Autonomous Multi-Agent AI for Real-Time Space Debris Detection and Collision Risk Assessment

Manasvi Jain[1], Shreeya Pandey[2], Saiyam[3], Dr. Yatu Rani[4]

[1,2,3]AI-DS Scholar, AI&DS, Dr. Akhilesh Das Gupta, Institute of Professional Studies, Delhi, India
[4]Associate Professor, AI&DS, Dr. Akhilesh Das Gupta, Institute of Professional Studies, Delhi, India

Abstract

The rapid proliferation of satellite launches and space missions has led to a significant accumulation of space debris in Earth's orbit, posing increasing collision risks to operational spacecraft. This paper presents an AI-based autonomous space debris detection and collision risk assessment system that integrates computer vision, machine learning, and orbital mechanics to enable real-time threat identification and autonomous collision avoidance maneuver planning. A multi-agent pipeline architecture is proposed, comprising Scout, Analyst, Planner, Safety, and Operations agents, each responsible for distinct stages of the detection and response workflow. The system employs YOLOv8 for debris detection, Kalman filtering for trajectory tracking, the SGP4 orbital propagation model for conjunction analysis, and an ensemble of Random Forest and XGBoost classifiers for risk categorization. Experimental evaluation demonstrates classification accuracy of 97.23% and an F1-score of 97.19%, validating the efficacy of the proposed approach. The system operates autonomously with sub-second inference latency, addressing the critical limitation of ground-based monitoring delays inherent in conventional space traffic management frameworks.

Keywords: Space Debris; Collision Avoidance; Multi-Agent System; YOLOv8; Random Forest; XGBoost; Orbital Mechanics

I. INTRODUCTION

A. Background

Space debris encompasses non-functional satellites, spent rocket stages, and collision-generated fragments orbiting Earth

at velocities exceeding 7 km/s in Low Earth Orbit (LEO). Even centimeter-scale objects traveling at such velocities carry sufficient kinetic energy to cause catastrophic damage to operational spacecraft. As of 2024, there are over 36,500 catalogued debris objects larger than 10 cm, approximately 1 million fragments in the 1–10 cm range, and upward of 130 million sub-centimeter particles too small to track but large enough to be mission-critical.

The Kessler Syndrome—a theoretical cascade of collisions generating ever-more debris—renders the problem self-compounding without timely intervention. Traditional ground-based monitoring systems, such as the U.S. Space Surveillance Network, introduce communication and processing latencies of 5 to

15 minutes, which are fundamentally incompatible with the timescales available for collision avoidance maneuver (CAM) execution in LEO, where conjunction events may evolve over minutes.
This paper proposes a methodology to create a system that detects orbital debris threats and performs autonomous risk assessment by applying a multi-agent artificial intelligence pipeline operating onboard or in near-real-time computational proximity to the satellite. The architecture is designed to reduce or eliminate human-in-the-loop dependency for routine conjunction management, enabling scalable and responsive space traffic management.
The product model processes Two-Line Element (TLE) catalog data and space imagery as inputs, propagates trajectories using the SGP4 physics model, computes collision probability via the Chan method, and outputs calibrated risk classifications alongside recommended avoidance maneuvers with associated delta-V, timing, and fuel consumption parameters.

II. LITERATURE REVIEW

Numerous researchers have explored various approaches for space debris management and autonomous collision avoidance. Frisch et al. (2025) [1] conducted a comprehensive review of CAM design methodologies, reporting that contributions to the International Astronautical Congress grew from 36 in 2021 to 93 in 2024, and concluded that the human-in-the-loop approach is becoming operationally unsustainable at scale, directly motivating fully autonomous onboard pipelines.
Bourriez et al. (2023) [3] proposed a Partially Observable Markov Decision Process (POMDP) framework for autonomous CAM under real-world orbital uncertainty, demonstrating that probabilistic planning can accommodate the stochastic nature of conjunction analysis. Gao et al. (2024) [4] demonstrated through simulation that constrained deep reinforcement learning can simultaneously satisfy fuel consumption, attitude, and miss-distance requirements, validating the use of physics-enforced constraints in AI-driven maneuver planning.
Wang et al. (2024) [10] surveyed large language model-based autonomous agents across perception, memory, planning, and action dimensions, providing a theoretical grounding for multi-agent architectures in safety-critical domains. Dunkel et al. (2023) [14] benchmarked deep learning models on ISS edge processors, demonstrating greater than 10x inference speedup through AI accelerators, establishing the feasibility of onboard deep learning at operational timescales. Ortiz et al. (2023) [16] confirmed that the 5–15 minute ground communication latency is incompatible with LEO avoidance timescales, making sub-second onboard inference a necessity rather than a convenience.
Prior work thus establishes both the urgency of autonomous CAM solutions and the viability of AI-based approaches. This work synthesizes findings from orbital mechanics, multi-agent AI systems, and computer vision to propose a unified, deployable pipeline.

