CEFET/RJ
DIRETORIA DE PESQUISA E PÓS-GRADUAÇÃO

A Collaborative UAV-UGV Framework for Solar Farm Inspection with Convolutional Neural Network and Synthetic-Real Data Generalization Analysis

Yuri Nascimento da Silva

Defesa de Mestrado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica, Centro Federal de Educação Tecnológica Celso Suckow da Fonseca, CEFET/RJ, como parte dos requisitos necessários à obtenção do título de mestre.

Orientadores:
Milena Faria Pinto
Fábio Augusto de Alcantara Andrade

Rio de Janeiro,
Janeiro, 2026

Ficha catalográfica elaborada pela Biblioteca Central do CEFET/RJ

S586 Silva, Yuri Nascimento da
A collaborative UAV-UGV framework for solar farm inspection with convolutional neural network and synthetic-real data generalization analysis / Yuri Nascimento da Silva. — 2026.
71f. : il. color. , enc.

Dissertação (Mestrado) Centro Federal de Educação Tecnológica Celso Suckow da Fonseca, 2026.
Bibliografia : f. 68-71
Orientadora: Milena Faria Pinto
Coorientador: Fábio Augusto de Alcantara Andrade

1. Drone. 2. Robôs móveis. 3. Coletores solares. 4. Redes neurais (Computação). 5. Percepção visual. I. Pinto, Milena Faria. (Orient.). II. Andrade, Fábio Augusto de Alcantara. III. Título.

CDD 623.7469

Elaborada pela bibliotecária Tania Mello – CRB/7 nº 5507/04

A Collaborative UAV-UGV Framework for Solar Farm Inspection with Convolutional Neural Network and Synthetic-Real Data Generalization Analysis

Defesa de Mestrado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica, Centro Federal de Educação Tecnológica Celso Suckow da Fonseca, CEFET/RJ, como parte dos requisitos necessários à obtenção do título de mestre.

Yuri Nascimento da Silva

Aprovada por:

Presidente, Milena Faria Pinto (orientadora)
CEFET/RJ

Fábio Augusto de Alcantara Andrade, D.Sc. (coorientador)
University of South-Eastern Norway (USN)

Alessandro Rosa Lopes Zachi, D.Sc.
CEFET/RJ

Murillo Ferreira dos Santos, D.Sc.
CEFET-MG

Paulo Fernando Ferreira Rosa
Instituto Mililtar de Engenharia (IME)

Rio de Janeiro,
Janeiro, 2026

ACKNOWLEDGMENTS

The author would like to thank FAPERJ, CNPQ, CEFET/RJ, and CAPES. This work was carried out with the support of the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

I would like to thank my advisors Dr. Milena Pinto and Dr. Fabio Andrade, for the trust placed in me and the guidance provided throughout the project's development.

To my parents Lenita Gonçalves do Nascimento and Erly José da Silva and my siblings Ellen Nascimento da Silva, Fabio Braz da Silva and Felipe Braz da Silva, for all the encouragement and family support throughout my academic development.

To Isabella Dias de Carvalho, for all the love, affection, companionship and encouragement during the days of development and writing.

Yuri Nascimento da Silva

Science can amuse and fascinate us all, but
it is engineering that changes the world.

Isaac Asimov

RESUMO

A Collaborative UAV-UGV Framework for Solar Farm Inspection with Convolutional Neural Network and Synthetic-Real Data Generalization Analysis

Yuri Nascimento da Silva

Orientadores:
Profa. Milena Faria Pinto
Prof. Fabio Augusto de Alcantara Andrade

Defesa de Mestrado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica, Centro Federal de Educação Tecnológica Celso Suckow da Fonseca, CEFET/RJ, como parte dos requisitos necessários à obtenção do título de mestre em Engenharia Elétrica.

Esta dissertação propõe uma arquitetura multirrobótica abrangente para a inspeção autônoma de parques fotovoltaicos, integrando plataformas robóticas aéreas e terrestres. O sistema emprega um Veículo Aéreo Não Tripulado (UAV) para a aquisição de imagens e uma Rede Neural Convolucional You Only Look Once versão 8 (YOLOv8) para a detecção em tempo real de painéis solares, enquanto um Veículo Terrestre Não Tripulado (UGV) atua como uma base móvel de suporte e estação de recarga. A arquitetura é estruturada hierarquicamente e possibilita o comportamento colaborativo por meio do planejamento otimizado de trajetórias baseado em um Algoritmo Genético, que minimiza o tempo de inspeção e o consumo de energia a partir das localizações dos painéis detectados. O UAV opera sob a estrutura de piloto automático PX4, comunica-se por meio do protocolo MAVLink e integra-se ao middleware do Sistema Operacional do Robô (ROS). Paralelamente, este estudo investiga o impacto da composição dos dados de treinamento no desempenho da detecção de objetos, analisando como diferentes proporções de imagens sintéticas e do mundo real afetam a capacidade de generalização do modelo YOLOv8. Conjuntos de dados sintéticos foram gerados por meio de simulações no ambiente ROSGazebo para complementar imagens aéreas reais. Os resultados experimentais demonstram que modelos treinados com uma combinação equilibrada de dados reais e simulados apresentam maior robustez entre domínios quando avaliados em conjuntos de teste reais, simulados e híbridos. Do ponto de vista quantitativo, o detector baseado em YOLOv8 proposto alcançou um F1-score de 0,91 nas tarefas de detecção de painéis solares, com baixas taxas de falsos positivos e falsos negativos. Além disso, configurações de treinamento que combinam dados reais e sintéticos, em especial a proporção 50%50%, superaram significativamente modelos treinados exclusivamente em um único domínio, confirmando

melhorias na generalização e na confiabilidade da detecção. O framework foi validado em ambientes de simulação realistas, confirmando sua robustez, escalabilidade e autonomia operacional para a inspeção inteligente e economicamente eficiente de parques solares.

Palavras-chave: Colaboração UAV-UGV. Inspeção de painéis solares. Sistemas multi-robôs. Planejamento de trajetória de cobertura. Otimização de trajetória. Inspeção robótica. Rede neural convolucional. Detecção visual com YOLOv8.

ABSTRACT

A Collaborative UAV-UGV Framework for Solar Farm Inspection with Convolutional Neural Network and Synthetic-Real Data Generalization Analysis

Yuri Nascimento da Silva

Advisor:
Profa. Milena Faria Pinto
Prof. Fabio Augusto de Alcantara Andrade

Master Thesis Defense submitted to Electrical Engineering Graduate Program, Federal Center of Technological Education Celso Suckow da Fonseca, CEFET/RJ, as partial fulfillment of the requirements for the degree of Master in Electrical Engineering.

This dissertation proposes a comprehensive multi-robot architecture for the autonomous inspection of Photovoltaic (PV) farms, integrating aerial and ground robotic platforms. The system employs an Unmanned Aerial Vehicle (UAV) for image acquisition and a You Only Look Once version 8 (YOLOv8) Convolutional Neural Network (CNN) for real-time solar panel detection, while an Unmanned Ground Vehicle (UGV) serves as a mobile support base and charging station. The architecture is hierarchically structured and enables collaborative behavior through optimized path planning based on a Genetic Algorithm (GA), which minimizes inspection time and energy consumption using the detected panel locations. The UAV operates under the PX4 autopilot framework, communicates via the MAVLink protocol, and integratesthe Robot Operating System (ROS). In parallel, this study investigates the impact of training data composition on object detection performance, analyzing how different proportions of synthetic and real-world images affect the YOLOv8 model's generalization. Synthetic datasets were generated using ROSGazebo simulations to complement real aerial imagery. Experimental results demonstrate that models trained with a balanced combination of real and simulated data achieve superior cross-domain robustness when evaluated on real, simulated, and hybrid test sets. Quantitatively, the proposed YOLOv8-based detector achieved an F1-score of 0.91 on solar panel detection tasks, with low false-positive and false-negative rates. Moreover, training configurations combining real and synthetic data (notably the 50%50% split) significantly outperformed models trained exclusively on a single domain, confirming improved generalization and detection reliability. The framework was validated in realistic simulation environments, confirming its robustness, scalability, and operational autonomy for intelligent and cost-effective solar farm inspection.

Keywords: UAV-UGV Collaboration. Solar Panel Inspection. Multi-Robot Systems. Coverage Path Planning. Path Optimization. Robotic Inspection. Convolution Neural Network. Visual Detection With YOLOv8.

Contents

I Introduction **15**

I.1 Initial Considerations and Work Motivation 15

I.2 Main Contributions 17

I.3 Organization 18

II Related Works **20**

II.1 Heterogeneous Robots Cooperation 20

II.2 UAVs Applied to Solar Panel Farms Inspection 21

II.3 Discussions 22

III Proposed Methodology **25**

III.1 Hardware Setup 26

III.2 Cooperative Framework 27

III.3 Yolov8 Neural Network Structure 29

III.4 Inspection Routine 30

III.4.1 Mathematical Formulation of the Optimization Problem 31

III.4.2 Mathematical Formulation to extract coordinates during inspection steps 32

III.5 Impact of Real and Simulated Data Composition 38

IV Results and Discussion **40**

IV.1 Solar Panels Detection 40

IV.2 Framework results 42

IV.3 Evaluation of the YOLO model 46

IV.3.1 Testing on real datasets 46

IV.3.2 Testing on simulated datasets 52

IV.3.3 Testing on 50% real and 50% simulated datasets 57

IV.3.4 Discussion and analysis 62

V Conclusions and Future Work **65**

V.1 Conclusions 65

V.2 Future Proposals and Limitations 66

Bibliographic References 68

List of Tables

II.1 Comparison of related works and the proposed approach. 23

IV.1 Confusion matrix for YOLOv8m predictions on the solar panel dataset. 40
IV.2 Comparison of confusion matrices for YOLOv8m predictions on real test datasets. 51
IV.3 Comparison of confusion matrices for YOLOv8m predictions on simulated test datasets. 57
IV.4 Comparison of confusion matrices for YOLOv8m predictions on 50% real and 50% simulated test datasets. 62
IV.5 Comparison of metrics by model predictions on real test datasets. 64
IV.6 Comparison of metrics by model predictions on simulated test datasets. 64
IV.7 Comparison of metrics by model predictions on 50% real and 50% simulated test datasets. 64

List of Figures

III.1 Flowchart of the general idea. (1) UGV navigates to a predetermined location within the solar farm. (2) UAV is deployed to perform aerial inspections. (3) After completing the inspection or when the UAV's battery level reaches a critical threshold, the UAV signals the UGV to prepare for landing. The UGV moves to a suitable position to facilitate a safe landing for the UAV. 25
III.2 Simulated Solar Panel Farm and the robots used in the simulation. In this context, the UGV can be seen as a mobile "drone-in-a-box" platform, enabling similar functionalities in environments where static docking stations are not viable. 26
III.3 ROS active nodes and messages. 27
III.4 Cooperation framework. 28
III.5 UAV and UGV cooperation flowchart. 29
III.6 YOLOv8 Structure. 30
III.7 Dataset images from (ISHII, 2023) and (SILVA, 2025). 31
III.8 2D illustration considering roll or pitch inclination equal to zero. 33
III.9 2D illustration considering a non-zero roll or pitch inclination. 34
III.10 Relation between the x and y axes, roll and pitch of the drone, with the width and height of the image captured by the camera. 35
III.11 Overview of the analysis of the real and simulated data composition pipeline. 39

IV.1 F1 and Precision x Recall Curve. 41
IV.2 Training and validation metrics of the YOLOv8m model over 30 epochs. Loss curves indicate effective convergence, while precision, recall, and mAP metrics demonstrate strong performance and generalization capability. 41
IV.3 YOLO detection process applied in the proposed scenario, with the X and Y coordinates of the centers of the solar panels referenced to the coordinates of the simulated world. . 42
IV.4 Path performed by the UAV during the pre-inspection routine. 43
IV.5 Multi-objective GA distance and time optimization graph. (a) Total Distance Evolution. (b) Total Time Evolution. 44
IV.6 Inspection path with the shortest distance traveled, optimized by the Genetic Algorithm. Panels highlighted in red require re-inspection; the red line shows the calculated route. . 45
IV.7 Inspection path with the shortest time, optimized by Genetic Algorithm. Panels highlighted in red require re-inspection; the red line shows the calculated route. 45

IV.8 UAV path when requesting a support action to the UGV. 46
IV.9 F1 and Precision-recall curve real training dataset tested on real test dataset. 47
IV.10 Real training dataset predicts on real test dataset. 47
IV.11 F1 and Precision-recall curve 75% real and 25% simulated training dataset tested on real test dataset. 48
IV.12 75% real training dataset predict on real test dataset. 48
IV.13 F1 and Precision-recall curve 50% real and 50% simulated training dataset tested on real test dataset. 49
IV.14 50% real and 50% simulated training dataset predict on real test dataset. 49
IV.15 F1 25% real and 75% simulated training dataset tested on real test dataset. 50
IV.16 25% real and 75% simulated training dataset predict on real test dataset. 50
IV.17 F1 and Precision-recall curve simulated training dataset tested on real test dataset. . . . 51
IV.18 Simulated training dataset predict on real test dataset. 51
IV.19 F1 and Precision-recall curve real training dataset tested on simulated test dataset. . . . 52
IV.20 Real training dataset predict on simulated test dataset. 52
IV.21 F1 and Precision-recall curve 75% real and 25% simulated training dataset tested on simulated test dataset. 53
IV.22 75% real and 25% simulated training dataset predict on simulated test dataset. 53
IV.23 F1 and Precision-recall curve 50% real and 50% simulated training dataset tested on simulated test dataset. 54
IV.24 50% real and 50% simulated training dataset predict on simulated test dataset. 54
IV.25 F1 and Precision-recall curve 25% real and 75% simulated training dataset tested on simulated test dataset. 55
IV.26 25% real and 75% simulated training dataset predict on simulated test dataset. 55
IV.27 F1 and Precision-recall curve simulated training dataset tested on real simulated dataset. 56
IV.28 Simulated training dataset predict on simulated test dataset. 56
IV.29 F1 and Precision-recall curve real training dataset tested on 50% real and 50% simulated test dataset. 57
IV.30 Real training dataset predict on 50% real and 50% simulated test dataset. 58
IV.31 F1 and Precision-recall curve 75% real and 25% simulated training dataset tested on 50% real and 50% 50% real and 50% simulated test dataset. 58
IV.32 75% real and 25% simulated training dataset predict on 50% real and 50% simulated test dataset. 59
IV.33 F1 and Precision-recall curve 50% real and 50% simulated training dataset tested on 50% real and 50% simulated test dataset. 59

IV.34 50% real and 50% simulated training dataset predict on 50% real and 50% simulated test dataset. 60
IV.35 F1 and Precision-recall curve 25% real and 75% simulated training dataset tested on 50% real and 50% simulated test dataset. 60
IV.36 25% real and 75% simulated training dataset predict on 50% real and 50% simulated test dataset. 61
IV.37 F1 and Precision-recall curve simulated training dataset tested on 50% real and 50% simulated dataset. 61
IV.38 Simulated training dataset predict on 50% real and 50% simulated test dataset. 62
IV.39 Comparison between detections from a 100% simulated training dataset versus a 75% real training dataset. 64

List of Abbreviations

3D Three-dimensional . . . 16
AGVS Automated Guided Vehicles . . . 20
AUVS Autonomous Underwater Vehicles . . . 20
CNN Convolutional Neural Network . . . 17
GA Genetic Algorithm . . . 17
GCS Ground Control Station . . . 26
IEA International Energy Agency . . . 21
MRGA Multi-Role Goal Assignment . . . 20
PV Photovoltaic . . . 21
ROS Robot Operating System . . . 17
TSP Traveling Salesman Problem . . . 17
UAV Unmanned Aerial Vehicle . . . 15
UGV Unmanned Ground Vehicles . . . 15
USVS Unmanned Surface Vehicles . . . 20
YOLO You Only Look Once . . . 17

Chapter I Introduction

I.1 Initial Considerations and Work Motivation

Unmanned Aerial Vehicle (UAV) is the most adaptable robotic platform. UAVs can be used in a wide range of fields (PINTO et al., 2019), (RAMOS et al., 2022a). Their applications range areas such as surveillance, search and rescue missions, inspection, package delivery, environmental monitoring, and topographic mapping, to name a few (PINTO et al., 2019), (PINTO et al., 2017), (CASTRO et al., 2023), (PUGLIESE; GUERRIERO; MACRINA, 2020), (GREEN et al., 2019), (MANFREDA et al., 2019). However, nowadays, UAVs have seen enhanced performance when deployed in teams or swarms, where multiple vehicles can coordinate to either collaboratively work on the same task or independently handle portions of a larger mission (TAHIR et al., 2019), (RAMOS et al., 2024).

According to Olayiwola et al. (OLAYIWOLA; ELSDEN; DHIMISH, 2024), the renewable energy sector is increasingly leveraging robotics, artificial intelligence, and drones to address challenges that hinder the full potential of renewable energy sources. This trend includes developing intelligent systems for inspecting solar panels to detect faults, as evidenced by various studies in the literature (MORANDO et al., 2022), (RAMÍREZ; CHAPARRO; MÁRQUEZ, 2022). These advanced technologies enable more efficient and accurate identification of issues, minimizing downtime and optimizing maintenance processes in solar farms.

However, integrating diverse robotic systems introduces unique challenges, such as task allocation, coordination, and managing limited battery life (CHERAGHI; SHAHZAD; GRAFFI, 2022). Effective collaboration between heterogeneous robots, such as UAVs and UGVs, can significantly enhance overall performance in various inspection scenarios (SHI et al., 2020). Unmanned Ground Vehicles (UGV) can transport heavy payloads and integrate multiple sensors and actuators, making them suitable for ground-level operations, albeit lacking the elevated view necessary for broader area assessments. Conversely, UAVs excel in providing aerial perspectives, offering a rapid evaluation of large areas from above. On the other hand, UAVs, particularly multirotor types, are constrained by limited payload capacity and short flight durations due to battery restrictions. While fixed-wing UAVs can offer extended flight times and energy efficiency, they are generally less suited for detailed inspections in complex environments, such as solar farms, due to their higher operational complexity and limited hovering capability (NARVÁEZ et al., 2020).

