioned into two groups using agglomerative hierarchical clustering with Ward linkage, where the grouping is based on minimising the increase in within-cluster variance at each merge step (Ward 1963). In this constrained case, the procedure produces exactly two clusters, each containing the pair of adjacent vertices that define one extremity of the streak. The centroid of each cluster provides a stable estimate of the corresponding endpoint. The streak centre is then defined as the midpoint between the two estimated endpoints. These points serve as reference markers for subsequent geometric and temporal analyses. In parallel, the celestial coordinates (RA/Dec) associated with the four corners are averaged within each cluster to derive endpoint astrometry, which is later converted into sexagesimal notation for database integration.

Once the streak endpoints are defined, they provide the geometric reference required to extrapolate the streak position across consecutive frames. The streak centre is retained as a compact descriptor for astrometric and database purposes.

### 3.5. Geometric extrapolation and inter-frame association

The purpose of the inter-frame association stage is to determine whether streaks detected in consecutive images correspond to the same physical object. The procedure accounts for the possibility that a streak may continue in either direction along its original axis in subsequent frames. To this end, two extrapolation hypotheses are considered: one projecting the markers forwards from the trailing endpoint, and another projecting them backwards from the leading endpoint. The process begins with a preliminary boundary check: if both predicted markers fall outside the valid image domain under both extrapolations, the track is regarded as unique and no further association is pursued. In such cases, the corresponding object is interpreted as producing only that single streak within the observed passage.

If this termination condition is not met, the subsequent step predicts the likely marker positions in following frames. For a detection with pixel markers  $\mathbf{m}_1, \mathbf{m}_2 \in \mathbb{R}^2$ , the segment is extrapolated to the image bounds to generate a dense set of candidate pixels along the inferred trail direction. Let  $\Delta\mathbf{m} = \mathbf{m}_2 - \mathbf{m}_1$ , and  $t_{\text{exp}}$  be the exposure time. A pixel-velocity estimate is defined as

$$\mathbf{v} = \frac{\Delta\mathbf{m}}{t_{\text{exp}}} \quad [\text{px s}^{-1}].$$

The temporal offset to a future frame ( $k$ ) is obtained by accumulating the inter-frame intervals described in Section 3.2. Each interval represents the elapsed time between the start of two consecutive exposures, comprising both the exposure duration of the earlier frame and the subsequent idle gap. To ensure that the propagation is anchored at the end of the initial exposure rather than its beginning, we subtract the contribution of the first exposure from the cumulative sum. The resulting offset  $\Delta t_k$  therefore corresponds to the effective time elapsed between the completion of the reference exposure and the start of frame  $k$ . This value provides the temporal baseline for projecting the streak forwards in time. To account for potential orientation inversions, we consider both forwards and backwards hypotheses:

$$\mathbf{m}_1^{(k)} = \mathbf{m}_2 + \mathbf{v} \Delta t_k, \quad \mathbf{m}_2^{(k)} = \mathbf{m}_1^{(k)} + \Delta\mathbf{m},$$

$$\tilde{\mathbf{m}}_1^{(k)} = \mathbf{m}_1 - \mathbf{v} \Delta t_k, \quad \tilde{\mathbf{m}}_2^{(k)} = \tilde{\mathbf{m}}_1^{(k)} - \Delta\mathbf{m}.$$

**Notation.** Superscript notation  $^{(k)}$  denotes predicted marker positions in frame  $k$ , while tildes ( $\tilde{\mathbf{m}}$ ) indicate the alternative backwards hypothesis accounting for possible inversions in the streak orientation. In this context,  $\mathbf{m}_1$  and  $\mathbf{m}_2$  are preserved as symbolic labels for the two streak endpoints, even though their physical roles as entry or exit points may swap under the backwards hypothesis. This convention ensures algebraic consistency in the propagation formulas, while accommodating both orientation scenarios.

For each candidate detection in frame  $k$ , the orientation angle is first compared to that of the reference streak. Consistency is required within an angular tolerance of  $\Delta\theta = 6^\circ$ . Only candidates passing this angular check proceed to the positional verification, where both of their markers must lie within a pixel tolerance of  $\tau = 5$  px from the extrapolated segment. A match is established only if both conditions are satisfied simultaneously, in which

case the detection is associated with the reference streak and the track is extended across frames.

