SPARK 2022 Project Report - Epsilon

QUINET Maxence
Computer Vision Course
University Of Luxembourg
Belval, Luxembourg
maxence.qnt@outlook.fr

SHARMA Naman
Computer Vision Course
University Of Luxembourg
Belval, Luxembourg
namansharma2621@gmail.com

MOGAL Faraz
Computer Vision Course
University Of Luxembourg
Belval, Luxembourg
faraz.mogal@gmail.com

NUNCIBELLO Andrea
Computer Vision Course
University Of Luxembourg
Belval, Luxembourg
andrea.nuncibello.001@student.uni.lu

***Abstract*—This report presents the results of the Epsilon group's object detection project for the University of Luxembourg using YOLO (You Look Only Once) on the SPARK 2022 dataset. We describe the dataset, the classes, the model choices, and the results obtained (mAP, accuracy, etc.), while illustrating the effectiveness of YOLO for this task for object detection tasks, such as the SPARK 2022 dataset.**

I. General Idea of the project

The aim of this project is to exploit advanced object detection techniques to analyse the SPARK 2022 dataset. Our group, Epsilon, chose the YOLO (You Only Look Once) algorithm because it is known for its high efficiency and accuracy in real-time object detection tasks. We concluded with the group that we wanted to offer a solution that could be applied in real life. Analysing satellites travelling at very high speed around the Earth requires rapid analysis. Choosing 'two stage detector' algorithms like Faster R-CNN was therefore not applicable for our chioix. One-stage detector algorithms such as YOLO were not. The objective was to detect and classify objects in the different classes present in the dataset, focusing on optimising detection performance measures such as mAP (Mean Average Precision), and accuracy. Through this work, we aimed to demonstrate the potential of YOLO for complex object detection tasks and understand its strengths when applied to real-world datasets such as SPARK 2022.

II. Introduction of the Dataset & Classes

A. General Overview

The dataset used for this project is the SPARK 2022 dataset, which consists of over 100,000 annotated images capturing various objects in Earth's orbit, including satellites and space debris. The dataset is structured into three main subsets: the training set contains 60,000 images, while the validation and test sets each comprise 22,000 images.

The dataset includes 11 distinct classes, representing specific satellite models and other orbital objects: *smart_1*, *cheops*, *lisa_pathfinder*, *debris*, *proba_3_csc*, *proba_3_ocs*, *soho*, *earth_observation_sat_1*, *proba_2*, *xmm_newton*, and *double_star*. Additionally, some images are classified as "no detection" when no object of interest is present, effectively serving as background images.

Identify applicable funding agency here. If none, delete this.

Each image is accompanied by corresponding annotations stored in CSV files for the training and validation datasets. These CSV files include the image filename, the class of the object present, and the bounding box coordinates for object localization.

B. Dataset Format Problem

During the dataset preparation process, a formatting issue was identified. While the images themselves were stored in JPEG format (e.g., `imgxxxxxx.jpg`), the filenames in the CSV files referenced them using the PNG format (e.g., `imgxxxxxx.png`). This inconsistency required a preprocessing step to standardize the filenames in the CSV files by converting all references to the correct JPEG format. Addressing this issue was crucial to ensuring seamless data loading and annotation alignment during model training.

III. How YOLO works & Various Versions

A. YOLO

YOLO (You Only Look Once) is a deep learning method for object detection, known for its speed and efficiency. Unlike other detection models that divide an image into several parts or go through several stages, YOLO processes the entire image in a single step, making it a one-stage detector algorithm. The fundamental principle of YOLO is based on dividing an image into a grid and predicting, for each cell of the grid, the potential bounding boxes and their probabilities associated with the object classes. This enables fast and accurate predictions to be made, even on complex images.

YOLO works in three main stages:

1) The input image is resized and divided into a grid (e.g., 13x13).
2) Each grid cell predicts multiple bounding boxes with confidence scores, as well as the probability that an object belongs to a specific class.
3) Redundant or low-confidence bounding boxes are eliminated using a Non-Maximum Suppression (NMS) algorithm.