Integrating the strengths of both UGVs and UAVs creates a more robust and adaptable inspection system. UGVs can serve as mobile charging platforms and logistical support units, enhancing the UAVs' operational range and efficiency. This synergy not only improves fault detection in solar panels but also addresses key operational challenges such as efficient task distribution, energy management, and the

coordination of diverse robotic systems, ultimately leading to more reliable and cost-effective renewable energy solutions. As Castro et al. (CASTRO et al., 2024) highlighted, integrating aerial and ground robots requires innovative strategies to enhance coordination and communication, ensuring seamless collaboration between these diverse systems.

UAVs face constraints, including limited battery life and short flight durations. At the same time, UGVs can carry larger payloads, making them ideal for tasks requiring heavy sensors, extensive equipment, and prolonged operational periods, especially when equipped with efficient energy solutions (BERGER et al., 2023). Overcoming the challenges of optimal task distribution, efficient energy management, and safe interactions in dynamic and complex environments is important for the effective operation of robot cooperation. Consequently, there is a pressing need to develop adaptive algorithms capable of dynamically adjusting the cooperation among heterogeneous robot teams, enabling them to respond flexibly to changing conditions and uncertainties inherent in real-world coverage path planning scenarios (CASTRO et al., 2023). These adaptive approaches are crucial for maximizing the potential of multi-robot systems in diverse and unpredictable operational environments.

The use of artificially generated data from simulated environments has emerged as a promising strategy to augment or even replace real-world datasets (RAMOS et al., 2022b). However, the actual benefits of combining synthetic and real images during model training are still under investigation, particularly regarding how such blends affect cross-domain generalization performance (KASHKOLI et al., 2024). Using synthetic data for object detection provides a flexible and scalable approach to train vision-based models, applicable across various fields, including robotics, autonomous navigation, surveillance, and medical diagnostics. The development of effective object detection systems heavily depends on access to large-scale, well-annotated datasets. Yet, collecting this type of data, particularly for domain-specific or rare scenarios, often demands significant time, labor, and financial resources, making simulation-based alternatives increasingly attractive (KASHKOLI et al., 2024) (LIDBERG, 2018).

Three-dimensional (3D) modeling and rendering technologies have had a significant impact on enabling the generation of lifelike synthetic datasets. In the study by Xiang et al. (XIANG; QI; LI, 2019), the authors explored the construction of adversarial 3D data to support robust model training. Inspired by this direction, the method adopted for Davoudi et al (KASHKOLI et al., 2024) advanced 3D simulation tools to replicate human figures and environmental contexts with remarkable visual fidelity, aiming to produce synthetic datasets that mirror real-world complexity and detail. In the field of robotics, synthetic visual data has become increasingly valuable for the development of perception systems, control policies, and autonomous decision-making. Simulated environments enable the generation of diverse scenarios, including edge cases and rare events that may be challenging to capture in the real world. The integration of synthetic data into simulation-to-reality (sim2real) pipelines has proven crucial for accelerating the

deployment of AI-based robotic systems in real-world applications Recent works have demonstrated the effectiveness of using synthetic images and depth maps for enhancing robotic situational awareness, object tracking, and navigation in unstructured environments (RAMOS et al., 2024)(SOUSA et al., 2025)(SILVA et al., 2022).

I.2 Main Contributions

This work proposes a comprehensive framework for collaboration between a UGV and a UAV, specifically designed for inspecting solar panels. In this approach, the UGV transports the UAV to a predefined location within the solar farm. Once positioned, the UAV takes off and ascends to a specified altitude to commence its pre-inspection mission. Utilizing the advanced Convolutional Neural Network (CNN) model, You Only Look Once (YOLO) version 8, the UAV identifies and detects a specific number of solar panels. This detection data is then used to execute an optimization algorithm, such as a Genetic Algorithm (GA), to design an efficient inspection path based on the detected coordinates of the solar panel centroids. The use of GA is motivated by its capacity to solve combinatorial optimization problems, such as the Traveling Salesman Problem (TSP), which closely models the challenge of efficiently connecting inspection waypoints. While alternative techniques such as A*, Dijkstra, or even exact TSP solvers exist, GA provides a good trade-off between computational cost and solution quality, especially in dynamic environments with varying numbers of targets. In this context, the input to the GA consists of the coordinates of the panel centers, enabling adaptive and scalable planning without requiring prior knowledge of the farm layout. The path planning considers various factors, including hover time, energy consumption, and optimal coverage, to ensure thorough inspection while minimizing time and resource use. This process makes the proposed solution highly versatile and applicable to a wide variety of solar farm configurations.

When the UAV's battery level drops below a critical threshold during the mission, it sends a signal to the UGV to relocate to a nearby position, facilitating a safe landing of the UAV. This interaction not only extends the UAV's operational range by providing a mobile recharging and relaunching platform but also minimizes downtime, ensuring a continuous inspection process. By integrating the capabilities of both robots, the framework aims to optimize the inspection efficiency of large-scale solar farms, addressing challenges such as extensive coverage requirements and energy management. Besides, this work presents a systematic analysis of the impact of training data composition on the model's generalization ability, considering different application scenarios (real, simulated, and hybrid). Additionally, the study offers a replicable methodology for creating representative synthetic datasets. It demonstrates, based on both quantitative and qualitative metrics, that a balanced combination of real and simulated data is essential for the success of embedded inspection systems in real-world environments. The framework is assessed by simulations performed in the Robot Operating System (ROS) (Stanford Artificial Intelligence Laboratory

et al., 2018,) along with the Gazebo platform (KOENIG; HOWARD, 2004). The main contributions of this research can be summarized as follows:

1. Developing a collaborative inspection framework for solar panel farms using UGVs and UAVs;
2. Implementing a YOLOv8 algorithm for solar panel detection, followed by a GA to optimize the route inspection, considering factors like hover time and energy consumption;
3. Performing tests in a realistic environment simulation to verify the proposed approach's functionality as a proof of concept;
4. Construction and analysis of different compositions of datasets with real and simulated images, evaluating the direct impact on model generalization;
5. Providing a replicable methodology for creating representative synthetic datasets, applicable to other visual inspection tasks in robotics.

   It is important to emphasize that the scope of this work is centered on the development of an autonomous UAV-UGV collaborative inspection framework, rather than on the evaluation of the operational or physical conditions of photovoltaic panels. The inspection task addressed in this study focuses on the detection, localization, and coverage planning of solar panels to support efficient and autonomous inspection routines. Issues related to panel health assessment, such as fault diagnosis, thermal anomaly detection, electrical performance analysis, or degradation monitoring, are intentionally not considered, as they would require additional sensing modalities and specialized diagnostic models.

I.3 Organization

The remainder of this dissertation is organized as follows:

- **Chapter II**: This chapter reviews the state-of-the-art in the fields of heterogeneous multi-robot cooperation and UAV-based inspection of PV farms. It discusses key challenges in robotic coordination, energy management, and perception systems, and provides a survey of relevant works that support the motivation and design choices of the proposed framework.
- **Chapter III**: This chapter presents the architecture of the proposed collaborative framework UAV-UGV. It describes the hardware setup, the hierarchical cooperative structure, the integration of the YOLOv8 convolutional neural network for solar panel detection, the genetic algorithm for path optimization, and the visual inspection routine. It also includes the design and generation of synthetic datasets used for training and evaluating the detection model.

- **Chapter IV**: This chapter presents and analyzes the experimental results obtained through simulations in the ROSGazebo environment. It includes performance metrics of YOLOv8 trained on different combinations of real and synthetic data and evaluates the detection generalization across real, simulated, and hybrid datasets. It also discusses the efficiency and autonomy of the UAV-UGV interaction during the inspection routine.
- **Chapter V**: This chapter summarizes the main findings of the research, highlighting the relevance of collaborative robotic architectures and balanced dataset composition for cross-domain generalization. It also outlines proposals for future developments, including real-world testing, multi-agent coordination, and the integration of thermal anomaly detection.

Chapter II Related Works

II.1 Heterogeneous Robots Cooperation

The collaboration among robotic systems has gained significant attention in recent decades, posing challenges to robot collaboration. This collaboration requires effective coordination of multiple robots to accomplish numerous tasks while adhering to specific constraints (CHAKRAA et al., 2023). To optimize the performance of a group of heterogeneous robots, it is essential to develop methods that characterize and exploit the robots' distinct capabilities in their strategy. Different researchers have addressed this issue (MAYYA et al., 2021), (HUANG; ZHANG; XIAO, 2019), (PROROK; HSIEH; KUMAR, 2017). For instance, in Lee et al. (LEE et al., 2023), the authors proposed a distributed task allocation framework based on the capability matching of heterogeneous robots, where the framework is composed of an ontological dynamic knowledge graph model and a hardware control scheme to model the capability and optimize resource utilization for collaborative tasks.

In the literature, several collaboration propositions were proposed (CHEN et al., 2023), (LI et al., 2023). These tasks range in various fields, involving different types of robots, such as Automated Guided Vehicles (AGVs) and UGVs on land, and in the air, airplanes, blimps, and UAVs. Additionally, water-based robots, such as Autonomous Underwater Vehicles (AUVs) and Unmanned Surface Vehicles (USVs), form another category of autonomous systems. This study focuses on the collaboration between a single UGV and UAV for inspecting solar panel farms. It is worth noting that this framework could be expanded to include multiple UAVs and additional UGVs, further enhancing the inspection process.

Different strategies have been developed to enable collaborative tasks with heterogeneous robots. For instance, Carreno et al. (CARRENO et al., 2020) proposed a novel task allocation strategy called Multi-Role Goal Assignment (MRGA) for heterogeneous robots in offshore missions. Their proposition was designed to improve the efficiency of multi-robot planning for dynamic, resource-constrained tasks.

Roldan et al. (ROLDÁN et al., 2016) developed a collaborative strategy to monitor environmental variables in greenhouses. The robots' objectives were to collect data on crucial factors like temperature, humidity, luminosity, and carbon dioxide levels to help maintain optimal conditions for plant growth. They used both ground and aerial robots, where the ground units offer robustness and longer operational capacity, and the aerial units provide agility and access to areas beyond the ground robots' reach.

In the work of Berger et al. (BERGER et al., 2023), the authors proposed a collaboration of UGV and UAV for the inspection process. One of the novelties of their proposition was the landing processing of the UAV into the UGV for the hot-swapping process. This allowed for extended operational time without manual intervention, showcasing how integrating different robot platforms can improve mission endurance and efficiency.

Similarly, Castro et al. (CASTRO et al., 2024) proposed two UAVs and one UGV work cooperatively to perform an inspection mission. The robots operate to perform an inspection mission. The tested scenario was an electrical substation, and the robots were posed with the challenges of partially known terrains and potential dynamic obstacles.

II.2 UAVs Applied to Solar Panel Farms Inspection

As highlighted by Meribout et al. (MERIBOUT et al., 2023), the depletion of global nonrenewable resources is driving an accelerated shift toward renewable energy sources, which are expected to dominate major electricity markets in the coming decades (Mornsun Power, 2023). According to forecasts by the International Energy Agency (IEA), the share of renewable energy in global power generation is projected to rise to 30% by 2030 and 50% by 2040, with Photovoltaic (PV) power accounting for approximately 10% and 20% of that share, respectively.

However, despite the promising growth of large-scale PV systems, challenges such as effective localization, real-time fault detection, and diagnostics during field operations persist. These issues have prompted ongoing research into advanced inspection techniques, as reviewed by Abubakar et al. (ABUBAKAR; ALMEIDA; GEMIGNANI, 2021) and Yahya et al. (YAHYA et al., 2022).

As highlighted by Pruthviraj et al. (PRUTHVIRAJ et al., 2023), UAVs are revolutionizing the solar energy industry by enabling cost-effective and efficient methods for detecting hotspots on solar panels, mapping solar farms, and inspecting facilities for issues like thermal bridging and corrosion. In a related study, Zefri et al. (ZEFRI et al., 2018) utilized UAVs to inspect PV installations through a combination of RGB and thermal infrared imagery, offering a comprehensive approach to fault detection. RGB imagery has proven effective in identifying defects such as cracks, discoloration, soiling, delamination, and corrosion through visual inspection (GRIMACCIA; LEVA; NICCOLAI, 2017), (AGHAEI et al., 2016). These visual monitoring techniques are key to maintaining solar panel performance and extending operational life.

In addition to UAVs, other types of robots have been applied to solar power plant inspection and maintenance. For instance, Felsch et al. (FELSCH et al., 2015) introduced a climbing robot specifically designed for inspecting and maintaining vertical structures within solar power plants. Similarly, Venkatnikhil et al. (VENKATNIKHIL; RAVICHANDRAN; KUMAR, 2022) developed a rover robot capable of cleaning and monitoring solar panels, highlighting the growing role of ground-based robots in solar farm maintenance.

These diverse robotic systems demonstrate the potential for integrating both aerial and ground platforms to enhance the efficiency and thoroughness of solar farm operations, addressing a range of maintenance challenges in the renewable energy sector.

II.3 Discussions

A comprehensive review of the literature reveals that a variety of strategies have been developed to enable cooperation among heterogeneous robotic platforms, ranging from distributed task allocation frameworks to multi-robot inspection systems in structured and unstructured environments. However, despite the diversity of approaches, most existing works focus either on task allocation, environmental monitoring, or single-modality inspection, without offering a fully integrated solution that combines perception, planning, and collaboration for PV inspection.

Table II.1 presents a detailed comparison of the main related studies in the domains of heterogeneous robot cooperation and UAV-based inspection. As shown, several works explore UAV–UGV cooperation (ROLDÁN et al., 2016; BERGER et al., 2023; CASTRO et al., 2024), while others focus exclusively on aerial inspection of PV installations (ZEFRI et al., 2018; PRUTHVIRAJ et al., 2023). Although these studies contribute important advances, such as hot-swapping mechanisms, environmental data collection, or the use of thermal and RGB imaging, none of them integrates the full pipeline required for adaptive, large-scale solar farm inspection.

More specifically, the reviewed works do not combine (i) collaborative UAV–UGV deployment, (ii) automatic solar panel detection using deep learning, and (iii) dynamic route optimization grounded on the detected panel coordinates. Moreover, none of the existing approaches addresses the operational interdependence between aerial and ground robots in terms of landing, recharging, and relaunching during a mission, which is essential for extended inspection coverage in large PV fields.

In contrast, the proposed framework offers a unified and extensible solution that integrates computer vision, combinatorial optimization, and multi-robot cooperation. By coupling a UGV with a UAV and employing YOLOv8 for real-time panel detection, followed by a Genetic Algorithm for inspection route optimization, this work addresses key limitations in the current state of the art. Additionally, the evaluation of real, simulated, and hybrid datasets further distinguishes the proposed system by highlighting its robustness and adaptability to diverse inspection scenarios.

Overall, the comparison in Table II.1 shows the novelty and practical relevance of the proposed approach, which fills critical gaps in existing research by delivering an end-to-end framework for autonomous and scalable PV inspection.

Table II.1: Comparison of related works and the proposed approach.

Reference	**Robotic Platform**	**Application Domain**	**Main Contributions / Limitations**
Lee et al. (2023) (LEE et al., 2023)	Heterogeneous ground and aerial robots	Distributed task allocation	Ontology-based capability matching. Does not address solar inspection nor adaptive vision-based planning.
Carreno et al. (2020) (CARRENO et al., 2020)	Heterogeneous robots (offshore missions)	Task allocation in dynamic environments	Focus on resource-constrained offshore operations. No UAV–UGV coordination for inspection or vision-based tasks.
Roldan et al. (2016) (ROLDÁN et al., 2016)	UAV + UGV	Greenhouse environmental monitoring	Environmental data collection. No object detection pipeline or optimized inspection path.
Berger et al. (2023) (BERGER et al., 2023)	UAV + UGV	General inspection missions	Introduces UAV landing on UGV for hot-swapping. Lacks solar panel detection and route optimization.
Castro et al. (2024) (CASTRO et al., 2024)	Two UAVs + one UGV	Electrical substation inspection	Handles partially known terrain and dynamic obstacles. Does not present a full detectionoptimizationexecution pipeline.
Felsch et al. (2015) (FELSCH et al., 2015)	Climbing robot	Vertical structure inspection	Application-specific; does not involve multi-robot cooperation.
Venkatnikhil et al. (2022) (VENKATNIKHIL; RAVICHANDRAN; KUMAR, 2022)	UGV	Cleaning and monitoring of PV panels	Ground robot only; no aerial coverage or global inspection planning.
Zefri et al. (2018) (ZEFRI et al., 2018), Pruthviraj et al. (2023) (PRUTHVIRAJ et al., 2023)	UAV	Solar panel inspection with RGB/TIR sensors	RGB/TIR-based fault detection. Do not integrate UGV support or path optimization.
Proposed Work	**UGV + UAV**	**Solar farm inspection**	**(i) Collaborative UGV–UAV framework; (ii) YOLOv8-based solar panel detection; (iii) GA for adaptive route optimization; (iv) UGV repositioning for landing and energy management; (v) Analysis of real/synthetic datasets**

This comparison reveals a fragmented scenario in which existing approaches address isolated aspects of robotic cooperation, such as capability matching, environmental monitoring, and visual inspection, but rarely integrate perception, planning, and execution cohesively. By explicitly coupling a deep-learning detection method with UAV path optimization and UGV-assisted energy management, the proposed system advances the state of the art toward a more autonomous and scalable solution. This integrated perspective not only improves the reliability of inspection missions but also establishes a new perspective for future extensions involving larger robotic teams, multi-modal sensing, and dynamic task reassignment in real solar farm deployments.