If no candidates meet these criteria, the missed-frame counter<sup>7</sup> is incremented and the procedure advances to the next image. In this work, a threshold of  $\text{max\_miss} = 2$  was adopted, meaning that a streak can remain unmatched for tolerating up to two consecutive frames without a valid match; the track is terminated only if no match is found in the third consecutive frame. If both the forwards and backwards extrapolation hypotheses fall entirely outside the valid image boundaries, the association is terminated without further evaluation, as the object is no longer observable within the field of view.

It is important to note that the association procedure is performed independently for each detected streak: every streak is individually propagated across subsequent frames using its own geometric and temporal predictions. This design ensures that simultaneous objects are correctly disentangled, with each one preserving its unique track identifier. For instance, if object A is first detected in frame  $I_1$  and object B in frame  $I_2$ , both will be independently tracked and correctly re-associated in frame  $I_3$ , where the two streaks coexist. Even in cases where object A is temporarily absent from  $I_2$ , the missed-frame tolerance ensures that its trajectory is consistently re-linked once it reappears. In this way, the procedure prevents cross-contamination between tracks and guarantees that each multi-frame track corresponds to a unique moving object. The outcome of the association stage is subsequently transformed into standardised records and subjected to a completeness assessment before being incorporated into the detection database.

### 3.6. Record standardisation and completeness assessment

A streak is labelled ‘complete’ if both markers lie strictly inside an inner margin of width  $\tau_{\text{edge}}$  from the four image borders, and ‘incomplete’ otherwise. Certain images exhibit alignment-induced dead margins, sometimes exceeding 80 px, due to telescope motion over the course of the night. To mitigate misclassification, a threshold of  $\tau_{\text{edge}} = 40$  px was adopted: values that are too small would mark incomplete streaks near these margins as complete, whereas overly large values would incorrectly flag truly complete streaks as incomplete. The threshold was empirically selected to balance both considerations across observing nights. Formally, with image bounds  $[0, W_{\text{img}}) \times [0, H_{\text{img}})$  and  $\tau_{\text{edge}} = 40$  px,

$$\tau_{\text{edge}} < x_{m_i} < W_{\text{img}} - \tau_{\text{edge}}, \quad \tau_{\text{edge}} < y_{m_i} < H_{\text{img}} - \tau_{\text{edge}}, \quad i \in \{1, 2\}.$$

This completeness status is incorporated into the normalised record together with the astrometric solution. Following the inter-frame association procedure described in Section 3.5, unique track identifiers (`Track_ID`) are assigned sequentially to detections and propagated to associated matches. In addition to per-detection parameters, constant metadata are uniformly attached to all detections belonging to a given observing set, including the observatory name, MPC code, telescope, sensor, field-of-view, observer, and analyst. Central and marker astrometry are stored in sexagesimal format, while pixel coordinates are rounded to two decimals for reproducibility. Image names are

<sup>7</sup> The missed-frame counter allows the association to tolerate up to two consecutive frames without a valid match. This prevents spurious breaks in the track linkage due to transient issues such as clouds, poor seeing, or temporary image degradation, ensuring that detections belonging to the same physical object are not mistakenly split into separate tracks.

normalised to include the observing date (UTC) derived from DATE-OBS.

On this basis, an MPC-style observation line is synthesised using the final Track\_ID, MPC-formatted date, astrometry and site code. The resulting standardised record is appended to the SQL-ready dictionary, ensuring that all detections can be directly ingested into the detection database.

#### 4. Satellite cross-identification and database export

Having established the geometric characterisation, temporal association, and standardised representation of the detected streaks, the final step of the pipeline addresses their physical interpretation. At this stage, each multi-frame detection is evaluated to determine its most likely origin and to quantify the confidence of its identification through cross-identification with external ephemerides and subsequent confidence scoring, prior to database ingestion. The procedure is organised into the following stages:

**Cross-identification against external ephemerides.** Each observation is converted into an MPC-style astrometric line (Section 3.6) and submitted in batch to the public Sat\_ID service of Project Pluto (Project Pluto 2025). The query returns a list of candidate satellites together with their angular offsets from the reported position. A conservative search radius of  $1^\circ$  is adopted in all submissions, ensuring that only plausible matches are returned.