B. Various versions of YOLO

Over the years, several YOLO versions have been developed to enhance its performance:

- **YOLOv1** (2016): The first version introduced the concept of single-stage object detection. While effective, it struggled with small object detection.
- **YOLOv2 and YOLOv3**: These versions introduced anchor boxes to better handle objects of varying sizes and deeper network architectures for improved accuracy.
- **YOLOv4**: Integrated new optimization techniques, such as data augmentation and advanced loss functions, to achieve better accuracy while maintaining high speed.
- **YOLOv5**: Although unofficial and developed by a separate community, it gained popularity due to its ease of use and strong performance.
- **YOLOv6 and YOLOv7**: These versions pushed the boundaries of speed and accuracy by incorporating advanced network optimization techniques.
- **YOLOv8**: The latest official version, notable for its improved compatibility with different frameworks and optimized performance for various tasks.
- **YOLOv9**: An experimental model built on the YOLOv5 codebase, incorporating Programmable Gradient Information (PGI) to improve adaptability and efficiency in complex object detection tasks [1].
- **YOLOv10**: Developed by Tsinghua University, this version eliminates the need for Non-Maximum Suppression (NMS) and introduces an architecture focused on precision and efficiency, achieving high performance with minimal latency [1].
- **YOLOv11**: (The version that we have used for this project) Explores newer innovations, including attention mechanisms, for better detection of small objects and complex contexts.

C. Version Used in This Project

For this project, we chose to use **YOLOv11**, the latest official and stable version of YOLO as of 2025. According to the official documentation [1], YOLOv11 offers several advantages that align well with the objectives of our task:

- **State-of-the-Art Performance:** YOLOv11 is optimized for both speed and accuracy, making it highly suitable for large-scale datasets like SPARK 2022.
- **Ease of Use:** The version is designed to be user-friendly, with streamlined integration for model training, validation, and inference.
- **Compatibility:** YOLOv11 provides better compatibility with various frameworks and is more adaptable for different object detection tasks compared to previous versions.
- **Improved Detection:** YOLOv11 demonstrates better handling of small objects and complex backgrounds, which are prevalent in the SPARK 2022 dataset.
- **Community Support:** As the official version, YOLOv11 benefits from a larger user base and active development, ensuring timely updates and support.

Initially, we tested **YOLOv8**, to evaluate its potential for further improvements. However, our results showed that YOLOv8 produced lower performance metrics, including mAP and accuracy, compared to YOLOv11. This could be attributed to its experimental nature and lack of optimization for certain scenarios, such as handling diverse object classes in SPARK 2022.

Given these observations, we finalized our choice on YOLOv11, which provided a reliable and high-performing solution for our object detection task. This decision highlights the importance of stability and proven efficiency in real-world applications.

IV. Results

Our results demonstrate the robustness of YOLOv11 on the SPARK 2022 dataset. The model achieves an overall accuracy of **86%**, as calculated from the confusion matrix (Fig. 1). Most classes, including *cheops*, *lisa_pathfinder*, and *double_star*, show excellent detection precision, with scores exceeding 95%. However, minor misclassifications are observed in classes such as *background* and *earth_observation_sat_1*.

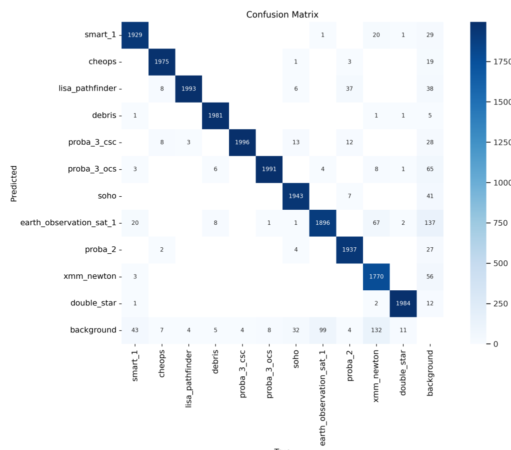

Fig. 1. Confusion matrix showing the classification results for each class.

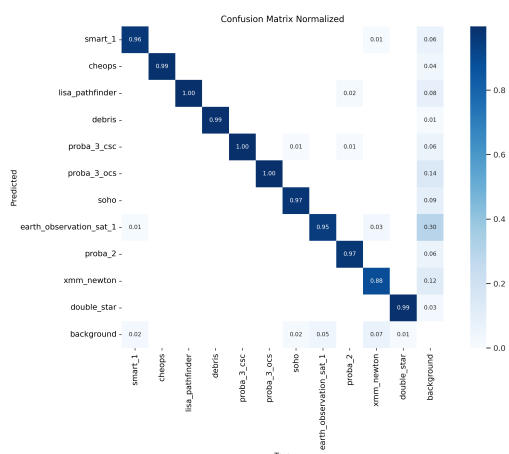

Fig. 2. Normalized confusion matrix highlighting the relative proportions of correct and incorrect predictions.

The **F1-Confidence curve** (Fig. 3) reveals an optimal F1-score of **0.93**, achieved at a confidence threshold of **0.395**, indicating a balanced trade-off between precision and recall. The **Precision-Recall curve** (Fig. 4) further supports this observation, with a mean Average Precision (mAP) of **96.9%** across all classes.