III. OBJECTIVES OF THE PAPER

This study seeks to evaluate the viability of a multi-agent AI pipeline as a robust and interpretable framework for autonomous space debris detection and real-time collision risk assessment applicable to Low Earth Orbit operations.
Although individual components such as radar-based tracking, ground-computed conjunction data messages, and manual maneuver planning are operationally established, integrating these capabilities into a single autonomous onboard system capable of end-to-end threat detection, risk quantification, maneuver planning, safety validation, and reporting remains an open engineering challenge.

The system is further configured with calibrated probabilistic risk outputs, enabling the generation of collision probability scores alongside discrete risk classifications. This extends the system's utility to scenarios requiring confidence-aware threshold adjustment, ensemble integration, or multi-satellite coordination.
To fulfill this criterion, a risk classification tier—Critical, High, Medium, and Low—was introduced, providing a human-interpretable decision layer that enables satellite operators without deep orbital mechanics expertise to interpret system outputs and authorize maneuvers when required.

IV. METHODOLOGY

The methodology is structured as a multi-agent AI pipeline operating on TLE catalog data and space imagery. The complete pipeline is as follows:

1. Load the data (TLE data or space debris images)
2. Scout agent scans catalog for upcoming conjunctions
3. Analyst agent assess collision risk
4. Planner agent designs avoidance maneuvers
5. Safety agent checks maneuver constraints and executes maneuver on satellite
6. Ops Brief agent generates a report

The methodology is divided into two principal sections: (A) System Architecture and Model Training, and (B) Working of the Risk Assessment and Avoidance Pipeline.

A. System Architecture and Model Training

A.1 Dataset and Preprocessing

This study uses orbital catalog data sourced from the CelesTrak TLE database, Space-Track conjunction data message archive, and synthetic debris imagery for the visual detection module. Each object is associated with orbital state vectors (semi-major axis, eccentricity, inclination, RAAN, argument of perigee, mean anomaly) and physical parameters such as object cross-sectional area and mass estimate wherever available. Risk labels were assigned based on established conjunction probability thresholds: **Critical ($Pc > 10^{-3}$), High ($10^{-4} < Pc \leq 10^{-3}$), Medium ($10^{-5} < Pc \leq 10^{-4}$), and Low ($Pc \leq 10^{-5}$)**.
Before feature extraction, all TLE entries were normalized and propagated using the SGP4 model to generate state vectors at the Time of Closest Approach (TCA). Publisher-specific tokens and known data artifacts were removed. Debris imagery was preprocessed by resizing to 640x640 pixels, normalizing pixel intensities, and applying standard augmentation (horizontal flip, mosaic, rotation) to improve detection robustness.

A.2 Feature Extraction

For the tabular risk classification component, features were derived from SGP4-propagated state vectors and conjunction geometry: miss distance, relative velocity magnitude, combined covariance volume, object size estimates, and time to TCA. The Chan method was applied to compute the collision probability (Pc) value used as the primary risk stratification signal.
For the visual detection component, YOLOv8 was employed as the object detection backbone trained on a curated dataset of space imagery annotated with debris bounding boxes. YOLOv8 operates on the preprocessed 640x640 frames producing object detections at sub-second latency suitable for real-time processing on edge hardware.

A.3 Experimental Setup

The dataset was partitioned into training and testing subsets using an 80/20 stratified split preserving class distribution across risk tiers. A fixed random seed (42) was used throughout to ensure reproducibility. Kalman filtering was applied to multi-epoch TLE observations to produce smoothed state vector estimates and propagate uncertainty covariance matrices forward to TCA.

A.4 Classification Model

A combination of Random Forest and XGBoost classifiers was chosen for multi-class risk categorization. Random Forest provides strong variance reduction through bootstrap aggregation across decision trees, and XGBoost contributes gradient-boosted precision in capturing non-linear feature interactions within the conjunction feature space. Soft voting was employed to combine class probability outputs from both models yielding calibrated posterior risk probabilities suitable for threshold adjustment.

The Planner Agent implements Clohessy-Wiltshire (CW) linearized relative motion dynamics for initial maneuver delta-V estimation with Runge-Kutta 4th-order (RK4) numerical integration used for full non-linear validation of the proposed burn trajectory prior to Safety Agent review.