Chapter III Proposed Methodology

This work aims to present an autonomous solution for inspecting solar panel farm facilities through the collaboration of a UAV and a UGV. The proposed system leverages the strengths of both robots to optimize the inspection process. Figure III.1 provides a flowchart illustrating the overall concept of the proposed approach.

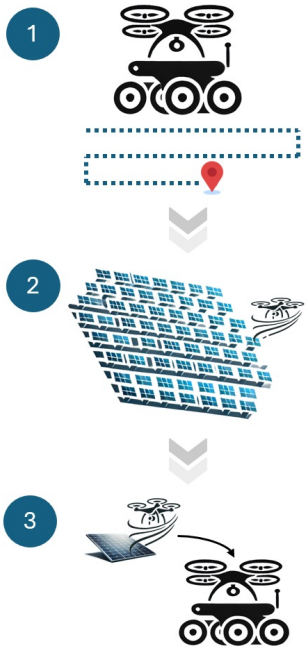


Figure III.1: Flowchart of the general idea. (1) UGV navigates to a predetermined location within the solar farm. (2) UAV is deployed to perform aerial inspections. (3) After completing the inspection or when the UAV's battery level reaches a critical threshold, the UAV signals the UGV to prepare for landing. The UGV moves to a suitable position to facilitate a safe landing for the UAV.

The process begins with the UGV transporting the UAV to a designated location within the solar farm. Once the predefined position is reached, the UAV takes off and ascends to a specific altitude to perform its inspection task. At this stage, the UAV uses the YOLOv8 CNN to detect the location of the solar panels. The coordinates of the detected centroids are then used to generate an optimized inspection path. Although the UAV is capable of collecting visual data that could support condition assessment, this work focuses specifically on the autonomous inspection strategy and path planning rather than the diagnostic evaluation of panel conditions. The inspection data gathered by the UAV is then processed through the optimization algorithm GA, which plans an optimal inspection route. This path planning considers hover time, energy consumption, and coverage efficiency to maximize the UAV's inspection capabilities while minimizing resource usage.

In the event of low battery levels, the UAV communicates with the UGV, which then relocates to a

nearby position to allow a safe landing and recharging. While continuous tracking of the UAV by the UGV could potentially reduce repositioning time, this work adopts a reactive relocation strategy to minimize unnecessary UGV movement and preserve ground energy resources. The chosen approach strikes a balance between responsiveness and energy efficiency in large-scale environments. This coordinated operation between the UAV and UGV ensures continuous inspection while addressing key challenges such as energy management and extensive coverage of the solar farm. The final step involves analyzing the inspection data and generating reports on the status of the solar panels.

III.1 Hardware Setup

In the proposed framework, the Ground Control Station (GCS) coordinates the UGV and UAV operations. The GCS has 16GB of RAM and is powered by a 2.6 GHz Core i7-6700HQ processor and an NVIDIA GeForce GTX960m with 4GB of VRAM. The chosen operating system distribution was a 64-bit Ubuntu 20.04. All algorithms were coded using Python 3.8.10. The employed UAV is the Iris model (3D Robotics, 2014), while the UGV is the Nexus robot. Communication between the GCS, the UAV, and the UGV is facilitated through a telemetry module installed on both the computer and the vehicles, utilizing the MAVLink communication protocol (MAVLink Contributors, 2022). This protocol ensures efficient and reliable data exchange between all system components. The PX4 software, which runs on the Pixhawk flight controller hardware, manages the UAV's flight operations. PX4 processes data from the UAV's peripheral sensors, including GPS, IMU, and cameras, and adjusts actuator outputs for stable and precise flight. This setup enables the UAV to perform aerial inspections of solar panels while maintaining synchronization with the UGV. Fig. III.2 presents both robots in the simulated solar farm.

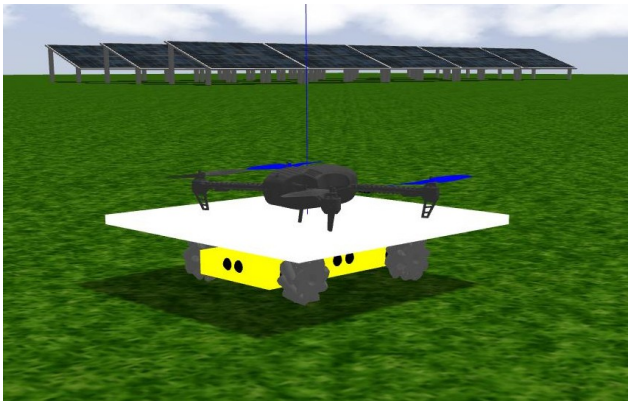


Figure III.2: Simulated Solar Panel Farm and the robots used in the simulation. In this context, the UGV can be seen as a mobile "drone-in-a-box" platform, enabling similar functionalities in environments where static docking stations are not viable.

The ROS Master plays an important role in managing the nodes within the ROS ecosystem, facilitating communication among the subsystems of each robotic agent. ROS employs a publish-subscribe and service-oriented architecture to exchange data and information between various components.

A core element within this framework is the Node, which refers to individual software modules responsible for specific tasks. These nodes form the building blocks of a robotic system, and they communicate by sending and receiving messages through a structure known as topics. Fig. III.3 visually represents the key nodes, topics, and their interactions during the simulation process.

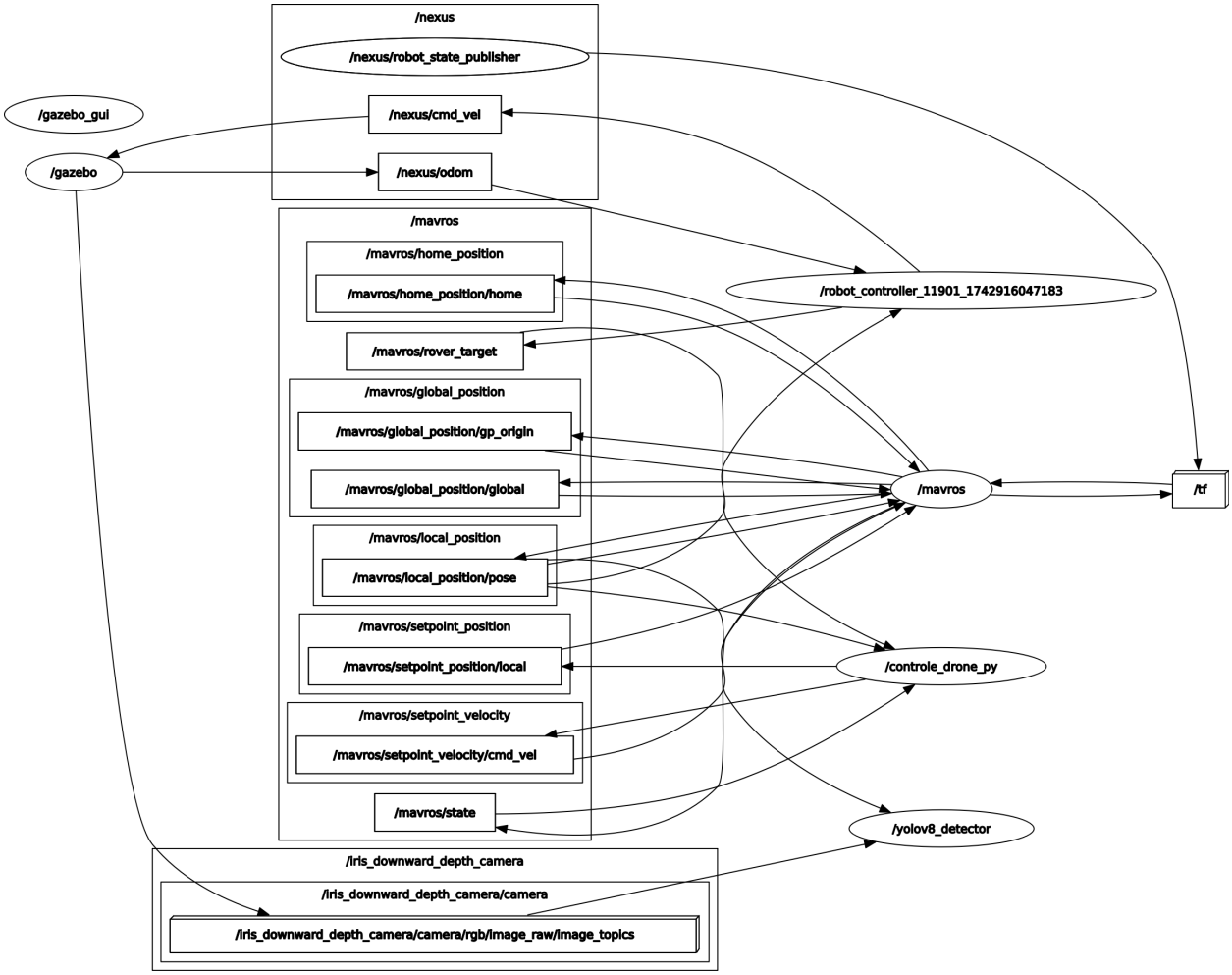


Figure III.3: ROS active nodes and messages.

III.2 Cooperative Framework

The cooperative framework is organized into a hierarchical structure consisting of three interconnected layers: the high, middle, and low, as illustrated in Fig. III.4. The proposed framework ensures real-time processing and efficient communication between layers through ROS topics and services.

The high layer focuses on mission-level management and decision-making, encompassing tasks such as planning inspection operations, assigning objectives, and overseeing coordination between the UAV and UGV. This layer employs a human-in-the-loop approach, enabling human operators to intervene or adjust mission parameters as needed, thereby ensuring safety and flexibility in dynamic environments.

The middle layer serves as the processing hub for sensor data and planning optimized actions. It utilizes advanced algorithms, such as YOLOv8 for detecting and assessing solar panels and the Genetic

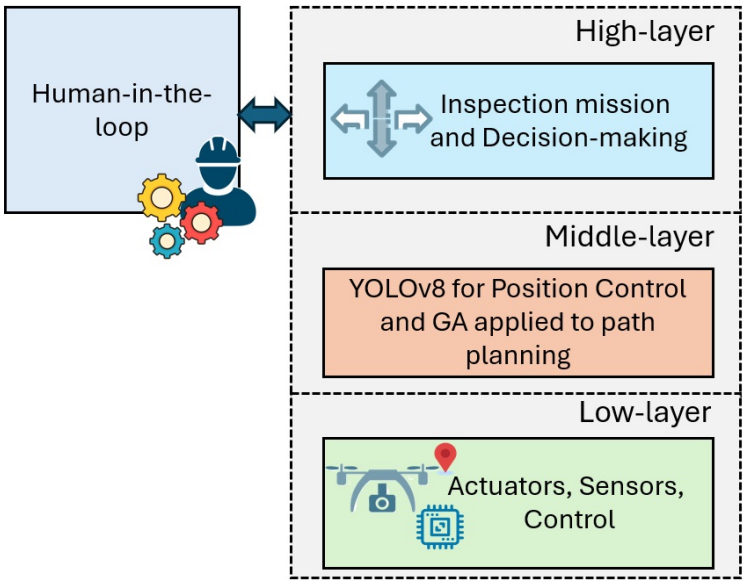


Figure III.4: Cooperation framework.

Algorithm for efficient path planning. This layer ensures mission-level decisions and hardware execution, allowing the cooperative operation between the UAV and UGV.

The low layer focuses on hardware-level operations, executing commands from the middle layer, and managing the system's physical components. This includes operating actuators for UAV flight and UGV mobility and integrating sensors such as GPS, IMU, and cameras to ensure precise navigation and inspection. By maintaining synchronization between the UAV and UGV, the low layer enables the seamless execution of tasks within the cooperative framework.

To provide a more precise visualization of the interaction flow between the UAV and UGV during the inspection mission, Fig. III.5 illustrates the detailed communication flowchart. This diagram illustrates the information exchange between the agents and the GCS, encompassing command dispatch, battery level monitoring, relocation triggers, and the cooperative decision-making process that facilitates seamless mission continuity. The flowchart is structured into three phases: mission initialization, in-flight monitoring, and emergency or low-battery handling. Initially, the GCS dispatches the inspection mission to the UGV, which navigates to the designated area and subsequently deploys the UAV. Once airborne, the UAV performs an initial scan to detect the positions of the solar panels using onboard YOLOv8-based detection algorithms. These detected positions are then used in the mission planning stage, where an optimized inspection path is generated. Only after this planning step does the UAV begin executing the detailed inspection mission. During the mission, telemetry data, including GPS position, battery level, and inspection progress, are continuously streamed back to the GCS. If the UAV's battery level falls below a predefined safety threshold, a decision node triggers a return-to-UGV request. This event activates a subroutine in which the GCS repositions the UGV based on the UAV's location and the estimated remaining flight time. The UGV then moves to a calculated safe landing position. Upon

successful docking, the UAV is recharged, and the mission can either resume or terminate based on the inspection coverage status.

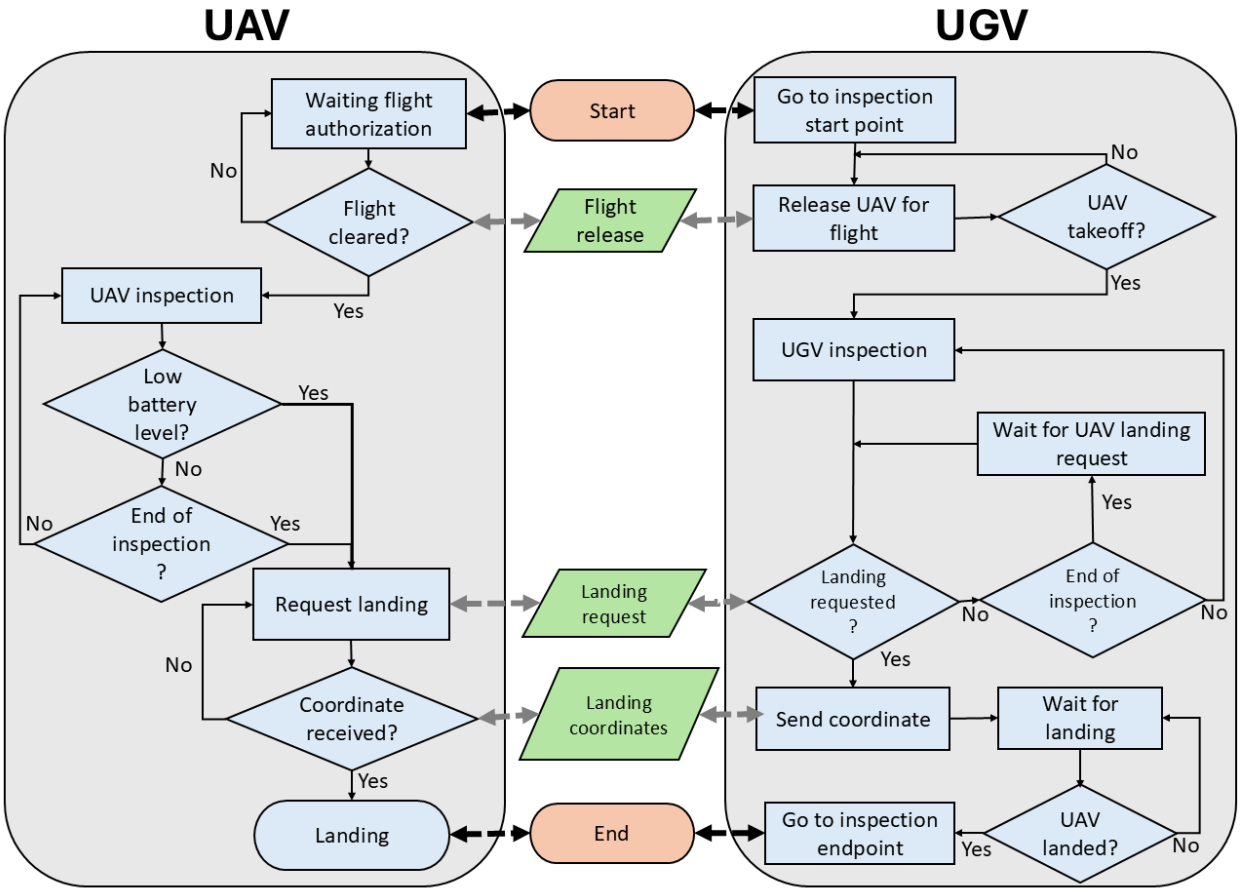


Figure III.5: UAV and UGV cooperation flowchart.

The communication model is fully implemented using the MAVLink protocol to ensure lightweight, low-latency, and robust message passing. This architecture also supports modular expansion, allowing for future integration of additional UAVs or UGVs without significant changes to the logic structure.

III.3 Yolov8 Neural Network Structure

YOLOv8 is a cutting-edge object detection and classification algorithm integrated into the proposed inspection framework, ensuring real-time and highly accurate detection of solar panels. The YOLOv8 architecture is composed of a backbone, neck, and head, as depicted in Fig. III.6. The backbone is responsible for feature extraction, employing advanced convolutional layers to identify patterns in the input images. The neck aggregates these features to create a feature pyramid, improving object detection at different scales. Finally, the head performs bounding box regression and object classification, outputting the location and condition of the detected solar panels.