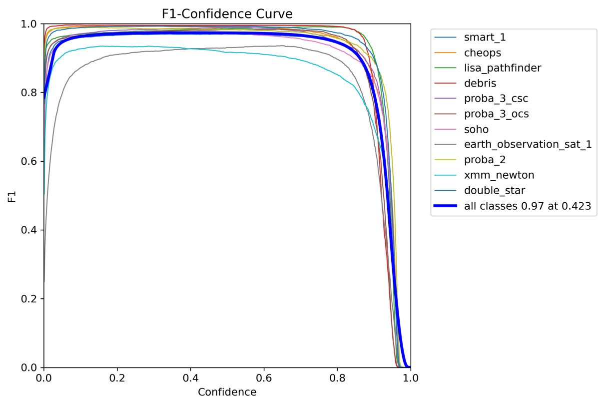

Fig. 3. F1-Confidence curve showing the relationship between F1-score and confidence threshold.

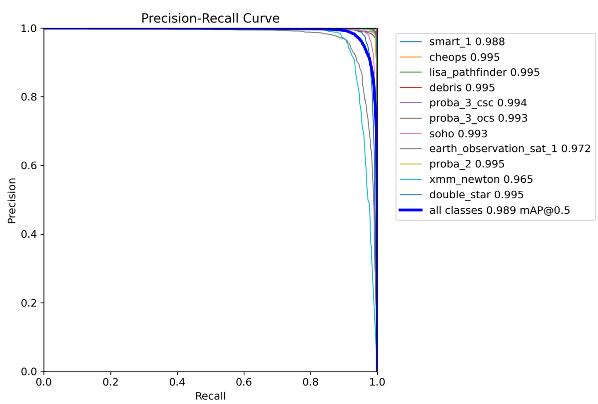

Fig. 4. Precision-Recall curve demonstrating the trade-off between precision and recall for each class.

Additionally, the **training and validation loss curves** (Fig. 5) exhibit a steady decline, highlighting the model's effective learning process.

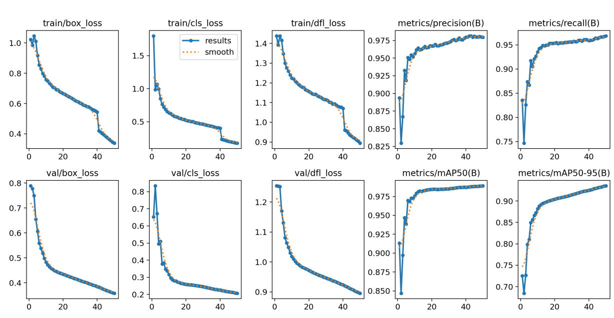

Fig. 5. Training and validation loss curves showing the convergence of the model during training.

Finally, the **label distribution and bounding box analysis** (Fig. 6 and Fig. 7) confirm the diversity of the dataset, which contributes to the model's generalization capabilities.

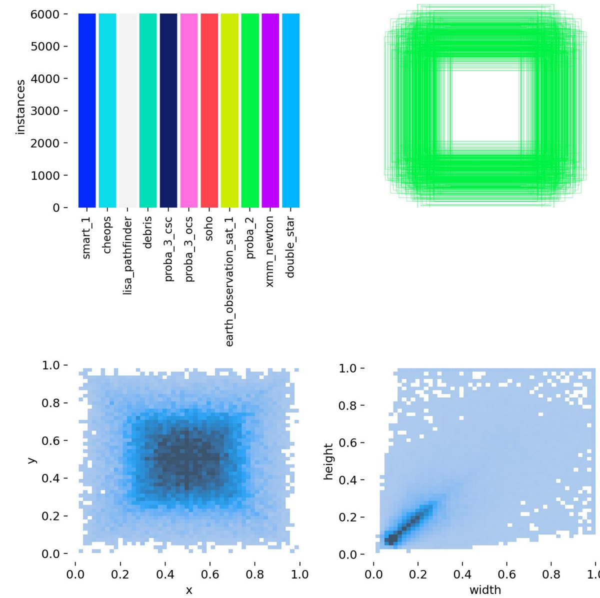

Fig. 6. Label distribution across all classes in the SPARK 2022 dataset.

