ISSN 2706-655X Vol. 8, Issue 2: 121-127, DOI: 10.25236/IJFET.2026.080215

Architecture Design and Cooperative Operation Mechanism of an Airport Unmanned Intelligent Operation System

Hongxin Tan[a,*], Yuejian Xu[b]

Science and Technology on Complex Aviation Systems Simulation Laboratory, Beijing, China
[a]zll_txx@163.com, [b]xuyuejian_123@163.com
**Corresponding author*

Abstract: *Unmanned intelligent operation is essential for airport ground support, covering baggage and cargo transfer, apron and pavement inspection, perimeter patrol, snow removal, cleaning and disinfection, and emergency rescue. However, existing practice often remains limited to single-scenario robots, isolated pilot deployments, or fixed-schedule dispatch, and therefore struggles with cross-scenario task sharing, airside safety compliance, and degraded operation. This paper proposes an airport unmanned intelligent operation system integrating a UAV fleet, heterogeneous ground robots, fixed sensing infrastructure, edge computing nodes, and a dispatch center. A dual-closed-loop architecture with six functional layers and two support pillars is designed: the data loop supports demand access, unified task modeling, situation construction, and progress tracking, while the control loop supports priority assessment, capability matching, joint scheduling, route re-planning, and execution feedback. A unified operation task model, a robot capability model, a market-based matching model, a joint task-allocation optimization model, and an explicit degradation rule are established. A discrete-event simulation with 120 Monte Carlo runs, 48 random operation tasks per run, and four baseline schemes is constructed on a 3 km × 2 km airport layout. Under the stated assumptions, the proposed scheme achieves a 96.8% task completion rate, reduces the mean response delay from 74.81 s to 52.64 s compared with a heterogeneous fleet under static allocation, and increases timely completion within 180 s from 76.9% to 87.5%. The results indicate that the value of the architecture lies not only in adding robots, but also in transforming heterogeneous unmanned equipment into a closed-loop, safety-compliant, and engineering-deployable operation capability.*

Keywords: *Airport unmanned operation, UAV and ground robot, Task allocation, Cooperative scheduling, Safety compliance, System reliability*

1. Introduction

Airports are highly organized operation sites in which baggage and cargo transfer, apron servicing, pavement and foreign object debris (FOD) inspection, perimeter patrol, snow and ice removal, cleaning and disinfection, and emergency rescue must be performed continuously under strict safety and timeliness constraints. Manual ground support is labor-intensive, weather-sensitive, and difficult to trace, while a single type of unmanned platform cannot simultaneously satisfy payload, endurance, coverage, precision, and airside safety requirements. UAVs provide rapid viewpoint reconfiguration and wide-area inspection, whereas ground robots provide payload capacity, close-range manipulation, and long-endurance service. Therefore, the core problem is no longer whether one robot can finish one job, but how to organize heterogeneous unmanned equipment, communication, computation, and decision resources into a dependable cooperative operation system covering the full set of airport logistics scenarios.

Cooperative heterogeneous multi-robot systems have been surveyed systematically, and task decomposition, coalition formation, and task allocation are identified as core problems [1]. A formal taxonomy of multi-robot task allocation provides the theoretical basis for matching tasks to robots [2]. For air-ground cooperation, memetic path planning and imitation-augmented deep reinforcement learning have improved UAV-UGV cooperative detection and tasking [3-4]. In airport applications, multi-robot coordination has been studied for autonomous runway FOD clearance [5]; small unmanned aircraft systems have been evaluated for airfield pavement crack detection and rating [6]; deep-learning-based small-scale FOD detection has reached real-time performance [7]; and computer-vision

ISSN 2706-655X Vol. 8, Issue 2: 121-127, DOI: 10.25236/IJFET.2026.080215

analysis has been reviewed across airport applications [8]. Airport digital-twin platforms support asset management and operation rehearsal [9], and LiDAR-equipped rovers have been demonstrated for smart airport FOD detection [10]. Nevertheless, these studies usually emphasize one scenario, one robot type, or one algorithm rather than a complete system architecture for airport-wide unmanned operation.