To adapt YOLOv8 for solar panel detection, the model was fine-tuned on the dataset provided

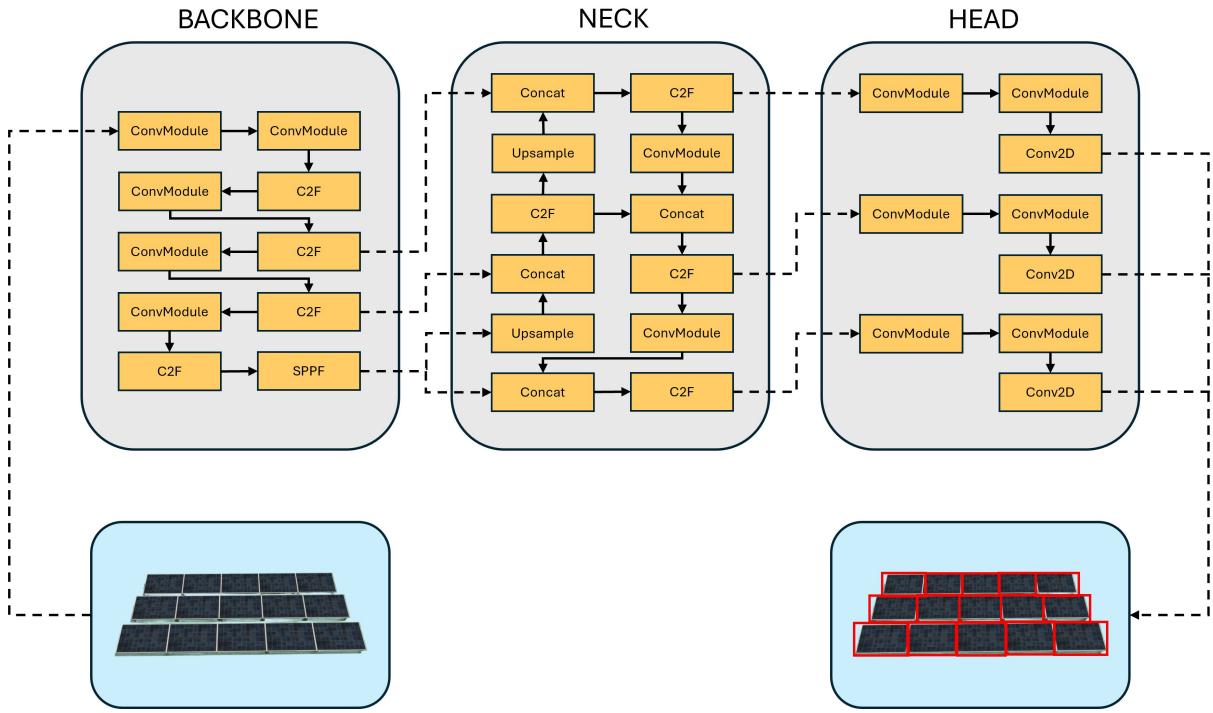


Figure III.6: YOLOv8 Structure.

in (ISHII, 2023). This dataset consists of high-resolution aerial images captured by the UAV, each annotated with bounding boxes around solar panels. These bounding boxes serve as the labels used during training. Additional simulated images were incorporated to improve generalization. Preprocessing techniques were also applied to enhance the model's robustness. Fig. III.7 illustrates some images used to train and validate the YOLO model.

In the proposed framework, real-time images captured by the UAV are transmitted to the GCS, where YOLOv8 operates to detect solar panels. This centralized processing approach uses the computational resources of the GCS, reducing the UAV's onboard processing load. The high inference speed of YOLOv8 ensures that the inspection process is efficient, allowing the UAV to cover large areas quickly.

III.4 Inspection Routine

The inspection routine is a coordinated process that integrates YOLOv8 for object detection and a GA for path optimization. The GA algorithm is a central point in optimizing the UAV's flight path during the inspection. The GA is a metaheuristic inspired by the process of natural selection and is particularly effective for solving complex optimization problems (MIRJALILI; MIRJALILI, 2019). In this application, the GA iteratively evaluates flight paths to identify the most efficient route for the UAV,



Figure III.7: Dataset images from (ISHII, 2023) and (SILVA, 2025).

considering flying time, hover time (in case of detecting an object of interest), and coverage path.

The inspection begins by pre-inspecting all solar panels to identify potential points of failure, dirt, or damage. After identifying any anomalies in the solar panels, the coordinates of the center point of each solar panel are saved and grouped into a set of coordinates for further inspection using UAVs equipped with other sensors and cameras to identify and classify the defects.

III.4.1 Mathematical Formulation of the Optimization Problem

The optimization problem addressed by the GA aims to determine the most efficient inspection route for the UAV, visiting all detected solar panels while minimizing both travel distance and total inspection time. Let $\pi = [p_1, p_2, \ldots, p_n]$ denote a permutation of the n inspection targets, where each p_i represents the centroid coordinates of a solar panel detected by the YOLOv8 network.

The total inspection cost function is defined as a weighted combination of the total distance $D(\pi)$ and total time $T(\pi)$:

$$\text{Minimize} \quad f(\pi) = \alpha \cdot D(\pi) + \beta \cdot T(\pi), \tag{III.1}$$

where α and β are weighting coefficients that balance the trade-off between energy efficiency and

mission duration, with $\alpha + \beta = 1$.

The total distance $D(\pi)$ can be computed as:

$$D(\pi) = \sum_{i=1}^{n-1} d(p_i, p_{i+1}), \tag{III.2}$$

where $d(p_i, p_{i+1})$ represents the Euclidean distance between consecutive inspection targets:

$$d(p_i, p_{i+1}) = \sqrt{(x_{p_{i+1}} - x_{p_i})^2 + (y_{p_{i+1}} - y_{p_i})^2}. \tag{III.3}$$

Similarly, the total time $T(\pi)$ is expressed as the sum of flight and hover times:

$$T(\pi) = \sum_{i=1}^{n-1} \frac{d(p_i, p_{i+1})}{v_{\text{uav}}} + \sum_{i=1}^{n} t_{h,i}, \tag{III.4}$$

Where v_{uav} is the UAV's average cruising speed, and $t_{h,i}$ denotes the hover duration over target p_i, which depends on detection confidence and inspection requirements.

Thus, the multi-objective optimization problem can be summarized as:

$$\text{Minimize:} \quad f(\pi) = \alpha \cdot D(\pi) + \beta \cdot T(\pi), \tag{III.5}$$

$$\text{Subject to:} \quad \pi \in \mathcal{P}(V), \tag{III.6}$$

$$p_i \neq p_j, \quad \forall i \neq j, \tag{III.7}$$

Where $\mathcal{P}(V)$ is the set of all possible permutations of the inspection targets. The problem is NP-hard and analogous to the TSP. Hence, a metaheuristic approach such as the Genetic Algorithm is adopted to obtain near-optimal solutions within an acceptable computational time.

III.4.2 Mathematical Formulation to extract coordinates during inspection steps

The coordinates of the centers of the solar panels are obtained from the drone's position and the distances between the centers of the bounding boxes and the center of the image. Since the distances in the image are measured in pixels, it is necessary to convert pixel displacements to meters. To achieve this, additional odometry information from the drone and the intrinsic parameters of the camera used during the pre-inspection are required.

The required parameters are:

- Drone – Positions (X, Y, Z) and orientations (Roll, Pitch, Yaw);
- Camera – Image width, height, and focal lengths fx and fy.

With these parameters, we can formulate the equations to obtain the coordinates, starting with the simplest scenario, i.e., the drone without any inclination (Fig. III.8), and then expanding the model to account for any roll and pitch angles (Fig. III.9).

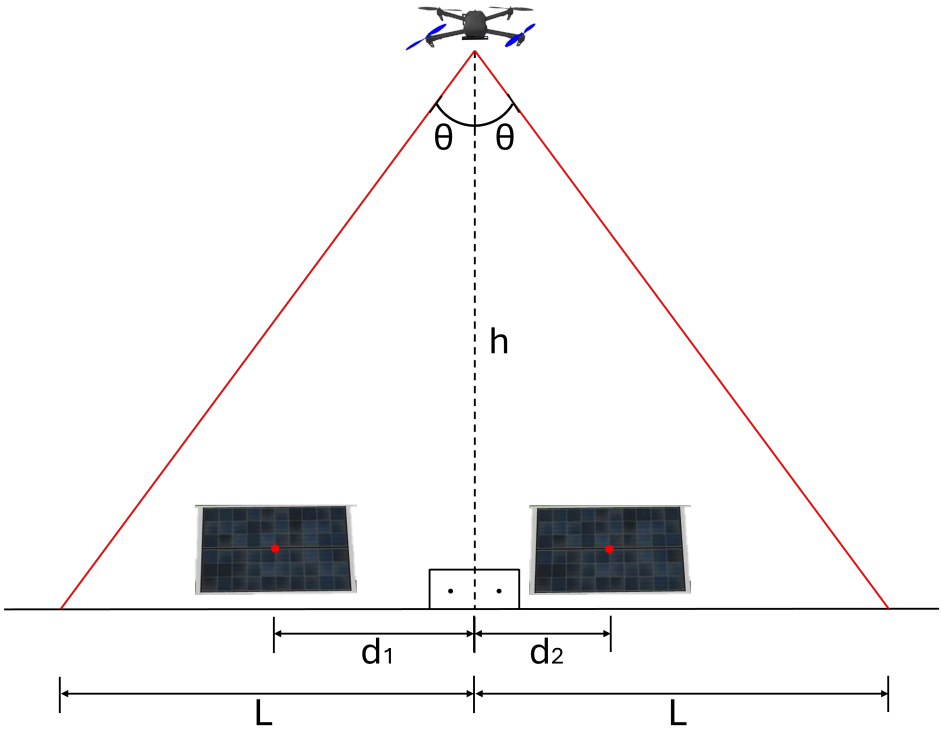


Figure III.8: 2D illustration considering roll or pitch inclination equal to zero.

For Figure III.8, we have:

$$\Phi = 2\arctan(\frac{T_P}{2f}) \tag{III.8}$$

Where Φ is the field of view (FOV), T_P the total number of pixels, and f the focal length.

$$\theta = \frac{\Phi}{2} \quad [deg] \tag{III.9}$$

$$h = D_z \quad [m] \tag{III.10}$$

$$L = h\tan(\theta) \quad [m] \tag{III.11}$$

$$P_P = \frac{b_{min} + b_{max}}{2} \quad [px] \tag{III.12}$$

$$C_P = \frac{T_P}{2} \quad [px] \tag{III.13}$$

Where D_z is the z coordinate (height) of the drone, P_P the center pixel of the panel, b_{min} and b_{max} are the bounding box limits, and C_P is the central pixel of the camera image.

To find the distance from the center of the panel to the center of the image, the following is performed:

$$d_1 = C_P - P_{P1} \quad [px] \tag{III.14}$$

$$d_2 = C_P - P_{P2} \quad [px] \tag{III.15}$$

$$P_1 = \frac{L}{f} d_1 + D_C \quad [m] \tag{III.16}$$

$$P_2 = \frac{L}{f} d_2 + D_C \quad [m] \tag{III.17}$$

Where P_{P1} and P_{P2} are the center pixels of panel 1 and panel 2 respectively, and D_C the drone coordinate.

For Fig. III.9, considering calculations for the camera's x and y axes with roll and pitch, respectively, since the image is rectangular (width greater than height), the vertical FOV (Φ_y) and horizontal FOV (Φ_x) are different. In addition, it must also account for the orientations of the drone's axes in the world frame and the camera's width and height axes, as illustrated in Fig. III.10.

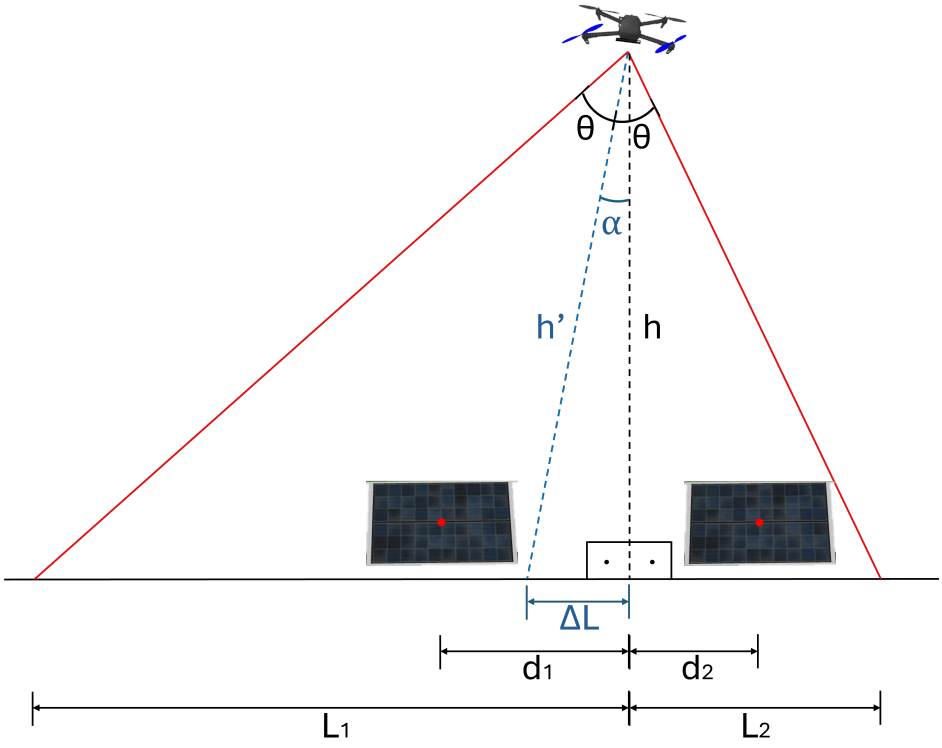


Figure III.9: 2D illustration considering a non-zero roll or pitch inclination.

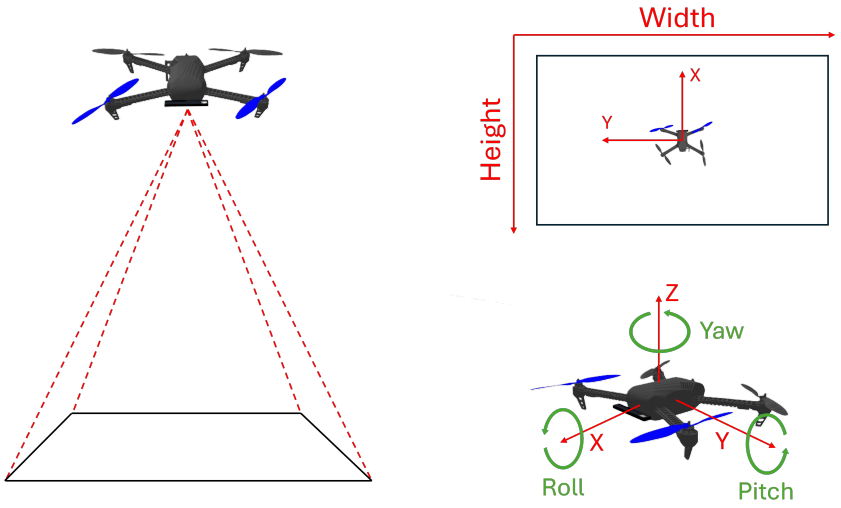


Figure III.10: Relation between the x and y axes, roll and pitch of the drone, with the width and height of the image captured by the camera.

The relation between the drone's x and y axes in the world frame and the image's pixel width and height coordinates can be expressed as follows:

$$Origin : x_0 = \frac{height}{2} \quad [px] \tag{III.18}$$

$$Pitch > 0 \Longrightarrow x_0' = x_0 + \Delta height \quad [px] \tag{III.19}$$

$$Pitch < 0 \Longrightarrow x_0' = x_0 - \Delta height \quad [px] \tag{III.20}$$

$$Origin : y_0 = \frac{width}{2} \quad [px] \tag{III.21}$$

$$Roll > 0 \Longrightarrow y_0' = y_0 - \Delta width \quad [px] \tag{III.22}$$

$$Roll < 0 \Longrightarrow y_0' = y_0 + \Delta width \quad [px] \tag{III.23}$$

In this way, we can verify that the sign indicating the direction of the displacement $\Delta height$ of the image center follows $\tan(Pitch)$, while the sign indicating the direction of the displacement $\Delta width$ follows $-\tan(Roll)$. Therefore, to compensate for the displacement of the captured image center relative to the drone's world coordinates (x, y), when Pitch $\neq$ 0, the displacement generated by $\tan(Pitch)$ must be subtracted from the x coordinate; and when Roll $\neq$ 0, the displacement generated by $\tan(Roll)$ must be added to the y coordinate.

With this, we can perform the calculations for both cases, taking into account the specific axes, rotations, and camera parameters of each configuration.

Case 1: For the horizontal axis of the image, we have $\Phi = \Phi_x$, T_P = width, $f = f_x$, α = roll.

$$\Phi_x = 2\arctan(\frac{width}{2\ f_x}) \quad [deg] \tag{III.24}$$

$$\theta = \frac{\Phi_x}{2} \quad [deg] \tag{III.25}$$

$$h = D_z \quad [m] \tag{III.26}$$

$$P_P = \frac{b_{min} + b_{max}}{2} \quad [px] \tag{III.27}$$

$$C_P = \frac{width}{2} \quad [px] \tag{III.28}$$

$$\Delta L = h\tan(\alpha) \quad [m] \tag{III.29}$$

$$L_1 = h\tan(\theta + \alpha) \quad [m] \tag{III.30}$$

$$L_2 = h\tan(\theta - \alpha) \quad [m] \tag{III.31}$$

$$L_1' = L_1 - \Delta L = h(\tan(\theta + \alpha) - \tan(\alpha)) \quad [m] \tag{III.32}$$

$$L_2' = L_2 + \Delta L = h(\tan(\theta - \alpha) + \tan(\alpha)) \quad [m] \tag{III.33}$$

Where L_1 and L_2 are the distances in meters, along the width axis, from the image boundaries to the drone's coordinate system (axis perpendicular to the plane), and L_1' and L_2' are the distances in meters from the image boundaries to the image center.