Two independent runs are executed: (i) using exclusively the set of complete streaks and (ii) using all the streaks. This dual approach mitigates the effect of incomplete streaks, whose truncated morphology can displace the centroid used in the ephemeris query, occasionally suppressing otherwise valid identifications. The results of both runs are merged (after filtering explained at the beginning of the next subsection) by retaining, for each MLD identifier and each satellite code, only the occurrence with the smallest angular offset.

**Offset filtering and Gaussian-based confidence scoring.** Before computing the confidence scores, candidates whose offsets are unreasonably large compared with the minimum offset are discarded. A candidate is removed if its offset is more than five times the minimum, or if it exceeds the minimum by more than  $0.4^\circ$ . The surviving offsets are then evaluated through a two-component confidence model governed by two Gaussian scales: a relative scale  $\sigma_{\text{rel}} = 0.2^\circ$  that measures how quickly confidence decreases for offsets deviating from the minimum, and an absolute scale  $\sigma_{\text{abs}} = 0.3^\circ$  that penalises cases in which the minimum offset itself is large.

We let the filtered offsets, sorted in ascending order, be

$$o_0 \leq o_1 \leq o_2 \leq \dots,$$

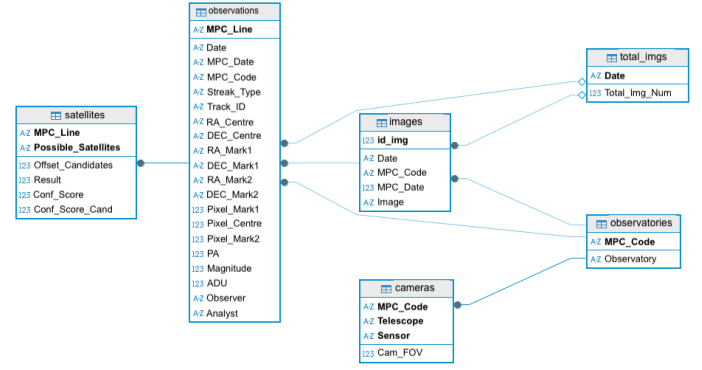
with  $o_{\text{min}} = o_0$ . The first component of the model assigns each offset a relative-distance weight,

$$w_i = \exp\left(-\frac{(o_i - o_{\text{min}})^2}{2\sigma_{\text{rel}}^2}\right),$$

which is normalised to

$$p_i = \frac{w_i}{\sum_j w_j}$$

so that candidates close to the minimum dominate the probability distribution.



**Fig. 8.** Database schema used for storing observation and object cross-identifications.

The second component penalises solutions where the minimum offset itself is large,

$$\text{abs\_factor} = \exp\left(-\frac{\sigma_{\text{min}}^2}{2\sigma_{\text{abs}}^2}\right).$$

The confidence of each candidate is then

$$\text{Conf\_Score\_cand}_i = p_i \cdot \text{abs\_factor},$$

and the global confidence score is defined as

$$\text{Conf\_Score} = p_0 \cdot \text{abs\_factor},$$

where  $p_0$  is the normalised weight associated with the candidate of minimum offset (i.e. the first element in the ordered set). All confidence values lie in  $[0, 1]$ .

**Pass-separation safeguard.** A secondary consistency step is applied after the confidence scoring to prevent overwriting the confidence values of detections belonging to different passes of the same satellite. For each satellite code, detections are examined within a small window of neighbouring MLD identifiers. When multiple candidates for the same satellite appear inside this window, the global confidence (Conf\_Score) assigned to each entry is set to the maximum of the individual per-candidate scores (Conf\_Score\_cand) found in that neighbourhood. This ensures that detections from the same pass retain coherent confidence values, while independent passes remain isolated and are not affected by one another. The result of this process is a unified table where each MLD entry is associated with the angular offset to the candidate satellite, the candidate satellite code, the per-candidate confidence score, and the local global confidence.

**Record enrichment and database export.** The consolidated identification results (offset, satellite code, Conf\_Score, and Conf\_Score\_cand) are appended to the standardised records constructed in Section 3.6. The final dataset is then exported into a lightweight SQLite database comprising five core tables (observatories, cameras, images, observations, and satellites), together with an auxiliary table imgTotales that stores the number of processed images per observing date (Figure 8).