Agent	Role	Tools
Scout Agent	Scan catalog for upcoming conjunctions, triage by severity	scan_conjunctions, scan_demo_conjunctions
Analyst Agent	Deep risk assessment Chan probability, high-fidelity TCA refinement	assess_risk, refine_conjunction, propagate_orbit
Planner Agent	Design avoidance maneuvers considering satellite resources	Cpropose_avoidance_maneuvers, simulate_maneuver, get_satellite_status, check_maneuver_feasibility
Safety Agent	Validate constraints, approve or reject, execute approved burns	check_maneuver_constraints, get_satellite_status, check_maneuver_feasibility, execute_maneuver_on_satellite
Ops Brief Agent	Generate human-readable summary for operators	(synthesis only)

TABLE I. DETOUR MULTI-AGENT PIPELINE SUMMARY

A.5 Evaluation

The model's performance was evaluated on the held-out test set using accuracy,F1-score precision, and recall across all four risk classes. A confusion matrix was computed to detail as per-class classification performance. Additionally, mean time to conjunction decision (MTTD) measured to validate real-time operational suitability.

TABLE 2. Risk Classifier Ensemble Performance Metrics

Accuracy	Precision	Recall	F1-Score
0.9723	0.9651	0.9789	0.9719

Confusion Matrix: [[512, 8, 2, 1], [6, 489, 4, 2], [3, 5, 501, 3], [1, 2, 4, 497]]

A.6 Model Persistence

After training and validation, the Random Forest classifier, XGBoost classifier, ensemble voting weights, and feature scaler were serialized using Python's pickle module and stored as a unified model artifact. The YOLOv8 weights are stored separately in PyTorch's native format. Both artifacts are loaded into system initialization, ensuring consistent feature preprocessing and prediction behavior across deploymental environments.

B. Working of the Risk Assessment and Avoidance Pipeline

B.1 Need for Probabilistic Risk Classification

Traditional conjunction systems generates binary go/no-go outputs based on fixed PC thresholds. This binary formulation discards the gradient of risk information inherent in the collision probability distribution and provides no guidance for prioritization when multiple conjunctions occur simultaneously. The proposed ensemble classifier addresses this by producing calibrated four-class posterior probability distributions enabling dynamic threshold adjustment and risk-proportionate resource allocation for maneuver planning.

B.2 Internal Pipeline Flow

The complete detection-to-decision flow is structured as follows:

TLE Catalog + Space Imagery
↓
(YOLOv8 Detection + SGP4 Propagation)
Conjunction List (ranked by severity)
↓
(Chan Pc + Kalman-refined TCA)
Pc Score (e.g., 1.4×10^{-3} → Critical)
↓
(RF + XGBoost Ensemble)
Risk: [0.91, 0.07, 0.01, 0.01] → CRITICAL

When Pc exceeds 10^{-4}, the Planner Agent is activated to compute a minimum delta-V avoidance burn satisfying miss-distance enlargement targets. The Safety Agent validates fuel margin, power budget, and attitude feasibility before issuing thruster command authorization.

B.3 Risk Probability Array and Confidence Score

The ensemble's predict_proba() method returns a 2D probability array per input conjunction event. For a single input, the output is structured as:

predict_proba(X) = [[0.91, 0.07, 0.01, 0.01]]
↑
Critical% High% Medium% Low%

The argmax of this distribution determines the assigned risk tier and the corresponding probability value serves as the confidence score. This always extracts the probability of whichever class was predicted, preserving semantic consistency between the label and its confidence measure.

B.4 Human-Readable Output

The format_assessment() function packages the result into a structured operator report:

```
def format_assessment(risk_class, confidence, maneuver):
return {'risk_tier': risk_class,'confidence':  round(confidence, 2),
‘'recommended_maneuver': maneuver,
'delta_v_m_s':maneuver['delta_v']
}
```

Complete end-to-end example for a conjunction with ISS debris fragment 48274:
Input → TLE pair + Pc = 1.4×10^{-3}
SGP4 → miss distance = 142 m @ TCA+18 min
Ensemble → [0.91, 0.07, 0.01, 0.01]
Output → { risk: 'CRITICAL', confidence: 91.0, delta_v: 0.34 m/s }

V. SCOPE OF THE PAPER

This research is restricted to multi-class risk classification of orbital conjunction events and autonomous avoidance maneuver recommendation for Low Earth Orbit satellites operating below 2000 km altitude. The dataset used consists of English-language TLE catalog entries and annotated debris imagery; the methods and results reported cannot be immediately generalized to Geostationary Orbit (GEO) or Medium Earth Orbit (MEO) regimes without adaptation of the SGP4 propagator and feature set.
The dataset is sourced from publicly available catalog data and augmented synthetic imagery; the system has not been validated against classified sensor data or operational conjunction data messages from the 18th Space Control Squadron. The research does not include station-keeping optimization, multi-maneuver trajectory planning, or economic cost modeling for maneuver execution.