Three limitations remain in practical construction. First, operation demands from flight plans, sensor alarms, and manual orders are handled by separate subsystems without a unified task model, so cross-scenario resource sharing is impossible. Second, task allocation and route planning are frequently separated from airside safety rules, platform energy state, and failure risk. Third, many pilot deployments lack explicit degradation rules, safety-compliance evidence, and engineering verification paths. To address these issues, this paper makes four contributions: (1) a dual-closed-loop and six-layer system architecture for airport unmanned intelligent operation is proposed; (2) a unified operation task model and a robot capability model are established; (3) a market-based matching and joint task-allocation method constrained by delay, energy, coverage, communication, and safety is designed; and (4) multi-baseline Monte Carlo simulation is used to evaluate completion, delay, timeliness, and robustness.

2. System Architecture Design

The system is designed around five engineering requirements: full-scenario coverage of airport logistics with a unified task entrance, rapid response to flight-support events within strict time windows, edge-cloud collaborative intelligence, fault-tolerant operation under degraded communication, and modular deployment with standardized interfaces.

The mission application layer covers six scenario families: baggage and cargo transfer, apron and pavement inspection, perimeter patrol, snow removal, cleaning and disinfection, and emergency rescue. The heterogeneous execution plane contains a UAV fleet, transport robots, inspection UGVs, and special-duty robots. UAVs carry EO/IR payloads for FOD survey, pavement inspection, and perimeter patrol, and can act as communication relays; transport robots include baggage, cargo, and catering AGVs and autonomous tow tractors; inspection UGVs carry EO and LiDAR sensors for apron patrol, lighting and navigation-aid checking; special-duty robots perform snow removal, firefighting, disinfection, and cleaning. The communication and security plane integrates a 5G/AeroMACS private network with mesh backup, RTK positioning, time synchronization, identity authentication, encryption, and audit. The collaborative intelligence plane performs demand access, task decomposition, capability matching, multi-agent scheduling, anomaly prediction, and model governance. The decision and service plane provides global dispatch, human-machine interaction, digital-twin-assisted rehearsal, and operation KPI management. The airport operation situation graph at the core of the two loops maintains spatiotemporal data, a task library, the airport high-definition map with geofences, operation rules, and model versions.

The architecture contains two coupled loops. The data loop follows the chain of demand access, unified task modeling, situation construction, and progress tracking. The control loop follows the chain of priority assessment, capability matching, joint scheduling, route re-planning, execution, and feedback. The two loops interact through the airport operation situation graph: when a flight-support event has high priority but no available robot satisfies its capability requirement, the control loop decomposes the task or requests a cooperative robot pair; when platform health or link quality changes, the data loop updates capability parameters and the scheduler recalculates the allocation. This design distinguishes the proposed system from a simple collection of single-scenario robots.

$$\tau_j = \langle c_j,\ \ell_j,\ w_j,\ [t_j^r, t_j^d],\ q_j \rangle \tag{1}$$

In Eq. (1), τ_j is the unified model of operation task j; c_j is the task category drawn from the six scenario families; l_j is the task location expressed in the airport high-definition map; w_j is the workload estimate; $[t_{jr}, t_{jd}]$ is the release-deadline time window; and q_j is the quality requirement, including precision, evidence, and reporting demands. This model provides a unified interface through which flight-plan events, sensor alarms, and manual orders enter the same scheduling framework.

For engineering implementation, each demand source encapsulates task category, location, time window, priority, payload requirement, and evidence requirement into a standard task message. Edge nodes attached to aprons, cargo areas, and the perimeter perform lightweight validation, local

ISSN 2706-655X Vol. 8, Issue 2: 121-127, DOI: 10.25236/IJFET.2026.080215

feasibility checking, and message caching; the regional scheduling node performs matching and allocation; and the dispatch center performs mission-level supervision. The hierarchical division prevents demand flooding and enables the system to continue local operation when the wide-area link is interrupted.

3. Cooperative Operation Model and Scheduling Method

The cooperative operation mechanism is organized as an event-driven closed loop. Demand sources first generate task messages with quality tags. Each robot reports a capability vector that is updated with health and energy state. The scheduler then performs market-based matching, multi-objective allocation, and route re-planning under airside safety constraints. If a robot fails or communication degrades, the reliability supervisor triggers redundant-link switching, task reassignment, or graceful degradation. The complete process is shown in Figure 1.

$$r_i(k) = \langle \mathcal{P}_i,\ \mathcal{S}_i,\ v_i,\ e_i(k),\ a_i(k) \rangle \tag{2}$$

In Eq. (2), $r_i(k)$ is the capability vector of robot i at time k; P_i is the payload and tool set, such as cargo bay, manipulator, snow brush, or fire monitor; S_i is the sensor set; v_i is the cruise speed; $e_i(k)$ is the available energy; and $a_i(k)$ is the availability determined by health state, maintenance schedule, and communication quality. Capability vectors of all registered robots form a real-time capability pool, which allows any task to be matched against the whole fleet rather than a preassigned subset.