## 5. Discussion

### 5.1. Comparison with manual inspection

A direct comparison between StreakMind and traditional manual inspection highlights several practical advantages of the automated approach. First, the method is inherently scalable: the

inference pipeline processes entire nights of observations in timescales of minutes, or even seconds for subsets of frames, whereas manual examination of the same dataset typically requires several hours of uninterrupted work. This difference becomes particularly relevant for observatories generating large nightly volumes of data, where exhaustive human inspection is no longer feasible.

In addition to its computational efficiency, the system exhibits a strong ability to detect faint streaks. Several of the detections recovered by StreakMind correspond to low S/N features that are difficult to identify through visual inspection alone. The automated processing therefore provides a more uniform sensitivity across the full dataset, independent of observer fatigue or subjective thresholds.

Another practical benefit is the immediate integration of results into the database structure described in Sect. 4. The direct generation of standardised records and satellite cross-identifications removes the need for manual cataloguing tasks and ensures reproducible handling of both detections and astronomical objects identifications, reducing the overhead typically associated with cataloguing and post-processing.

Regarding geometric accuracy, the model performs robustly for streaks spanning up to approximately half of the image width. In this regime, both the detection rate and the OBB shape reconstruction remain highly reliable, making StreakMind a robust alternative to manual inspection. For very long streaks (exceeding ~50% of the image width), the detection remains stable, but the geometric adjustment requires a dedicated photometric-based post-processing step. This additional stage slightly reduces the positional precision compared to the short- and mid-length regime, although the streaks are still consistently identified and classified.

Overall, StreakMind provides a substantial improvement in efficiency, reproducibility, and sensitivity compared to manual inspection. It also maintains competitive geometric performance across most of the streak-length distribution encountered in this dataset.

## 5.2. Perspectives for integration and future work

The results presented in this work demonstrate the feasibility of an end-to-end automated pipeline for the detection, geometric characterisation, association, and identification of streaks produced by artificial satellites and space debris in ground-based astronomical images. All identifications discussed in this paper correspond exclusively to artificial objects, including operational satellites, rocket bodies and space debris, cross-matched against external satellite ephemerides.

Several extensions are currently under development to enhance both the scientific output and the operational usability of the pipeline. A primary forthcoming addition is a dedicated photometric analysis module, which will enable the estimation of the apparent brightness of the objects associated with each detected streak, along with S/N values that will allow for a more rigorous assessment of detection sensitivity. This functionality builds naturally upon the photometric information already exploited for geometric elongation and will allow for the inclusion of brightness measurements as first-class parameters in the detection database. In parallel, a graphical user interface is being developed to facilitate routine use of the pipeline by observers and analysts, reducing the barrier to adoption in operational environments and collaborative campaigns.

On the identification side, the current satellite cross-matching strategy will be extended through additional external

services. In particular, an independent verification layer based on the SatChecker database<sup>8</sup>, which provides orbital elements and metadata for a wide range of satellites, rocket bodies, and debris objects, is planned to further improve robustness and confidence assessment.

Beyond artificial objects, future versions of the pipeline will incorporate dedicated identification pathways for NEOs. Automated queries to the Skybot service (Berthier et al. 2006) and to JPL's Horizons system (Giorgini et al. 1996) will allow for the acquisition of ephemerides for known asteroids predicted to traverse the field of view. This extension will enable the confirmation of faint asteroid streaks that may be present in the images but remain undetected during the initial inference stage, as well as the automatic generation of MPC-compatible astrometric reports for submission to the MPC.

Finally, an important medium-term objective is the deployment of the detection model at additional ground-based observatories and the exploitation of larger archival datasets. Transfer learning offers a promising strategy in this context, allowing the model trained on La Sagra Observatory data to be rapidly adapted to new sites using comparatively small, site-specific datasets. As shown in this work, the performance of such models can be further enhanced through the controlled inclusion of synthetically generated streaks, particularly for underrepresented distributions. In the longer term, training the initial model on a more diverse, multi-observatory dataset may further improve generalisation and portability across instruments and observing conditions.

An important consideration for the broader applicability of the pipeline is its dependence on the specific observational setup, including telescope optics, camera characteristics, pixel scale, and typical seeing conditions. While the current model has been trained and validated exclusively on data from La Sagra Observatory, the underlying architecture is instrument-agnostic: all image pre-processing, OBB detection, and photometric elongation procedures operate on pixel-based representations and do not embed instrument-specific parameters.