To obtain the panel's distance in meters along the width axis relative to the image center, which is displaced by ΔL from the drone's coordinate, we have:

$$d_1 = C_P - P_{P1} \quad [px] \tag{III.34}$$

$$d_2 = C_P - P_{P2} \quad [px] \tag{III.35}$$

$$P_{W1} = \frac{L_1'}{fx} d_1 \ + \ \Delta L \quad [m] \tag{III.36}$$

$$P_{W2} = \frac{L_2'}{fx} d_2 \ + \ \Delta L \quad [m] \tag{III.37}$$

Where P_{W1} and P_{W2} are the distances in meters along the width axis of the center pixels of panel 1 and panel 2, respectively, to the drone's coordinate system (axis perpendicular to the plane).

Case 2: For the vertical axis of the image, we have $\Phi = \Phi_y$, $T_P =$ height, $f = f_y$, $\alpha =$ pitch.

$$\Phi_y = 2\arctan(\frac{height}{2\ fy}) \quad [deg] \tag{III.38}$$

$$\theta = \frac{\Phi_y}{2} \quad [deg] \tag{III.39}$$

$$h = D_z \quad [m] \tag{III.40}$$

$$P_P = \frac{b_{min} + b_{max}}{2} \quad [px] \tag{III.41}$$

$$C_P = \frac{height}{2} \quad [px] \tag{III.42}$$

$$\Delta L = h\tan(\alpha) \quad [m] \tag{III.43}$$

$$L_1 = h\tan(\theta + \alpha) \quad [m] \tag{III.44}$$

$$L_2 = h\tan(\theta - \alpha) \quad [m] \tag{III.45}$$

$$L_1' = L_1 - \Delta L = h(\tan(\theta + \alpha) - \tan(\alpha)) \quad [m] \tag{III.46}$$

$$L_2' = L_2 + \Delta L = h(\tan(\theta - \alpha) + \tan(\alpha)) \quad [m] \tag{III.47}$$

Where L_1 and L_2 are the distances in meters, along the height axis, from the image boundaries to the drone's coordinate system (axis perpendicular to the plane), and L_1' and L_2' are the distances in meters from the image boundaries to the image center.

To obtain the panel's distance in meters along the height axis relative to the image center, which is displaced by ΔL from the drone's coordinate, we have:

$$d_1 = C_P - P_{P1} \quad [px] \tag{III.48}$$

$$d_2 = C_P - P_{P2} \quad [px] \tag{III.49}$$

$$P_{H1} = \frac{L_1'}{fy} d_1 \ - \ \Delta L \quad [m] \tag{III.50}$$

$$P_{H2} = \frac{L_2'}{fy} d_2 \ - \ \Delta L \quad [m] \tag{III.51}$$

Where P_{H1} and P_{H2} are the distances in meters along the height axis of the center pixels of panel 1 and panel 2, respectively, to the drone's coordinate system (axis perpendicular to the plane).

Finally, with the distances in meters corrected by ΔL, we can rotate them with respect to the world frame using the drone's Yaw angle, and then add them to the drone's actual world coordinates, thus obtaining the panel's x and y coordinates in the world.

$$D_H = P_H \cos(Yaw) - P_W \sin(Yaw) \quad [m] \tag{III.52}$$

$$D_W = P_H \sin(Yaw) + P_W \cos(Yaw) \quad [m] \tag{III.53}$$

$$P_x = D_x + D_H \quad [m] \tag{III.54}$$

$$P_y = D_y + D_W \quad [m] \tag{III.55}$$

Where D_H and D_W are the distances in meters along the height axis and the width axis of the image, respectively, from the center pixels of the panel to the drone's coordinate system (axis perpendicular to the plane) rotated by the yaw angle relative to the world frame. Finally, P_x and P_y are the coordinates of the solar panel referenced to the world frame from D_x and D_y, which are the coordinates of the drone.

III.5 Impact of Real and Simulated Data Composition

Analyzing the composition between synthetic images extracted from the simulated environment and images from real environments during detector training is essential to ensure the robustness of perception, both in the digital twin and in onboard operation in UAVs used in real environments. While the simulated environment allows for the generation of large volumes of data by controlling lighting, texture, and geometry in different scenarios, Gazebo's simulated environment still generates visual discrepancies compared to real environments that can compromise model training performance. Examples of these discrepancies include optical noise, structural irregularities, and other uncontrollable variables present in real-world environments that are often difficult to represent in a simulated environment.

With real-world images, we can capture these environmental variations more accurately, but acquiring real-world images is more expensive in terms of time and equipment, and also offers less scalability for composing a dataset. Therefore, understanding the combined effect of real and simulated images on the learning process is fundamental to mitigating potential negative interferences due to the difference between the two environments, and thus seeking to improve the generalization capacity of YOLOv8 and ensure consistent performance of the inspection system, both in simulated environments and in UAVs operating in real environments.

To investigate the impact of training data composition on visual inspection performance, we designed a complete perception pipeline for UAV-based detection of solar panels using the YOLOv8 neural network. This section describes the main components of the proposed approach, which encompasses: (i) the architectural details and configuration of the YOLOv8m model; (ii) the simulation environment adopted for generating synthetic images; (iii) the construction and preprocessing of a dataset composed of both real and synthetic images; and (iv) the training and evaluation protocol applied across different

test domains.

Figure III.11 provides an overview of the proposed methodology. The process begins with the independent acquisition of real images, from a publicly available aerial dataset, and synthetic images generated within the Gazebo simulation environment. Both image sources undergo a preprocessing stage that includes resizing, normalization, and data augmentation techniques, aiming to enhance visual diversity and improve model robustness. Following this, the images are combined into mixed datasets with varying proportions of real and simulated data. Specifically, five training configurations were constructed: 100%-0%, 75%-25%, 50%-50%, 25%-75%, and 0%-100%. Each configuration was used to train a separate instance of the YOLOv8 model, allowing for a systematic investigation of how data composition affects learning performance and generalization.

In the final stage, all trained models were evaluated on three distinct test sets, that is, one composed exclusively of real images, one with only synthetic images, and a hybrid test set containing a balanced mix of both. This evaluation setup was designed to quantify the influence of training data distribution on model accuracy across domain-specific and cross-domain scenarios.

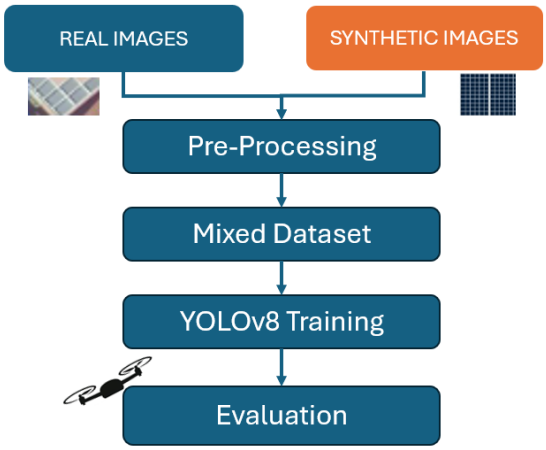


Figure III.11: Overview of the analysis of the real and simulated data composition pipeline.

By structuring the methodology around these components, we ensure a systematic comparison between models trained with different proportions of real and synthetic data, allowing us to assess the generalization capabilities of each configuration across three types of testing domains.

Chapter IV Results and Discussion

IV.1 Solar Panels Detection

YOLOv8 includes a variety of pre-trained detection models. In this study, all pre-trained YOLOv8 variants were evaluated. However, due to hardware limitations, it was not feasible to train the YOLOv8l and YOLOv8x models, as their memory requirements exceeded the available capacity of the GPU used. The experiments were conducted using the following setup: Ultralytics YOLOv8 version 2.2.48, Python 3.8.10, PyTorch 2.3.1, CUDA 12.1, and an NVIDIA GeForce GTX 960M with 4037 MiB of VRAM. Thus, given these hardware limitations, the analysis focused on the lighter YOLOv8 models, specifically YOLOv8n and YOLOv8m. These models balance performance and computational efficiency, making them suitable for environments with restricted resources (ULTRALYTICS, 2024). Both were trained on the same dataset using identical initial conditions: the number of epochs was set to 30, and the image size was set to 480, while all other parameters were left at their default values. Although both YOLOv8n and YOLOv8m performed similarly in detecting solar panels, YOLOv8m was chosen for integration into the framework due to its slightly higher capacity and better potential for scalability in more demanding scenarios. This choice was supported by the model's superior performance on key evaluation metrics during training and validation on the custom dataset. The code is available at GitHub Repository, and the synthetic images produced in Gazebo for this study are included in the dataset presented in (SILVA, 2025).

A series of quantitative metrics were analyzed to evaluate the performance of the YOLOv8m model in detecting solar panels. The confusion matrix presented in Table IV.1 reveals that the model performs well in correctly classifying solar panel instances, with a low incidence of false positives and false negatives.

Table IV.1: Confusion matrix for YOLOv8m predictions on the solar panel dataset.

Actual / Predicted	**Solar Panel**	**Background**
Solar Panel	553	23
Background	84	0

As shown in Fig. IV.1, the F1-score curve indicates that the model achieves its optimal balance between precision and recall at a confidence threshold of 0.422, resulting in an F1-score of 0.91. This suggests robust performance even with a moderate confidence threshold. The Precision-Recall (PR) curve depicted presents an Area Under the Curve (AUC) of 0.946, confirming the model's ability to maintain high precision while preserving excellent recall. The high area under the curve indicates a balanced and reliable detection performance, with strong precision and recall across various confidence thresholds.

Figure IV.2 illustrates the training and validation metrics of the YOLOv8m model over 30 epochs.

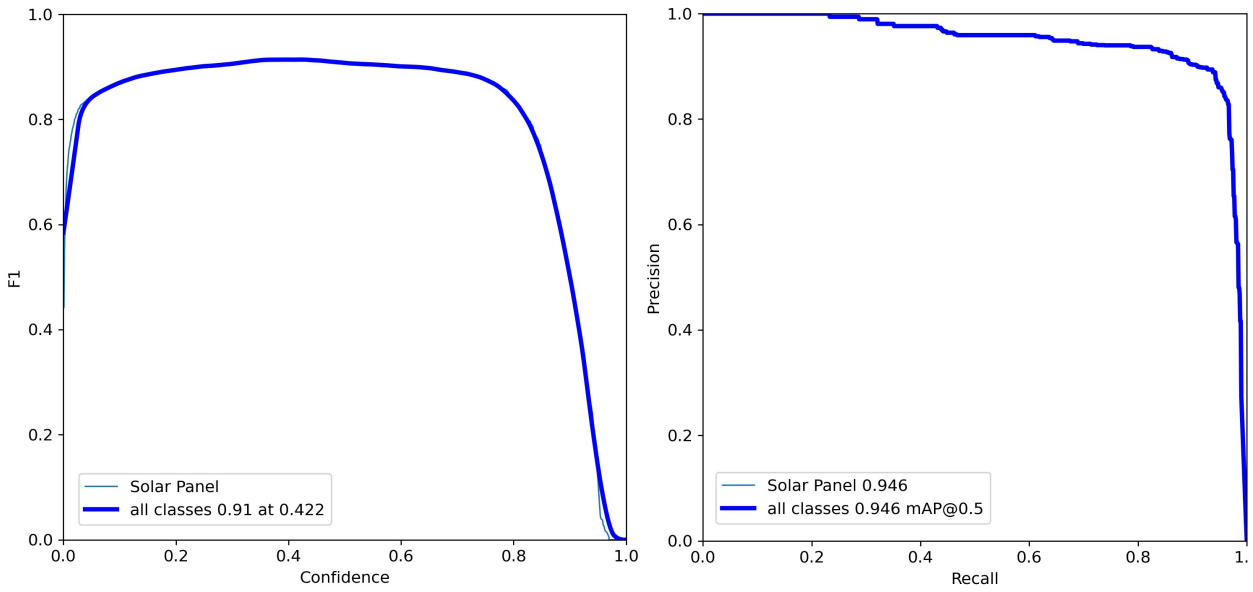


Figure IV.1: F1 and Precision x Recall Curve.

The loss curves of the bound box (box_loss), classes (cls_loss), and DFL (dfl_loss) for both training and validation sets show a consistent downward trend, indicating effective learning and convergence without signs of overfitting. The precision and recall curves steadily increase, reaching values above 0.85, demonstrating the model's capacity to accurately identify solar panels while minimizing false positives and false negatives. Similarly, the mAP@50 and mAP@5095 metrics show high values, with mAP@50 approaching 0.9, suggesting excellent object localization and classification performance.

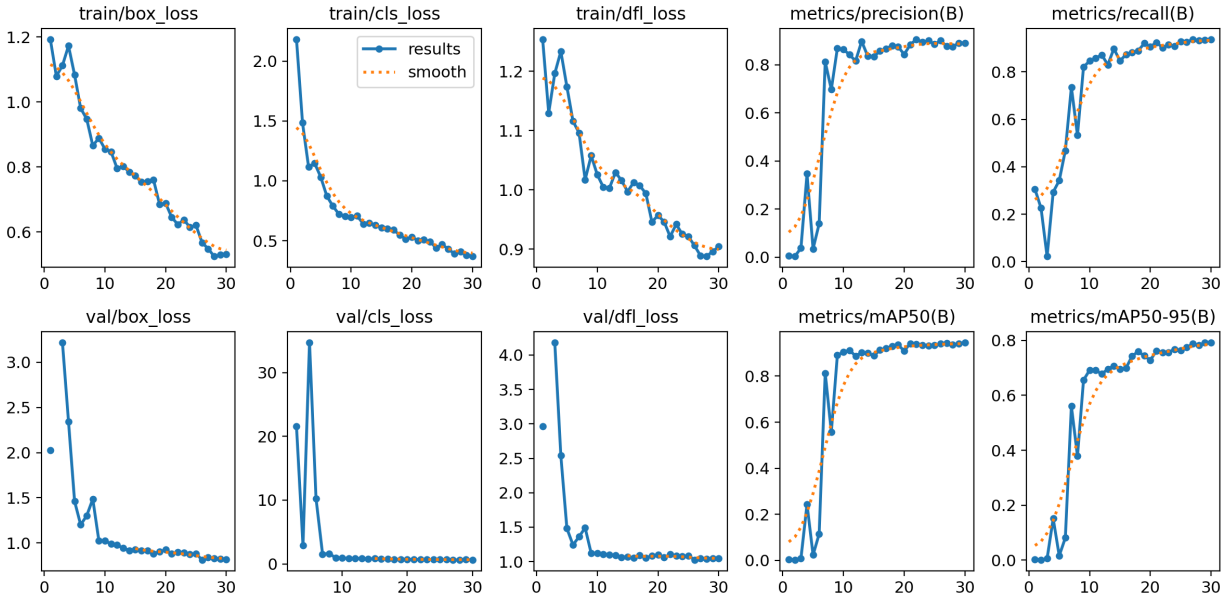


Figure IV.2: Training and validation metrics of the YOLOv8m model over 30 epochs. Loss curves indicate effective convergence, while precision, recall, and mAP metrics demonstrate strong performance and generalization capability.

IV.2 Framework results

The integrated UAV-UGV inspection framework was validated in a simulated solar panel farm environment using ROS Noetic and the Gazebo simulator. The results confirmed the system's ability to perform coordinated inspections, in which the UAV, successfully detected solar panels in real-time, while the UGV provided logistical support and battery recharging functionality. Fig. IV.3 shows the detection process applied in the proposed scenario.

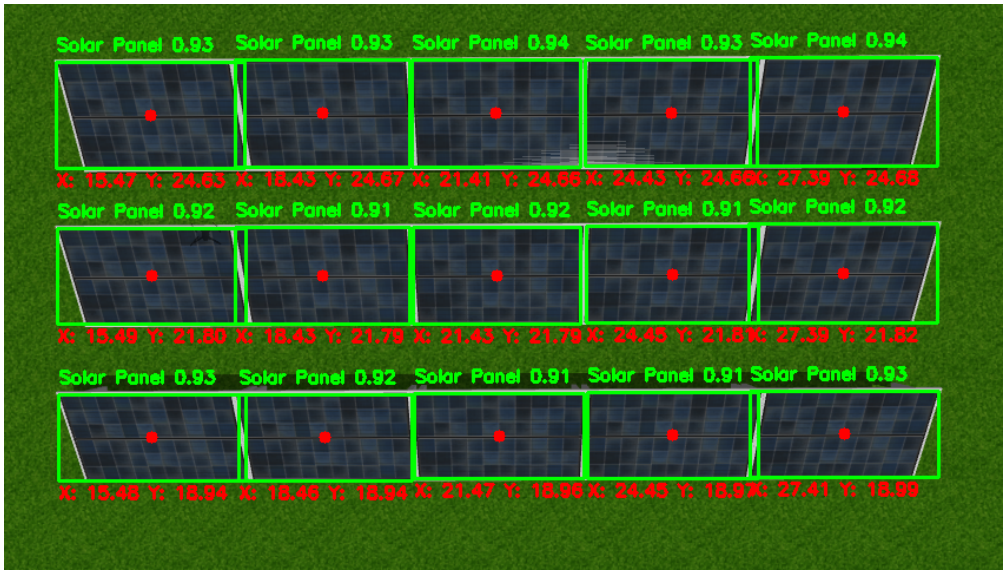


Figure IV.3: YOLO detection process applied in the proposed scenario, with the X and Y coordinates of the centers of the solar panels referenced to the coordinates of the simulated world.