3.1 Demand access and unified task modeling

First, flight-plan and airport operational database (AODB) events, fixed-station alarms, inspection results, and manual orders are converted into unified task messages. A unified clock and the airport high-definition map align task locations with geofences, stand layouts, and dynamic no-fly/no-go zones. Each task is assigned a priority level according to its coupling with flight operation: flight-support tasks directly affecting departure, such as baggage loading and snow removal before takeoff, receive the highest level; routine patrol and cleaning tasks receive lower levels.

Second, the validation module checks each task against airside rules: whether the target area is inside an active runway protection zone, whether the requested time window conflicts with aircraft movement, and whether the required payload exists in the registered fleet. Invalid tasks are returned with reasons; valid tasks enter the versioned demand pool sorted by priority and time window. This design makes every scheduling decision traceable to an explicit rule set, which is essential for safety audit in airport operation.

3.2 Capability assessment and market-based matching

$$S_{ij}(k) = a_i(k)\, q_i^c(k)\, \varphi(c_j, \mathcal{P}_i) \exp\left(-\frac{d_{ij}(k)}{v_i \Delta t_j} \right) \tag{3}$$

In Eq. (3), $S_{ij}(k)$ is the matching score between robot i and task j; $q_{ic}(k)$ denotes communication quality; $\varphi(c_j, P_i)$ is the capability fit function, which equals one when the payload and tool set of robot i covers the requirement of task category c_j and decays otherwise; $d_{ij}(k)$ is the route distance between the robot and the task location; and Δt_j is the remaining time window. Compared with distance-only dispatch, this formulation incorporates capability fit, communication quality, availability, and urgency, and is therefore more suitable for multi-scenario airport operation.

A market-based bidding procedure is then executed. The demand pool broadcasts task summaries; candidate robots compute their matching scores and bid prices that reflect travel cost, energy cost, and risk; and the scheduler collects bids within a bounded time window. Cooperative tasks that require more than one robot, such as UAV-guided FOD disposal or escorted crossing of an active taxiway, are announced as bundled tasks, and a UAV-UGV pair can submit a joint bid with a rendezvous point. The reliability supervisor continuously monitors link quality, battery level, and fault probability, and lowers the effective score of degraded robots rather than excluding them abruptly.

ISSN 2706-655X Vol. 8, Issue 2: 121-127, DOI: 10.25236/IJFET.2026.080215

3.3 Joint task allocation and route re-planning

$$\min_{u=(x,r)} J(u)=\alpha T(u)+\beta E(u)+\gamma D(u)-\eta C(u)-\lambda R(u) \tag{4}$$

Here, $u = (x, r)$ denotes the joint decision vector consisting of task assignment x and route re-planning r. T denotes response and completion delay; E denotes fleet energy consumption; D denotes communication and computational load; C denotes scenario coverage utility; and R denotes operation risk measured against airside safety rules. The coefficients alpha, beta, gamma, eta, and lambda are adjusted according to scenario type. For flight-support and emergency tasks, delay and risk are prioritized; for routine patrol and cleaning, energy and coverage balance are emphasized.

The constraints include task uniqueness, time windows, robot energy, payload compatibility, geofences and dynamic no-fly/no-go zones, safety separation from aircraft and vehicles, communication quality, and maximum route deviation. A rolling-horizon hierarchical solver is adopted: a rule engine immediately handles highest-priority flight-support events; the market-based allocator generates feasible assignments; and route refinement is performed only for the selected robots. This design avoids solving a large global optimization problem from scratch at every event and is more suitable for online deployment.

$$m^*(k)=\max\{\, m\in\{1,2,3,4\}:\ q_{\text{link}}(k)\geq\theta_m^{\text{link}},\ \ a_{\text{fleet}}(k)\geq\theta_m^{\text{fleet}}\,\} \tag{5}$$

The degradation strategy is defined in four levels, and Eq. (5) selects the highest level m* whose link-quality threshold and fleet-availability threshold are both satisfied. Level 1 is normal cloud-edge collaborative operation. Level 2 switches to mesh backup and reduces nonessential data streams. Level 3 enables autonomous regional clusters and local task caching. Level 4 retains only safety-critical functions, such as runway incursion warning and emergency stop. This explicit degradation chain improves mission continuity and makes the system behavior predictable under faults.