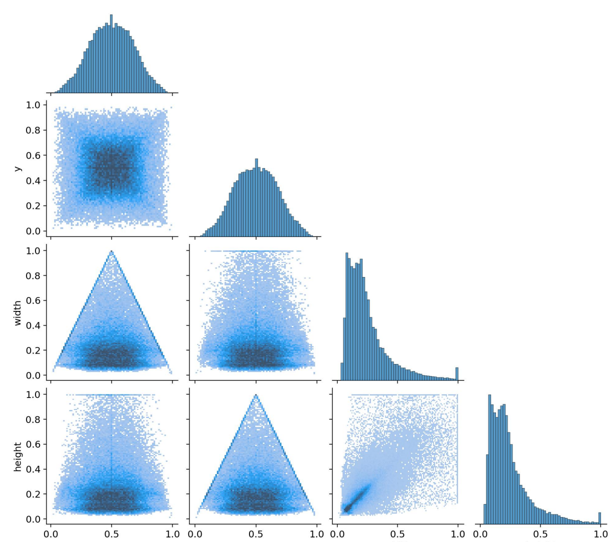

Fig. 7. Bounding box distribution across the dataset, showing correlations between object positions and dimensions.

V. Example Predictions

To showcase the performance of our trained model, we present an example of predictions on the validation dataset. The image below illustrates bounding boxes generated by the model, indicating the detected objects along with their respective class labels and confidence scores.

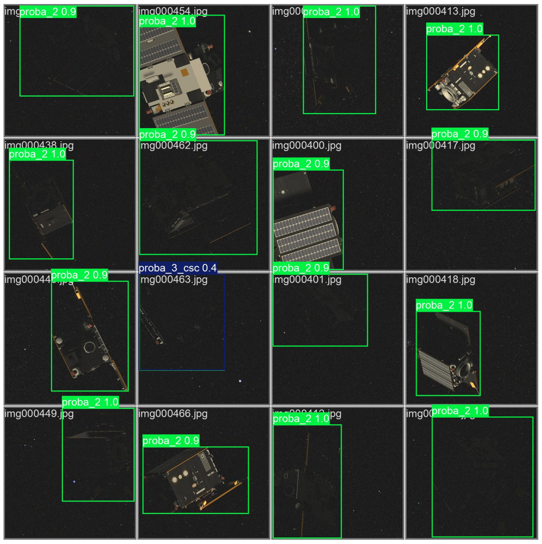

Fig. 8. Example of predictions made by the model on the validation dataset. Bounding boxes show the detected objects with their class labels and confidence scores.

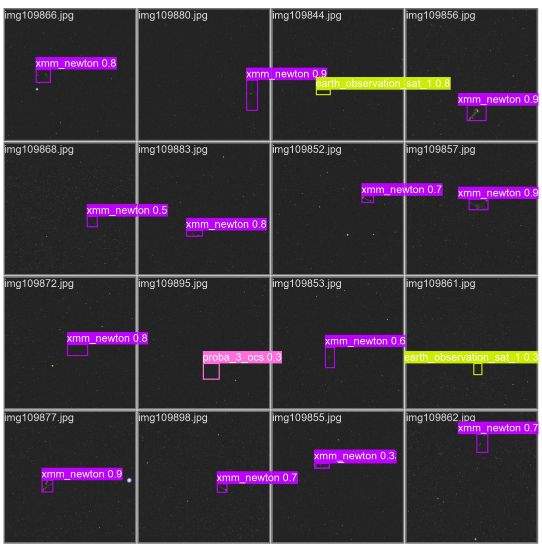

Fig. 9. Example 2 of predictions made by the model on the validation dataset with very small object. Bounding boxes show the detected objects with their class labels and confidence scores.

The results demonstrate the model's ability to accurately localize and classify objects in complex images, despite varying object sizes and lighting conditions. This highlights the robustness of the YOLOv11 algorithm in handling challenging datasets such as SPARK 2022.

Acknowledgment

We would like to express our sincere gratitude to Professors Djamila AOUADA and Anis KACEM for their insightful courses and the valuable knowledge they have shared with us at the University of Luxembourg. Their teachings have been instrumental in deepening our understanding of computer vision and artificial intelligence.

We also extend our heartfelt thanks to our supervisor, Peyman ROSTAMI ABENDANSARI, for his guidance, support, and constructive feedback throughout the course of this project. His expertise and encouragement have been essential to the success of our work.

Contribution & Workflow

The 4 members of the group worked collaboratively, meeting every week at the Belval Learning Centre to make progress at the same time and solve problems together. In addition to this, the 4 members held several meetings each week with their supervisor to make the most of the project.

References

[1] Ultralytics, "Yolo models," 2025, accessed: 15-Jan-2025. [Online]. Available: https://docs.ultralytics.com/fr/models/featured-models