As illustrated in Fig. IV.3, the detection bounding boxes generated by the onboard computer vision algorithm extract the centroids of each identified panel. Once these coordinates are stored, they are processed by a GA to compute an optimized inspection path. The optimization is guided by a multi-objective cost function that considers total travel distance and total inspection time. The decision variables include the sequence of visited panels and associated hover durations. The genetic algorithm seeks to minimize the combined cost, defined based on the equation III.1 explained earlier and represented as in the equation IV.1:

$$Cost = \alpha \cdot T_D + \beta \cdot T_T, \tag{IV.1}$$

where α and β are weighting coefficients that balance the trade-off between minimizing UAV energy consumption and mission duration, in this work, we chose $\alpha = 0.3$ and $\beta = 0.7$. The variables T_D and T_T denote the total distance and total time, respectively, computed during the optimization process.

The GA aims to minimize the UAV's travel distance and hover time while ensuring complete coverage of anomaly regions, thereby improving energy efficiency and overall mission duration. Fig. IV.4 displays the flight path executed by the UAV during a typical pre-inspection routine, before applying the Genetic

Algorithm for path optimization.

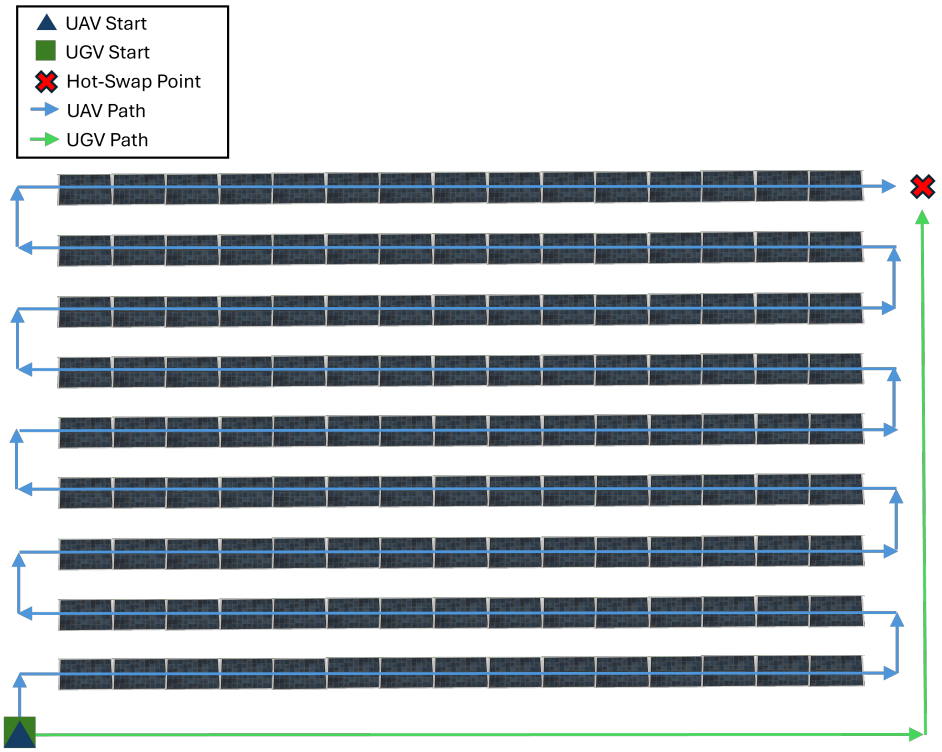


Figure IV.4: Path performed by the UAV during the pre-inspection routine.

Following the initial pass over the PV area, the system determines which panels require further analysis based on the YOLOv8 detection confidence or visual anomaly indicators. The centroids of these target panels are aggregated and used as input to the GA, which then computes a refined route that revisits only the necessary locations. This focused reinspection strategy reduces unnecessary coverage and increases the operation's efficiency.

Fig. IV.5 presents the results of the multi-objective optimization, highlighting the trade-off between distance and execution time for a single robot.

Considering panels with dimensions of 2 $\times$ 1 meter (weight $\times$ height), distance of 2 meters between centers for panels in the same row, 4 meters between centers of different rows, and different inspection times for each panel, the multi-objective GA with this panel distribution and inspection times configuration found a path of 136.2128 meters for total distance optimization and a path of 706.2455 seconds for total time optimization.

The multi-object GA was run with the following settings:

- Time limitation set to 10 seconds;
- Population of 50 individuals;
- Crossover changing from 3 to 1/3 of the individual's total tasks;
- VND (Variable Neighborhood Descent) as the search algorithm.

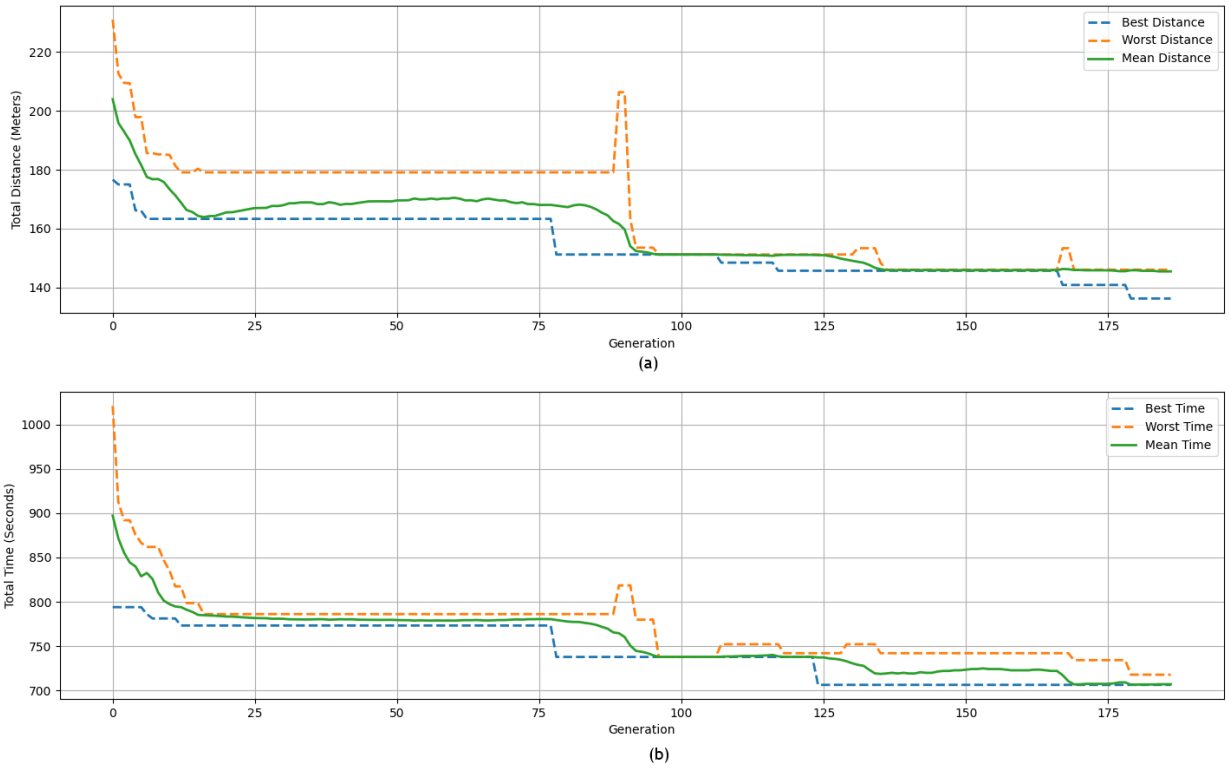


Figure IV.5: Multi-objective GA distance and time optimization graph. (a) Total Distance Evolution. (b) Total Time Evolution.

Note that the imposed time constraint does not hinder the system's real-time operation. Crossover occurs by exchanging tasks between robots, randomly selecting two individuals (parents), and generating a new individual (child) to be added to the novel population. Among the selected parents, while the first is chosen to maintain the number of tasks distributed among the robots, the second parent chooses the tasks that will replace the original ones of the first.

This process is repeated until the number of children equals the number of individuals in the population, totaling 100. After that, parents and children are subjected to a criterion to select the most suitable ones to make up the new population of 50 individuals.

VND is a local search method that deterministically explores diverse neighborhood structures, based on the idea that different neighborhood structures do not necessarily have the same local minimum. Thus, deterministic neighborhood change can solve the local optimum trap problem. VND performs three types of changes in the robot's tasks. After each change, it checks whether there was a performance improvement and maintains the change if positive; if negative, it keeps the original tasks.

The changes follow a priority order, and VND ends after any of them is applied.

1. It performs exchanges with the closest tasks;

2. It performs exchanges with the most distant tasks;

3. It performs a random exchange in two tasks.

Figs. IV.6 and IV.7 illustrate the results of this process. The colored red panels indicate panels selected for further inspection. In contrast, the red path represents the optimal route calculated by the GA for the best distance IV.6 and best time IV.7.



Figure IV.6: Inspection path with the shortest distance traveled, optimized by the Genetic Algorithm. Panels highlighted in red require re-inspection; the red line shows the calculated route.

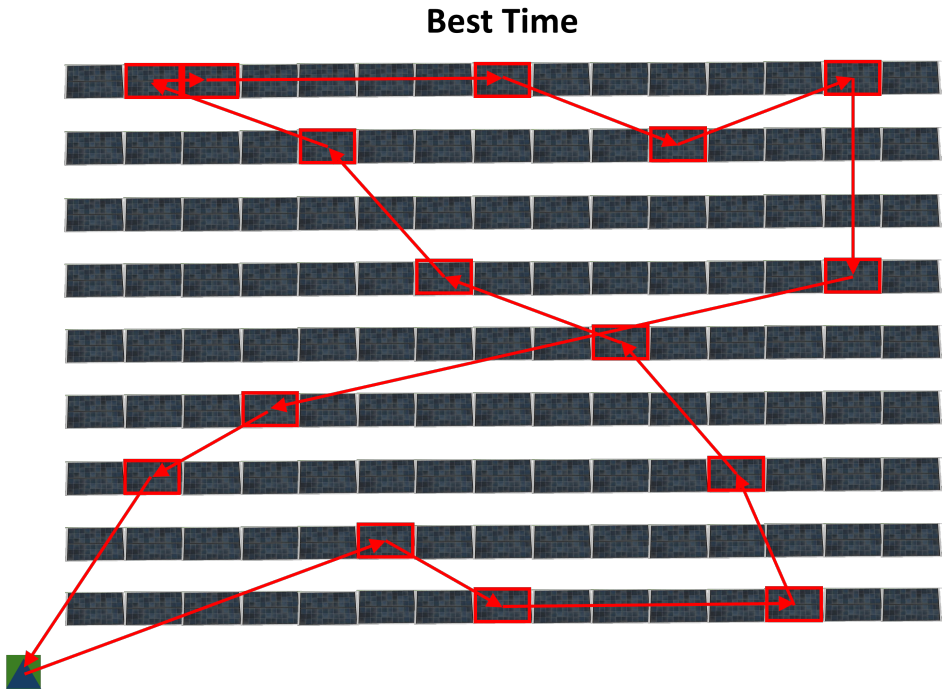


Figure IV.7: Inspection path with the shortest time, optimized by Genetic Algorithm. Panels highlighted in red require re-inspection; the red line shows the calculated route.

This integration of real-time perception and intelligent planning demonstrates the adaptability and

autonomy of the proposed framework. Fig. IV.8 depicts the UAV's trajectory in response to a low battery event, which triggers a support action from the UGV. Upon receiving a signal from the UAV, the GCS commands the UGV to reposition itself to a safe landing location. This enables the UAV to land, recharge, and resume the inspection process, extending the mission's endurance.

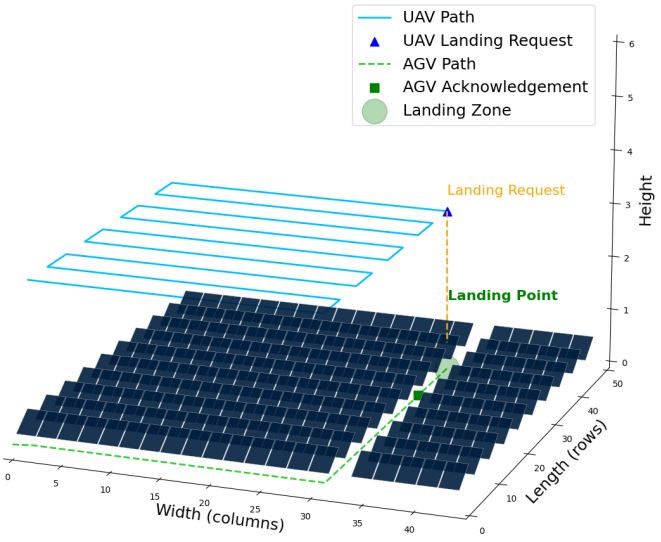


Figure IV.8: UAV path when requesting a support action to the UGV.

IV.3 Evaluation of the YOLO model

In this section, we evaluate the performance of YOLOv8 models trained with different compositions of real and simulated data. The tests were conducted on three distinct datasets: real images, simulated images, and a combined dataset containing both types of data. Each model was trained with a specific proportion of real and simulated images, ranging from 100% real to 100% simulated.

IV.3.1 Testing on real datasets

Initially, the model trained on a dataset of 200 exclusively real images was tested on 25 real images, as illustrated in Figure IV.9, which shows the F1-score and the corresponding precision-recall curve. Figure IV.10 visually presents the detection results. In this scenario, a satisfactory performance was observed, although with limitations in terms of generalization to unseen data.

Subsequently, tests were conducted with models trained on mixed data. Figure IV.11 shows the F1-score and precision-recall curve for the model trained with 75% real images (200 images) and 25% simulated images (66 images), and Figure IV.12 displays the detection results. An improvement in generalization capability can be observed, with a good balance between precision and recall.

With the ratio of 50% real (200 images) and 50% simulated (200 images), a worsening of the overall performance was observed in relation to the previous one. As evidenced in Figure IV.13 with the

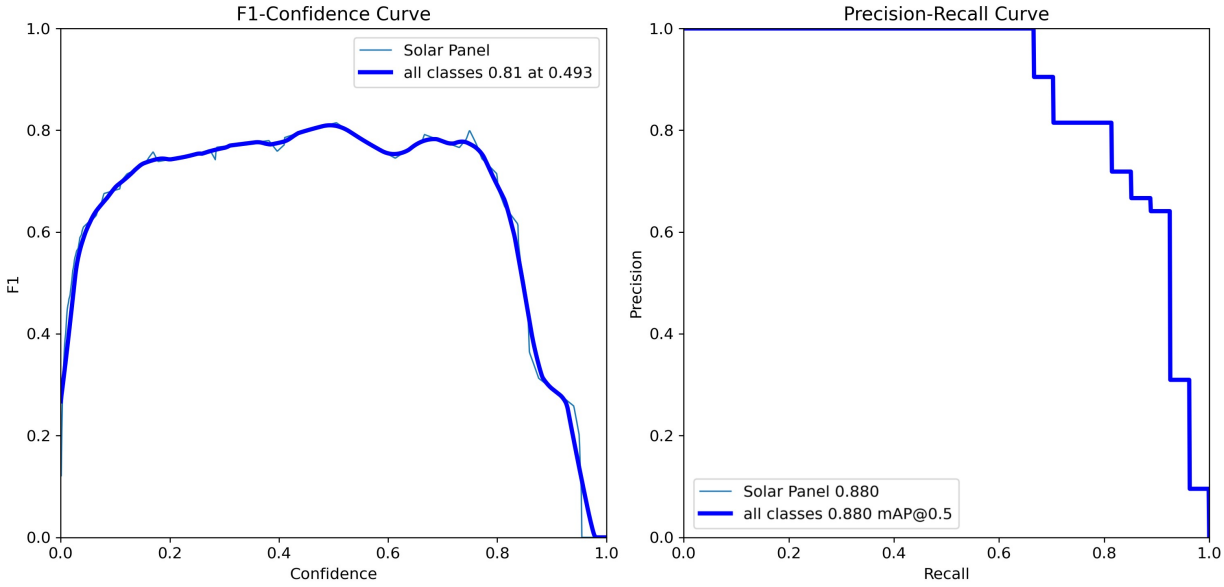


Figure IV.9: F1 and Precision-recall curve real training dataset tested on real test dataset.



Figure IV.10: Real training dataset predicts on real test dataset.

F1-score and precision-recall curve and in Figure IV.14 with the detections.

When reducing the amount of real data in the training, going to 25% real (66 images) and 75% simulated (200 images), the results started to deteriorate. Figure IV.15 shows the F1-score and the precision-recall curve for the trained model, and Figure IV.16 shows the detections.

Finally, the model trained only with simulated data (200 images), whose results are illustrated in Figure IV.17, presented inferior performance when tested in real images, with a high number of false

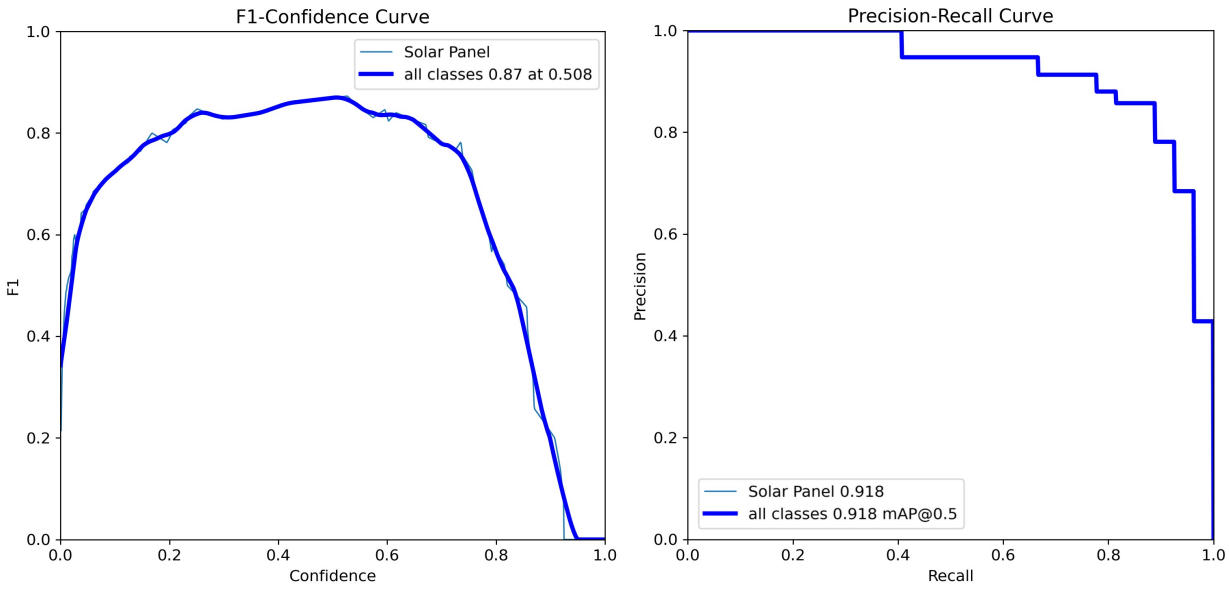


Figure IV.11: F1 and Precision-recall curve 75% real and 25% simulated training dataset tested on real test dataset.