Figure 1 summarizes the cooperative operation and scheduling process. The inner data loop performs demand access, unified task modeling, situation building, and progress tracking around the airport operation situation graph; the outer control loop performs priority assessment, matching and bidding, joint scheduling, route re-planning, and cooperative execution. When situation confidence is low or a high-priority flight-support event arrives, the control loop triggers re-tasking. The reliability supervisor watches both loops and provides health-state inputs to matching and scheduling.

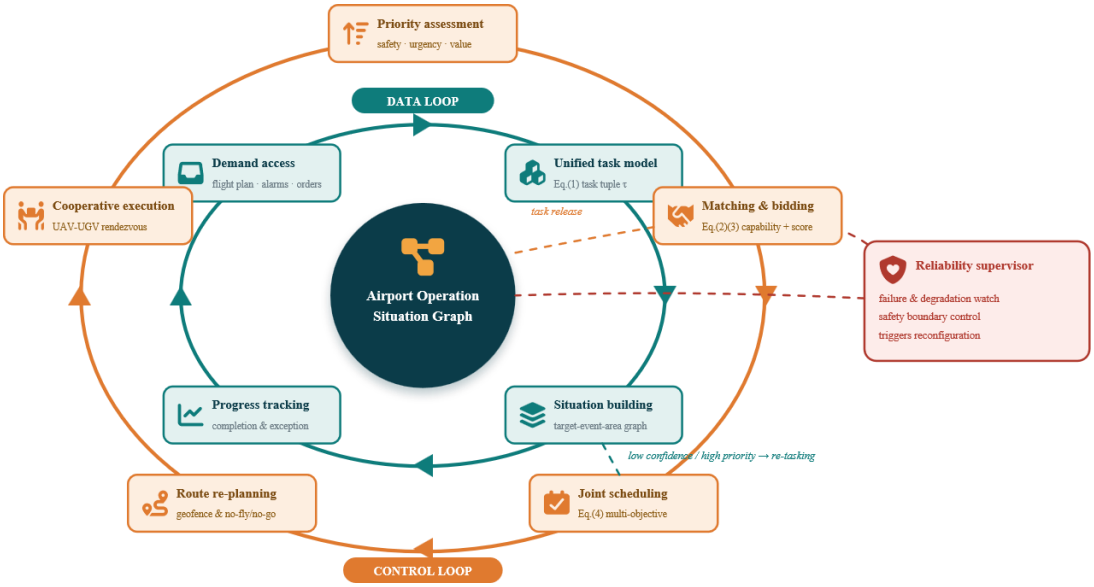


Fig.1. Event-driven cooperative operation and joint scheduling mechanism

4. Multi-Baseline Simulation and Analysis

A discrete-event simulation is constructed on a 3 km × 2 km airport layout to evaluate the architecture under explicit assumptions. Forty-eight operation tasks are randomly generated in each run

ISSN 2706-655X Vol. 8, Issue 2: 121-127, DOI: 10.25236/IJFET.2026.080215

across the six scenario families. Task release times are uniformly distributed within 600 s, and execution is evaluated until 720 s. The scenario contains one runway, one parallel taxiway, an apron with twelve stands, a terminal, a cargo area, and a perimeter line. The fleet contains 6 UAVs and 8 UGVs, including 3 transport robots, 3 inspection UGVs, and 2 special-duty robots. Platform availability, communication quality, cruise route, service duration, and task location are all modeled as variables. The main parameters are listed in Table 1.

Table 1. Main parameters of the multi-baseline simulation

Object	Parameter	Value
System	Mission area / tasks per run	3 km ×2 km / 48 random tasks
Platforms	UAVs / UGVs (transport, inspection, special-duty)	6 / 8 (3, 3, 2)
Motion	UAV / UGV cruise speed	15 / 6 m/s
Payload	Transport / special-duty robot payload	800 / 500 kg
Time	Task release window / evaluation horizon	600 s / 720 s
Availability	UAV / UGV	0.95 / 0.97
Comm.	Link quality / retry interval	0.90-0.99 / 2 s
Evaluation	Monte Carlo runs / timely threshold	120 / 180 s

Four schemes are compared: fixed-schedule dispatch according to a precompiled timetable, a homogeneous UGV fleet without UAV cooperation, a heterogeneous fleet with static allocation and without event-driven re-tasking, and the proposed cooperative architecture. The metrics include task completion rate, mean response delay, 95th-percentile delay, and timely completion within 180 s. Each scheme is evaluated over 120 Monte Carlo runs.