Future deployment on other telescopes, including large-aperture instruments, will rely on transfer learning and site-specific fine-tuning. The adaptation strategy will depend on the characteristics of the new instrument, such as field of view, pixel scale, and noise properties. In cases of moderate differences, existing models can be adjusted using relatively small local datasets. If the new data distribution significantly departs from that of the original training set, new models may be trained from scratch. Synthetic streak augmentation, as demonstrated in this work, will further support adaptation by reproducing observational conditions not represented in the training data. Taken together, these approaches ensure that the pipeline can be extended to different observatories and camera configurations, facilitating integration into diverse operational environments while preserving detection performance and reliability.

## 6. Conclusions

We have introduced StreakMind, an AI-based pipeline designed for the automated detection and characterisation of linear streaks in ground-based astronomical images originally acquired for astrometric and photometric purposes. The system integrates a YOLO11 oriented bounding box detector trained on a stratified combination of real and synthetic images together with a geometric framework that converts FITS-based annotations into

<sup>8</sup> <https://satchecker.readthedocs.io/en/latest/>

consistent PNG-space OBB labels. This approach enables robust operation across heterogeneous observing conditions and streak morphologies that span a wide range of lengths and signal-to-noise ratios while maintaining astrometric coherence throughout the pipeline.

Quantitative evaluation on an independent test set showed that the detector achieves high levels of precision and recall under a strict IoU threshold, with most real streaks successfully detected. Performance is particularly reliable for short- and intermediate-length streaks, for which both the detection rate and the geometric reconstruction of the OBBs remain stable. For very long streaks, detection remains robust, while geometric refinement benefits from the dedicated photometric post-processing stage. The bright-star filtering procedure proved effective during the subsequent inference stage by suppressing the diffraction spikes that dominate the false-positive detections. These results are reflected in the final performance metrics on the test set, where the model reached a precision of 94% and a recall of 97% under a strict IoU threshold.

A further advantage of the pipeline lies in the direct integration of the inference results into a normalised detection database. Each detection is transformed into an MPC-style record that is enriched with geometric and temporal information and subsequently cross-identified with external satellite ephemerides using a sequential offset filtering and confidence evaluation process. This produces a structured and reproducible dataset suitable for large-scale statistical analyses of space-object contamination and for the systematic exploitation of archival observations.

Overall, StreakMind represents a first step towards efficient analyses of orbital-object contamination. Moreover, we have provided a foundation for the integration of StreakMind into broader astrometric workflows.

*Acknowledgements.* This work was carried out in collaboration with the Instituto de Astrofísica de Andalucía (IAA-CSIC), which operates the La Sagra Observatory. René Duffard and Nicolas Morales acknowledge financial support from the Severo Ochoa grant CEX2021-001131-S funded by MCIN/AEI/10.13039/501100011033. Google Colab was employed for part of

the model training and experimentation. This work has made use of data from the European Space Agency (ESA) mission *Gaia* (<https://www.cosmos.esa.int/gaia>), processed by the Gaia Data Processing and Analysis Consortium (DPAC, <https://www.cosmos.esa.int/web/gaia/dpac/consortium>). Funding for the DPAC has been provided by national institutions, in particular the institutions participating in the Gaia Multilateral Agreement.