Figure IV.12: 75% real training dataset predict on real test dataset.

negatives and practically null graphs, which is confirmed in the detections shown in Figure IV.18.

Table IV.2 presents the comparison of confusion matrices for the different YOLOv8m models when evaluated on a test set composed exclusively of real images (25 images). The results show that the model trained with 75% real data and 25% simulated data achieved the best overall performance in this scenario, with 25 true positives (TP), only four false positives (FP), and two false negatives (FN). This model slightly outperformed the one trained with 50% real data, which obtained the same number of

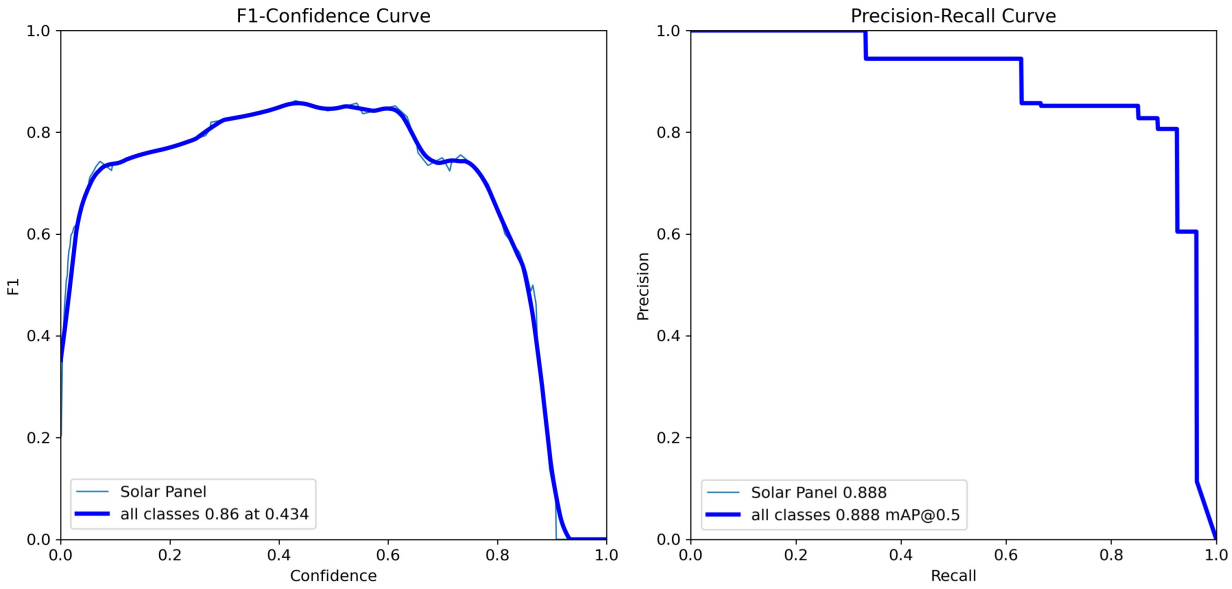


Figure IV.13: F1 and Precision-recall curve 50% real and 50% simulated training dataset tested on real test dataset.



Figure IV.14: 50% real and 50% simulated training dataset predict on real test dataset.

TP and FN but had twice as many FP (8). Meanwhile, the model trained exclusively with real data, although it also achieved 24 TP, showed a higher number of FP (10), suggesting a greater tendency to identify solar panels incorrectly.

On the other hand, models predominantly trained with simulated data, especially those trained solely on simulated images, performed significantly worse. The 100% simulated model, according to Table IV.2, failed to correctly identify any images (0 TP) and classified all real samples as false negatives

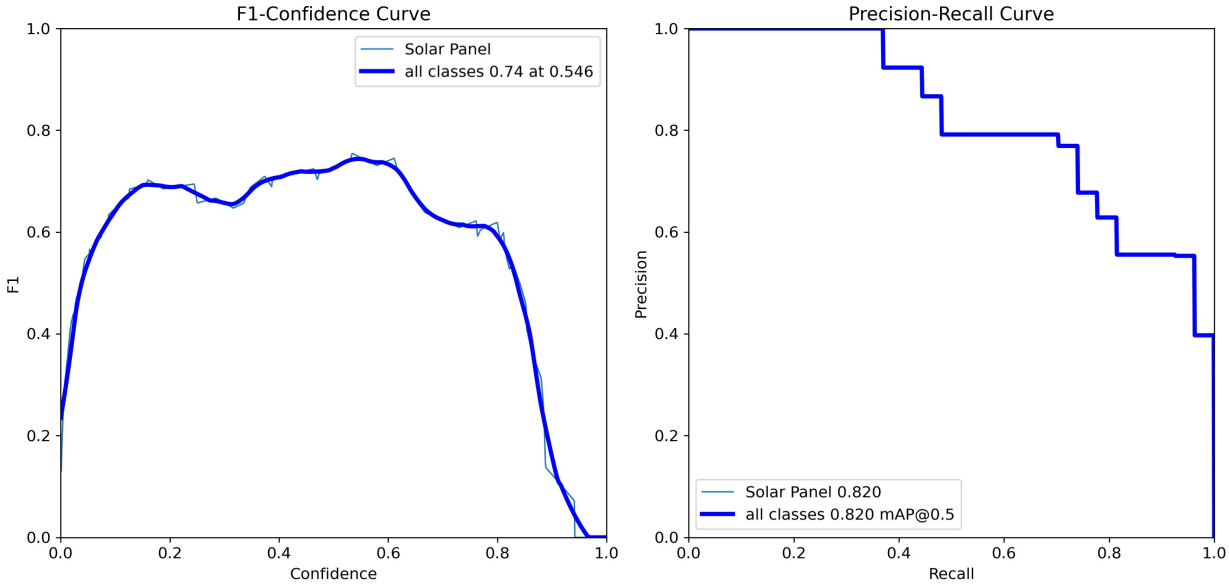


Figure IV.15: F1 25% real and 75% simulated training dataset tested on real test dataset.



Figure IV.16: 25% real and 75% simulated training dataset predict on real test dataset.

(27 FN). However, as shown in Figure IV.18, despite failing in most cases, the 100% simulated model was able to individually identify some panels, unlike previous models, which tended to detect clusters of panels.

Overall, the results in Table IV.1 reinforce the importance of including real data in the training process and demonstrate that the combined use of real and simulated dataespecially in a 75/25 ratiocan significantly enhance the model's accuracy and reliability in real-world applications.

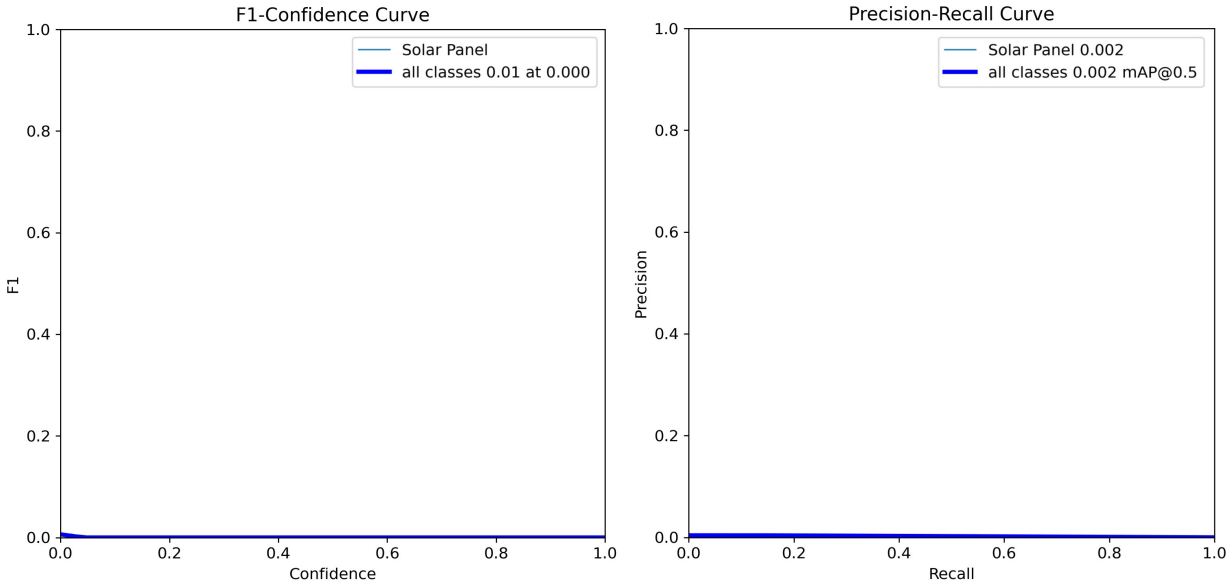


Figure IV.17: F1 and Precision-recall curve simulated training dataset tested on real test dataset.



Figure IV.18: Simulated training dataset predict on real test dataset.

Table IV.2: Comparison of confusion matrices for YOLOv8m predictions on real test datasets.

Dataset	TP	TN	FP	FN
Real	24	0	10	3
75% Real	25	0	4	2
50%	25	0	8	2
75% Simulated	23	0	15	4
Simulated	0	0	0	27

IV.3.2 Testing on simulated datasets

In this stage, the performance of the YOLOv8m models was evaluated when tested exclusively on simulated images. The objective was to assess the generalization capability of models trained with different proportions of real and simulated data when applied to virtual environments, which are helpful for large-scale validations and automated anomaly generation.

The model trained exclusively on real data showed unsatisfactory performance in the simulated domain, with a low F1-score and a low area under the precision-recall curve, illustrated in Figure IV.19. As seen in the detections presented in Figure IV.20, some panels were not detected, while in others, multiple partial detections occurred within the same panel. These results highlight that although real data is crucial for field applications, it is not sufficient to ensure good performance in synthetic environments, especially when there are discrepancies in textures, contrasts, or noise levels.

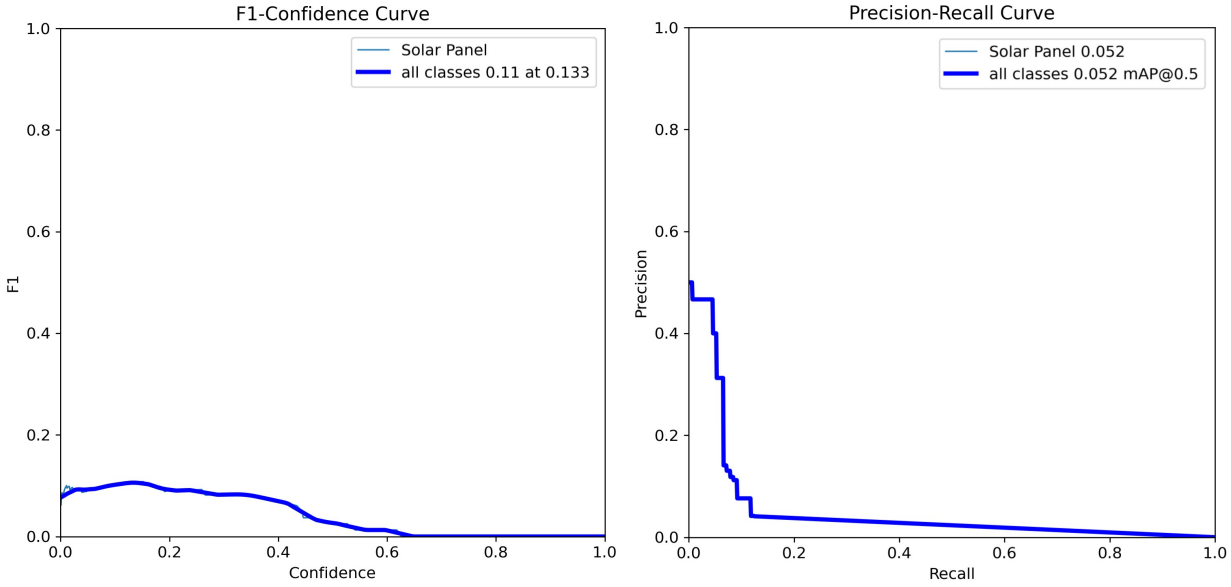


Figure IV.19: F1 and Precision-recall curve real training dataset tested on simulated test dataset.

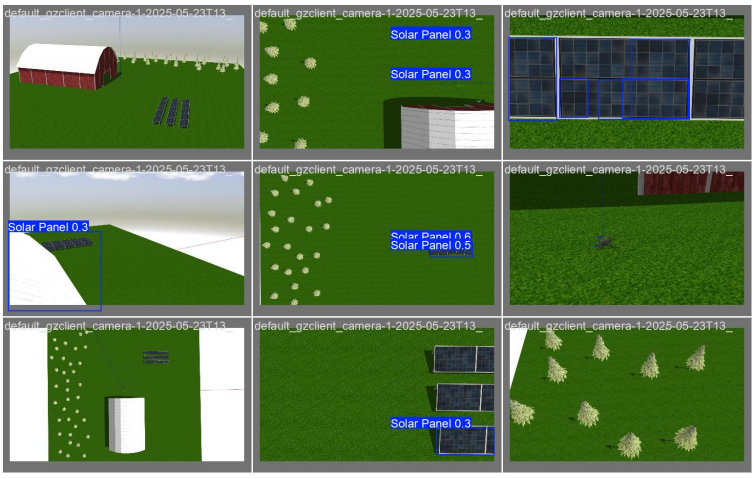


Figure IV.20: Real training dataset predict on simulated test dataset.

By including 25% simulated data in the training set (200 real images + 66 simulated images), a performance improvement was observed. Figure IV.21 shows an increase in the F1-score and reveals a better balance between precision and recall. The detections in Figure IV.22 illustrate the improved detection of the panel and the correction of the error involving multiple detections within the same panel. However, the performance is still lower than that achieved by models trained with a higher proportion of simulated data.

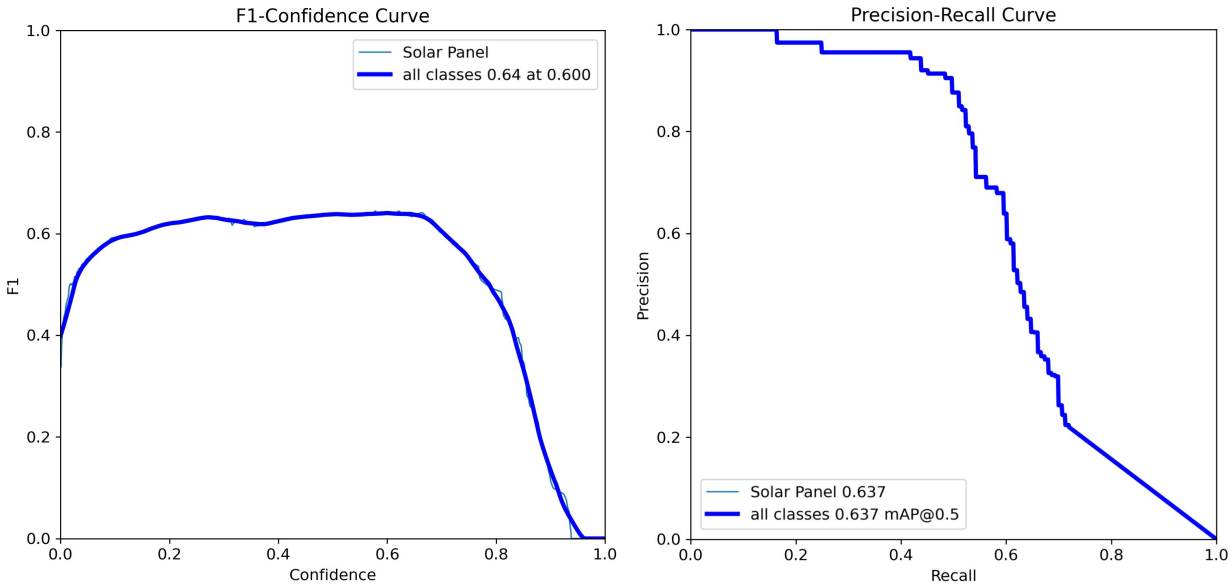


Figure IV.21: F1 and Precision-recall curve 75% real and 25% simulated training dataset tested on simulated test dataset.