Table 2 summarizes the statistical results. Fixed-schedule dispatch completes only 61.3% of the tasks, because random events cannot be covered by a precompiled timetable. The homogeneous UGV fleet improves the completion rate to 88.7%, but its mean response delay reaches 96.53 s because aerial survey and rapid relay are unavailable. The heterogeneous fleet with static allocation further increases the completion rate to 93.5% and reduces the mean delay to 74.81 s. The proposed architecture achieves a 96.8% completion rate, a mean delay of 52.64 s, and a P95 delay of 156.38 s.

Table 2. Statistical comparison of the four operation schemes

Method	Completion (%)	Mean / P95 delay (s)	Timely ≤180 s (%)
Fixed-schedule dispatch	61.3	158.42 / 402.61	55.2
Homogeneous UGV fleet	88.7	96.53 / 268.14	68.4
Hetero. static allocation	93.5	74.81 / 214.72	76.9
Proposed	96.8	52.64 / 156.38	87.5

Compared with the heterogeneous fleet under static allocation, the proposed scheme reduces the mean response delay by 29.6%, reduces the P95 delay by 27.2%, and increases timely completion within 180 s from 76.9% to 87.5%. The paired-run improvement in mean response delay is 22.17 s with a 95% confidence interval of 19.86-24.48 s, and the difference is statistically significant at $p < 0.001$. These results indicate that the main benefit comes from closed-loop scheduling rather than merely adding more robots.

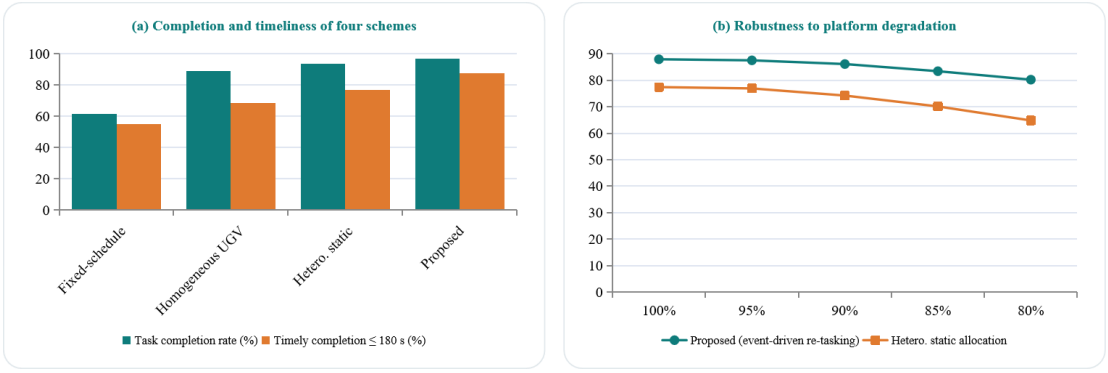


Fig.2.Multi-baseline simulation results and robustness to platform degradation

ISSN 2706-655X Vol. 8, Issue 2: 121-127, DOI: 10.25236/IJFET.2026.080215

Figure 2 further shows the robustness analysis. When mean platform availability decreases from 100% to 80%, the proposed scheme consistently maintains a higher timely-completion rate than static allocation, and the advantage grows from 10.5 to 15.4 percentage points. This advantage comes from three mechanisms: redundant capability across robot types, market-based re-bidding after degradation, and task reassignment supervised by the reliability pillar. Nevertheless, the results are obtained under simplified geometric and communication assumptions and should be interpreted as architecture-level feasibility evidence rather than field-test performance.

5. Engineering Implementation and Application Deployment

The proposed architecture can be implemented incrementally. In the first stage, the dispatch center, the unified task entrance, and one scenario family, typically apron inspection, are deployed to form a baseline capability. In the second stage, transport robots and special-duty robots are connected through standardized task messages, and edge validation plus regional scheduling are enabled. In the third stage, joint scheduling across all scenario families, digital-twin rehearsal, health monitoring, and degraded operation are introduced. This staged path reduces integration risk and allows each stage to produce usable capability.