## References

- Astropy Collaboration, (Robitaille, T. P., et al.) 2013, *A&A*, 558, A33
- Berthier, J., Vachier, F., Thuillot, W., et al. 2006, in *Astronomical Society of the Pacific Conference Series*, 351, Astronomical Data Analysis Software and Systems XV, eds. C. Gabriel, C. Arviset, D. Ponz, & S. Enrique, 367
- Bertin, E. 2013, PSFEX: Point Spread Function Extractor, Astrophysics Source Code Library [[record ascl:1301.001](https://ui.adsabs.org/abs/2013ascl.conf..001B)]
- Billot, N., Hellmich, S., Benz, W., et al. 2024, *J. Space Saf. Eng.*, 11, 498
- Botke, W. F., Morbidelli, A., Jedicke, R., et al. 2002, *Icarus*, 156, 399
- Duev, D. A., Mahabal, A., Ye, Q., et al. 2019, *MNRAS*, 486, 4158
- Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., & Zisserman, A. 2010, *Int. J. Comput. Vis.*, 88, 303
- Gaia Collaboration (Vallenari, A., et al.) 2023, *A&A*, 674, A1
- Giorgini, J. D., Yeomans, D. K., Chamberlin, A. B., et al. 1996, in *AAS/Division for Planetary Sciences Meeting Abstracts*, 28, 25.04
- Gonçalves, L. F. P., Billot, N., Hellmich, S., et al. 2026, *Acta Astronaut.*, 239, 61
- Goodfellow, I., Bengio, Y., & Courville, A. 2016, *Deep Learning* (Cambridge, MA: MIT Press), <http://www.deeplearningbook.org>
- Irureta-Goyena, B. Y., Helou, G., Kneib, J.-P., et al. 2025, *PASP*, 137, 054503
- Jocher, G., & Qiu, J. 2024, *Ultralytics YOLO11*
- Khanam, R., & Hussain, M. 2024, arXiv e-prints [arXiv:2410.17725]
- Kruk, S., García Martín, P., Popescu, M., et al. 2022, *A&A*, 661, A85
- Lieu, M., Conversi, L., Altieri, B., & Carry, B. 2019, *MNRAS*, 485, 5831
- Parrott, D. 2020, *J. Am. Assoc. Variable Star Observers*, 48, 262
- Pöntinen, M., Granvik, M., Nucita, A. A., et al. 2023, *A&A*, 679, A135
- Project Pluto. 2025, Sat\_ID: Satellite Identification Service, [https://www.projectpluto.com/sat\\_id.htm](https://www.projectpluto.com/sat_id.htm), accessed: 2025-09-26
- Stark, D. V., Grogin, N., Ryon, J., & Lucas, R. 2022, *Improved Identification of Satellite Trails in ACS/WFC Imaging Using a Modified Radon Transform*, Instrument Science Report ACS 2022-8, 25
- Stănescu, M., Popescu, M. M., Curelaru, L., et al. 2025, *A&A*, 704, A13
- Tody, D. 1986, *SPIE Conf. Ser.*, 627, 733
- Vaduvescu, O., Stănescu, M., Popescu, M., et al. 2025, *New A*, 119, 102410
- Ward, J. H. 1963, *J. Am. Statist. Assoc.*, 58, 236
- Xia, G.-S., Bai, X., Ding, J., et al. 2018, *IEEE Trans. Pattern Anal. Mach. Intell.*, 41, 1
- Zhai, C., Shao, M., Saini, N., et al. 2024, *PASP*, 136, 034401

## Appendix A: OBB construction and clipping procedure

We first computed the directional vector and its length:

$$\mathbf{d} = (x_2 - x_1, y_2 - y_1), \quad L = \|\mathbf{d}\| = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}.$$

To provide contextual background, the streak was extended by a fixed margin  $m = 4$  px along its axis, yielding a nominal half-length:

$$\frac{L'}{2} = \frac{L + 2m}{2} = \frac{L}{2} + m.$$

The OBB is defined with a constant width  $W = 16$  px, corresponding to a symmetric expansion of  $\pm W/2$  pixels along the direction orthogonal to the streak.

We define a local coordinate system  $(d_\ell, d_w)$ , where  $d_\ell$  is aligned with the streak axis (longitudinal component), and  $d_w$  is perpendicular (width direction). The initial (unrotated) corner positions in this frame are

$$V_{\text{local}} = \left\{ \left(-\frac{L'}{2}, -\frac{W}{2}\right), \left(\frac{L'}{2}, -\frac{W}{2}\right), \left(\frac{L'}{2}, \frac{W}{2}\right), \left(-\frac{L'}{2}, \frac{W}{2}\right) \right\}.$$

The vertex sequence defined in  $V_{\text{local}}$  follows a counter-clockwise order starting from the bottom-left corner in the local frame. However, after rotating and translating these coordinates into the PNG image reference system (which includes a vertical flip), the vertices are explicitly reordered to match the clockwise format expected by object detection frameworks. This reordering does not alter the geometry of the box and is described in detail below.

The angle of the streak was computed as

$$\theta = \text{atan2}(y_2 - y_1, x_2 - x_1),$$

and the centre of the OBB (after margin extension) is

$$C_x = \frac{x_1 + x_2}{2} + m \cos \theta, \quad C_y = \frac{y_1 + y_2}{2} + m \sin \theta.$$

Before rotation, we applied a symmetric clamping to the half-width of the OBB to ensure it remains within vertical bounds:

$$\frac{W_{\text{clamped}}}{2} = \min\left(\frac{W}{2}, C_y, H_{\text{img}} - C_y\right).$$

The clamping is performed before the longitudinal adjustment to guarantee that, after axis-aligned scaling is applied, the vertical extent of the box remains fully contained within the image bounds even in cases where the centre lies close to the edges.