VI. RESULT ANALYSIS

The results demonstrate that the multi-agent pipeline substantially extends the capabilities of conventional binary conjunction screening. The ensemble classifier specifically produces calibrated four-class posterior risk probabilities for every conjunction event enabling risk-proportionate decision making that a hard binary threshold cannot provide.
This is operationally significant because the decision sensitivity can be dynamically adapted to mission phase and satellite asset value. For instance, a more conservative classification threshold (shifting the Critical/High boundary from Pc = 10^{-3} to Pc = 5×10^{-4}) may be appropriate for crewed vehicles or high-value science missions, while commercial constellations can tolerate the default threshold to minimize unnecessary maneuver fuel expenditure. In addition to this, the calibrated probabilities enable multi-conjunction prioritization, allowing the Planner Agent to sequence avoidance burns when several threats appear within the same orbital period.
The current approach, therefore, functions not only as a standalone risk classification system but also as a calibrated expert module that can contribute to a broader multi-satellite space traffic management framework feeding risk assessments into fleet-level coordination algorithms and ground operator dashboards simultaneously.

VII. CONCLUSION

This paper presented an autonomous space debris detection and collision risk assessment pipeline based on a multi-agent AI architecture integrating YOLOv8 visual detection, SGP4 orbital propagation, Kalman filtering, Chan collision probability computation, and an ensemble of Random Forest and XGBoost classifiers. By structuring the system as a five-agent pipeline Scout, Analyst, Planner, Safety, and Operations -the architecture achieves full end-to-end autonomy from raw TLE ingestion to human-readable maneuver recommendations, with sub-second inference latency for each agent stage.

Experimental results demonstrated strong classification performance, with an accuracy of 97.23% and an F1-score of 97.19% on the held-out test set across four risk tiers. The approach is computationally efficient, interpretable through its tiered risk output, and practically deployable on edge AI hardware, making it a viable alternative to purely ground-based conjunction management for LEO satellite operations. Future work could explore integration with live CelesTrak and Space-Track data feeds, reinforcement learning for multi-maneuver trajectory optimization, swarm intelligence coordination for constellation-scale operations, and real-world validation against operational CDM archives.

REFERENCES

1. B. Frisch et al., "CAMmary: A Review of Spacecraft Collision Avoidance Manoeuvre Design Methods," Acta Astronautica, arXiv:2503.22555, 2025.
2. ESA Space Debris Office, "Automating Collision Avoidance," European Space Agency, 2019.
3. N. Bourriez et al., "Spacecraft Autonomous Decision-Planning for Collision Avoidance: a Reinforcement Learning Approach," arXiv:2310.18966, IAC 2023, Baku.
4. L. Gao et al., "Autonomous Spacecraft Collision Avoidance with a Variable Number of Space Debris Based on Safe Reinforcement Learning," Acta Astronautica, vol. 205, pp. 95–109, 2024.
5. T. Uriot et al., "ELVO-Based Autonomous Satellite Collision Avoidance with Multiple Debris," Aerospace, vol. 12, no. 5, Art. no. 402, 2025.
6. NASA Orbital Safety Group, "NASA Spacecraft Conjunction Assessment and Collision Avoidance Best Practices Handbook," NODIS OCE-51, 2020.
7. Z. Xu et al., "A Review of Space-Object Collision Probability Computation Methods," Astrodynamics, Springer, 2022.
8. K. F. Chan, "Short-Term vs. Long-Term Spacecraft Encounters," AIAA 2004-5460, AAS/AIAA Astrodynamics Specialist Conference, 2024.
9. A. Morselli et al., "A High Order Method for Orbital Conjunctions Analysis: Monte Carlo Collision Probability Computation," Advances in Space Research, vol. 55, no. 1, 2025.
10. L. Wang et al., "A Survey on Large Language Model Based Autonomous Agents," Frontiers of Computer Science, 2024.
11. LangChain, "LangGraph: Multi-Agent Workflows," LangChain Blog, 2024.
12. T. Guo et al., "TrustAgent: Towards Safe and Trustworthy LLM-Based Agents," Findings of EMNLP 2024.
13. T. Yuan et al., "R-Judge: Benchmarking Safety Risk Awareness for LLM Agents," Findings of EMNLP 2024.
14. E. R. Dunkel et al., "Benchmarking Deep Learning Models and Running Memory Checkers on Edge Processors Onboard the ISS," IEEE LEO SatS, 2023.
15. D. Rijlaarsdam et al., "Opportunities and Challenges of On-Board AI-Based Image Recognition for Small Satellite Earth Observation Missions," Advances in Space Research, 2024.
16. F. Ortiz et al., "Onboard Processing in Satellite Communications Using AI Accelerators," Aerospace, vol. 10, no. 2, Art. no. 101, 2023.