For baggage and cargo transfer, flight-plan events generate transport tasks automatically, transport robots execute them under geofence constraints, and UAVs provide advance observation of the service road and stand area. For FOD inspection and disposal, UAVs perform scheduled runway sweep, inspection UGVs confirm and remove detected debris, and the evidence chain is archived for audit. For snow removal, special-duty robots are organized into formation operation according to the snowfall forecast and flight schedule, while UAVs evaluate the cleared surface. For emergency rescue, the rule engine preempts routine tasks, dispatches the nearest capable robots, and reserves redundant communication links.

Engineering deployment should follow a unified data model and interface specification. Task messages should contain category, location, time window, priority, payload requirement, evidence requirement, and safety constraints. Capability reports should contain payload and sensor sets, position, energy, health state, and communication quality. Scheduling commands should contain task priority, route, time window, safety constraints, and rollback rules. Model updates should use version management, offline validation, digital-twin rehearsal, and controlled online release. These measures make the architecture auditable, maintainable, and reproducible.

The main engineering risks include RTK degradation near hangars, communication congestion in peak hours, robot endurance in severe weather, sensor fouling on apron surfaces, and model performance degradation across seasons. Mitigation measures include multi-source positioning, adaptive bandwidth allocation, energy-aware scheduling and opportunity charging, periodic sensor self-checking, and continuous model evaluation. The architecture should therefore be treated as a living system in which data, models, robots, and operation rules are jointly maintained.

6. Conclusion

This paper proposes and evaluates a dual-closed-loop airport unmanned intelligent operation system. The architecture integrates a UAV fleet, heterogeneous ground robots, edge intelligence, cloud scheduling, communication security, safety compliance, and reliability supervision into a unified framework covering the main airport logistics scenarios. A unified operation task model, a robot capability model, market-based matching, joint task allocation, and an explicit degradation rule are established to clarify how heterogeneous unmanned equipment is converted into timely, safe, and credible operation capability. Multi-baseline Monte Carlo simulation shows that, under the stated assumptions, the proposed architecture improves completion and timeliness and reduces both mean and tail response delay compared with heterogeneous fleets under static allocation. Future work will focus on real airport data, hardware-in-the-loop verification, certification-oriented safety evidence, human-robot mixed operation, and field validation in operational airports.

References

[1] Rizk Y, Awad M, Tunstel E W. Cooperative heterogeneous multi-robot systems: A survey. ACM Computing Surveys, 2019, 52(2): 1-31.

ISSN 2706-655X Vol. 8, Issue 2: 121-127, DOI: 10.25236/IJFET.2026.080215

[2] Gerkey B P, Mataric M J. A formal analysis and taxonomy of task allocation in multi-robot systems. The International Journal of Robotics Research, 2004, 23(9): 939-954.

[3] Li J, Sun T, Huang X, et al. A memetic path planning algorithm for unmanned air/ground vehicle cooperative detection systems. IEEE Transactions on Automation Science and Engineering, 2022, 19(4): 2724-2737.

[4] Zhang J, Yu Z, Mao S, et al. IADRL: Imitation augmented deep reinforcement learning enabled UGV-UAV coalition for tasking in complex environments. IEEE Access, 2020, 8: 102335-102347.

[5] Öztürk S, Kuzucuoğlu A E. A multi-robot coordination approach for autonomous runway foreign object debris (FOD) clearance. Robotics and Autonomous Systems, 2016, 75: 244-259.

[6] Sourav M A A, Mahedi M, Ceylan H, et al. Evaluation of small uncrewed aircraft systems data in airfield pavement crack detection and rating. Transportation Research Record, 2023, 2677(1): 653-668.

[7] Mo Y, Wang L, Hong W, et al. Small-scale foreign object debris detection using deep learning and dual light modes. Applied Sciences, 2024, 14(5): 2162.

[8] Talbi A, Abdellaoui M, Pons-Prats J, et al. An overview on computer vision analysis in the airport applications. IEEE Access, 2025, 13: 120446-120471.

[9] Keskin B, Salman B, Koseoglu O. Architecting a BIM-based digital twin platform for airport asset management: A model-based system engineering with SysML approach. Journal of Construction Engineering and Management, 2022, 148(5): 04022020.

[10] Elrayes A, Ali M H, Zakaria A, et al. Smart airport foreign object debris detection rover using LiDAR technology. Internet of Things, 2019, 5: 1-11.