Each of the four vertices in  $V_{\text{local}}$  is then rotated by the angle  $\theta$  and translated to the centre  $(C_x, C_y)$ , yielding their global coordinates in the PNG reference frame:

$$\begin{pmatrix} x_g^{(k)} \\ y_g^{(k)} \end{pmatrix} = \begin{pmatrix} C_x \\ C_y \end{pmatrix} + \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} d_\ell^{(k)} \\ d_w^{(k)} \end{pmatrix}, \quad k = 1, \dots, 4.$$

To keep all box vertices within the image frame along the streak direction,

$$x_g \in [0, W_{\text{img}}), \quad y_g \in [0, H_{\text{img}}),$$

we evaluated each vertex and computed a positive scaling factor  $f$  (applied to  $d_\ell$ ) whenever a corner lies out of bounds:

$$\text{if } x_g < 0: \quad f = \frac{-C_x + \sin \theta d_w}{\cos \theta d_\ell}, \quad (\text{A.1})$$

$$\text{if } x_g > W_{\text{img}}: \quad f = \frac{W_{\text{img}} - C_x + \sin \theta d_w}{\cos \theta d_\ell}, \quad (\text{A.2})$$

$$\text{if } y_g < 0: \quad f = \frac{-C_y - \cos \theta d_w}{\sin \theta d_\ell}, \quad (\text{A.3})$$

$$\text{if } y_g > H_{\text{img}}: \quad f = \frac{H_{\text{img}} - C_y - \cos \theta d_w}{\sin \theta d_\ell}. \quad (\text{A.4})$$

The final scaling factors for the two halves of the box are defined as

$$f_{\text{back}} = \min(f \text{ for all corners with } d_\ell < 0),$$

$$f_{\text{forward}} = \min(f \text{ for all corners with } d_\ell > 0),$$

with both saturated to the range  $[0, 1]$ .

We then computed the corrected half-lengths:

$$\ell_{\text{back}} = \frac{L'}{2} \cdot f_{\text{back}}, \quad \ell_{\text{forward}} = \frac{L'}{2} \cdot f_{\text{forward}},$$

and we redefined the OBB corners accordingly,

$$V_{\text{local}}^{\text{clipped}} = \left\{ \left(-\ell_{\text{back}}, -\frac{W_{\text{clamped}}}{2}\right), \left(\ell_{\text{forward}}, -\frac{W_{\text{clamped}}}{2}\right), \left(\ell_{\text{forward}}, \frac{W_{\text{clamped}}}{2}\right), \left(-\ell_{\text{back}}, \frac{W_{\text{clamped}}}{2}\right) \right\}. \quad (\text{A.5})$$

This asymmetric scaling ensures that if one end of the box exceeds the image boundaries, only that portion is trimmed, preserving the remainder of the annotation and maintaining geometrical consistency.

The order of vertices in  $V_{\text{local}}^{\text{clipped}}$  follows the same convention as in  $V_{\text{local}}$ , preserving the counter-clockwise sequence in the local frame. However, this order is later adjusted once the corners are projected into global coordinates, using the same reordering strategy previously described to ensure consistency in the PNG reference frame.

**Vertex ordering for annotation.** The final ordering of the OBB corners is explicitly enforced to match the format required by YOLO-based detection frameworks. This format expects a clockwise sequence of vertices starting from the top-left corner of the image.

To ensure this, the four projected vertices are analysed in the PNG coordinate system (with top-left origin and vertical flip). The procedure is as follows:

- Select the two vertices with the smallest  $x$  coordinates (i.e. the leftmost).
  - Among these vertices, the one with the smallest  $y$  is labelled as ‘top left’ (TL), and the other is labelled as ‘bottom left’ (BL).
  - The two remaining vertices (with largest  $x$  values) are labelled analogously. The smallest  $y$  becomes ‘top right’ (TR), and the other becomes ‘bottom right’ (BR).
- This process produces the final ordered set:

$$V_{\text{ordered}} = \{\text{TL}, \text{TR}, \text{BR}, \text{BL}\}.$$