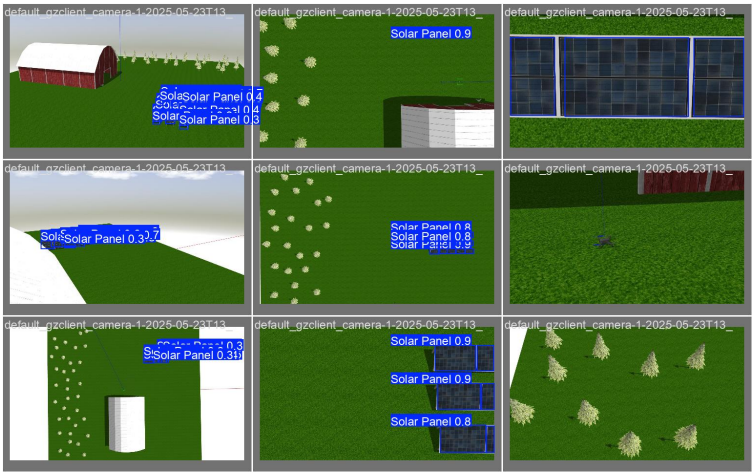


Figure IV.22: 75% real and 25% simulated training dataset predict on simulated test dataset.

With a balanced training set composed of 50% of each data type (200 real images + 200 simulated images), the results showed a significant improvement. Figure IV.23 presents a higher and more stable F1-score, and the Precision-recall curve indicates good model coverage in the simulated domain, as shown in Figure IV.24. This combination proved to be effective in ensuring robust performance in both

real and virtual environments.

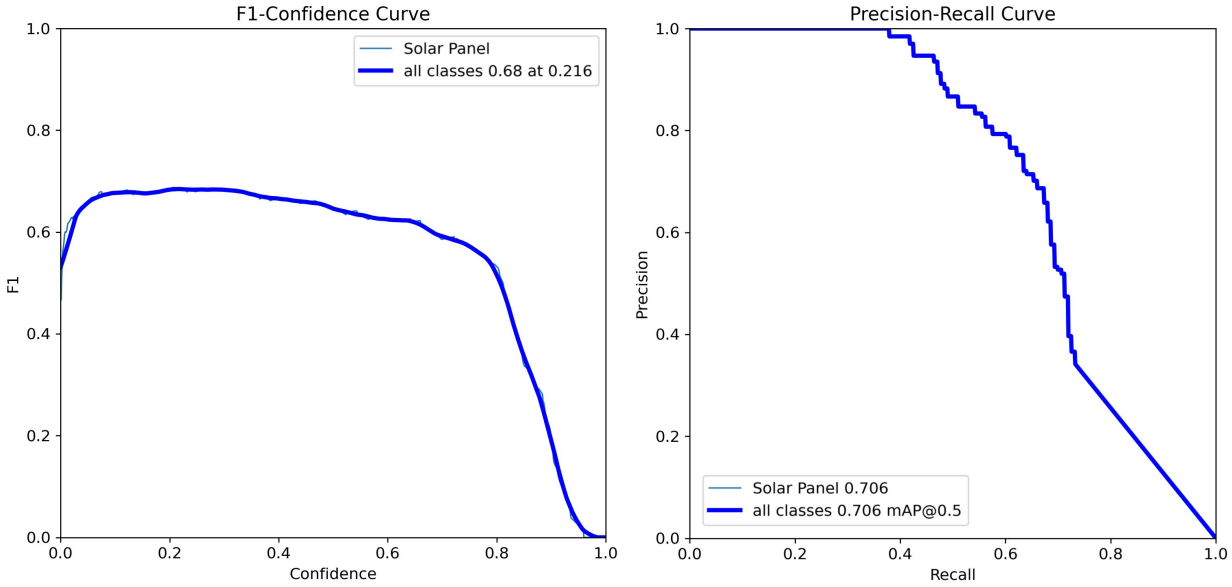


Figure IV.23: F1 and Precision-recall curve 50% real and 50% simulated training dataset tested on simulated test dataset.

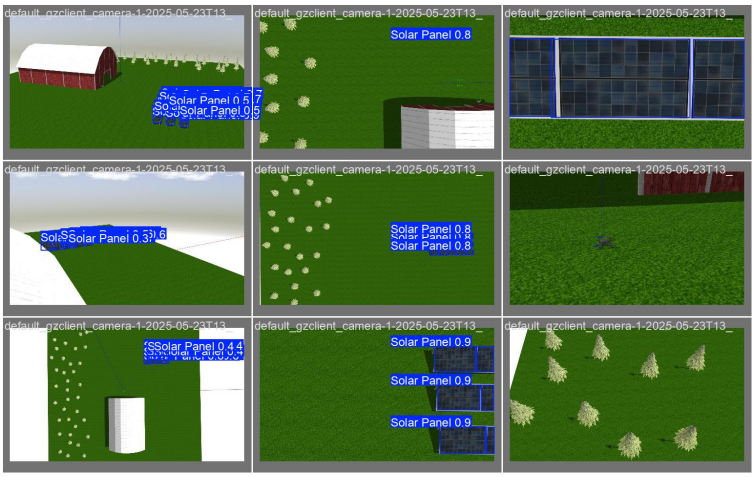


Figure IV.24: 50% real and 50% simulated training dataset predict on simulated test dataset.

The model trained with a predominance of simulated data (66 real images and 200 simulated images) performed slightly better in this domain. As shown in Figures IV.25 and IV.26, both the F1-score and the precision-recall curve improved, reinforcing the benefit of extensive exposure to the patterns present in the test environment.

Finally, the model trained exclusively with simulated data achieved the best performance in this scenario, as evidenced by Figure IV.27, which shows a high and consistent F1-score and demonstrates a large area under the precision-recall curve. These results indicate that, for tasks carried out in purely simulated environments, the exclusive use of synthetic data for training is sufficientand even desirable, as illustrated by the detections in Figure IV.28.

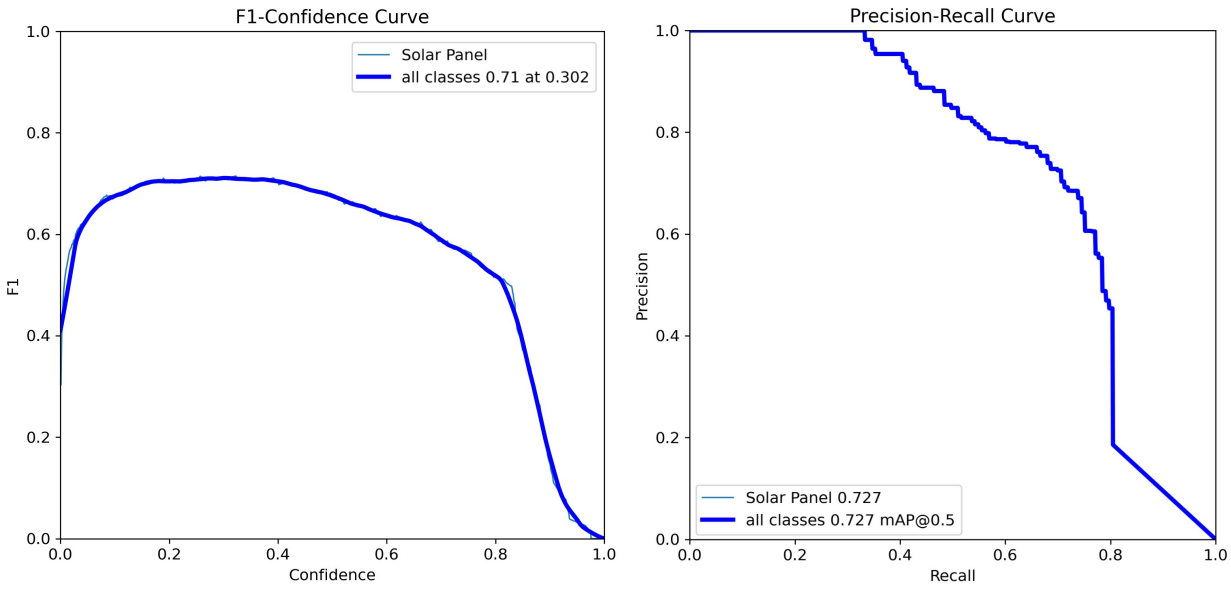


Figure IV.25: F1 and Precision-recall curve 25% real and 75% simulated training dataset tested on simulated test dataset.

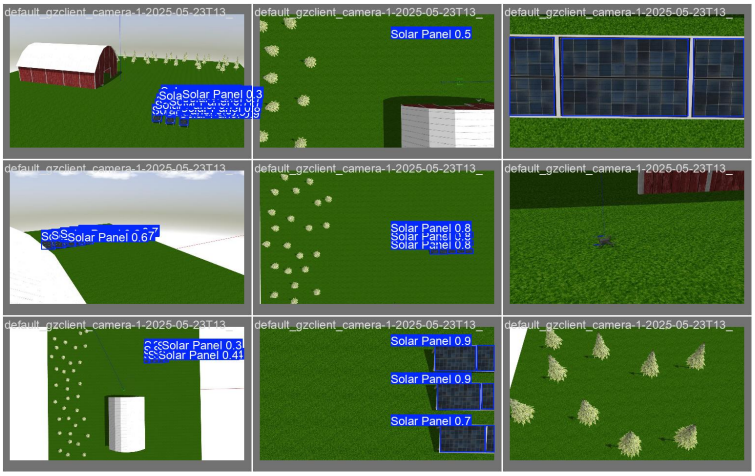


Figure IV.26: 25% real and 75% simulated training dataset predict on simulated test dataset.

Table IV.3 presents the confusion matrix results for the models trained with different data compositions when tested exclusively on simulated images. The values highlight the influence of the nature of the training data on the model's ability to recognize patterns in the virtual environment. The model trained solely with real data performed the worst in this scenario, with only 8 TP and 145 FN. This indicates that nearly all the solar panels present in the simulated images were not detected, reflecting a poor generalization capability for the synthetic domain. This behavior aligns with the previously discussed F1-score and precision-recall curves, confirming that training exclusively with real data is not suitable for simulated scenarios.

The introduction of simulated data in the training process led to progressive improvements. The model trained with 75% real data and 25% simulated data achieved 93 TP and reduced FNs to 60, although it still exhibited a high number of FP (36). With a 50% real and 50% simulated data split,

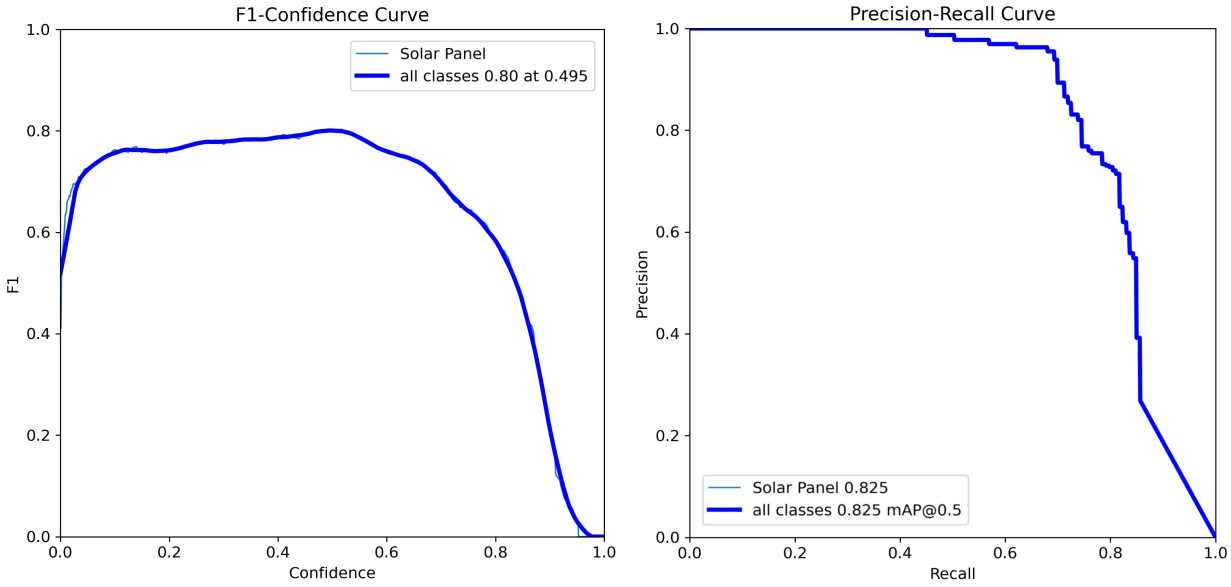


Figure IV.27: F1 and Precision-recall curve simulated training dataset tested on real simulated dataset.

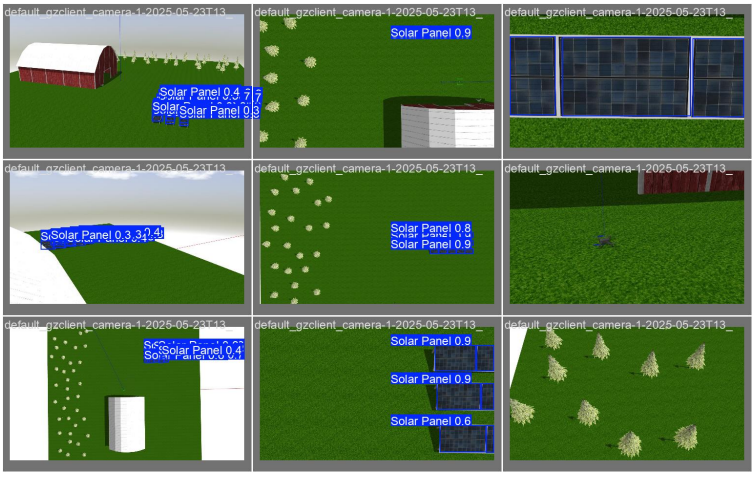


Figure IV.28: Simulated training dataset predict on simulated test dataset.

further improvements were observed: 99 TP and a reduction of FP to 20, with a significant overall improvement in the balance between sensitivity and precision.

Models with a predominance of simulated data achieved the best performances. The model trained with 75% simulated and 25% real data obtained 110 TP and 38 FN, while the model trained exclusively with simulated data achieved the highest number of TP (115) and the lowest number of FN (38), with a moderate number of false positives (25). This result confirms that, for tasks in synthetic environments, the similarity between training and testing data is crucial.

These findings reinforce the conclusion that using simulated data in training is essential when the goal is to apply the model in artificial contexts. Furthermore, the analysis of Table IV.1 shows that although the absolute performance of the models improves with a higher proportion of simulated data, using a balanced fraction that includes real data is also effective, especially when aiming for a versatile

model suitable for multiple domains.

Table IV.3: Comparison of confusion matrices for YOLOv8m predictions on simulated test datasets.

Dataset	TP	TN	FP	FN
Real	8	0	7	145
75% Real	93	0	36	60
50%	99	0	20	54
75% Simulated	110	0	38	43
Simulated	115	0	25	38

IV.3.3 Testing on 50% real and 50% simulated datasets

In this step, all models were tested on a hybrid dataset composed of 50% real and 50% simulated images (200 real images + 200 simulated images). The goal was to evaluate how well each model adapts to scenarios with realistic and artificial variability, representing a common practical situation in automated inspection applications. The model trained only with real data showed poor performance in this hybrid scenario. As illustrated in Figure IV.29, the F1-score was unsatisfactory, which is corroborated by the low area under the precision-recall curve. This result demonstrates that, although effective in real domains, this model has difficulty adapting to synthetic content not seen during training, as shown in Figure IV.30.

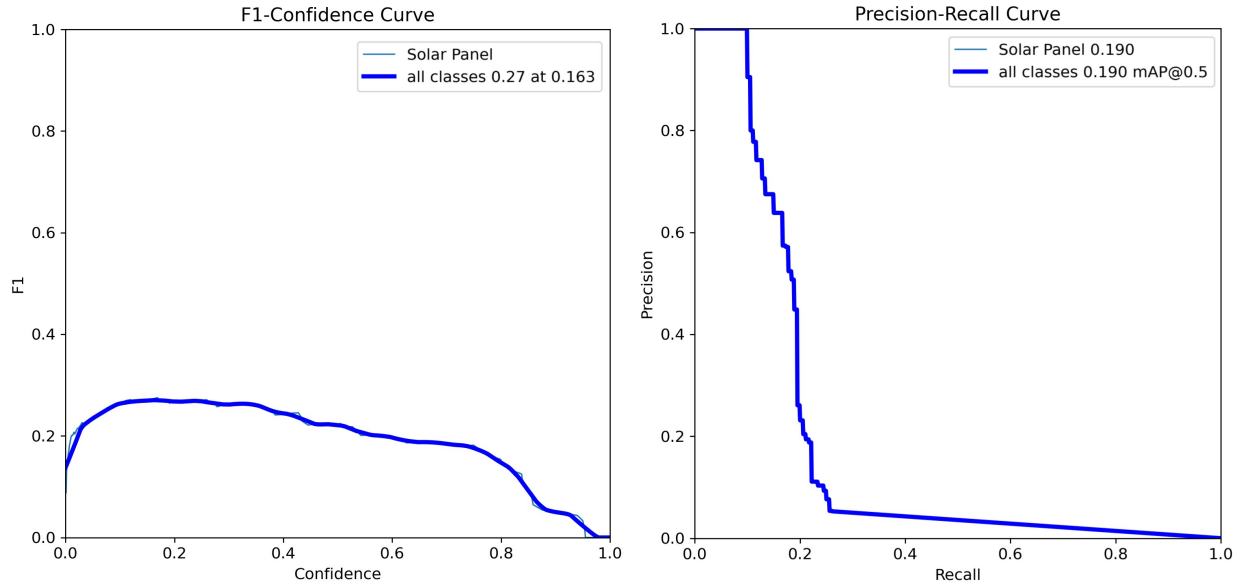


Figure IV.29: F1 and Precision-recall curve real training dataset tested on 50% real and 50% simulated test dataset.

With the addition of 25% simulated data to the training set (200 real images and 66 simulated images), performance improved substantially. Figure IV.31 shows a significant increase in the F1-score and confirms a greater balance between precision and recall. This improvement indicates that even a small amount of simulated data can enhance generalization to mixed domains, as evidenced by the

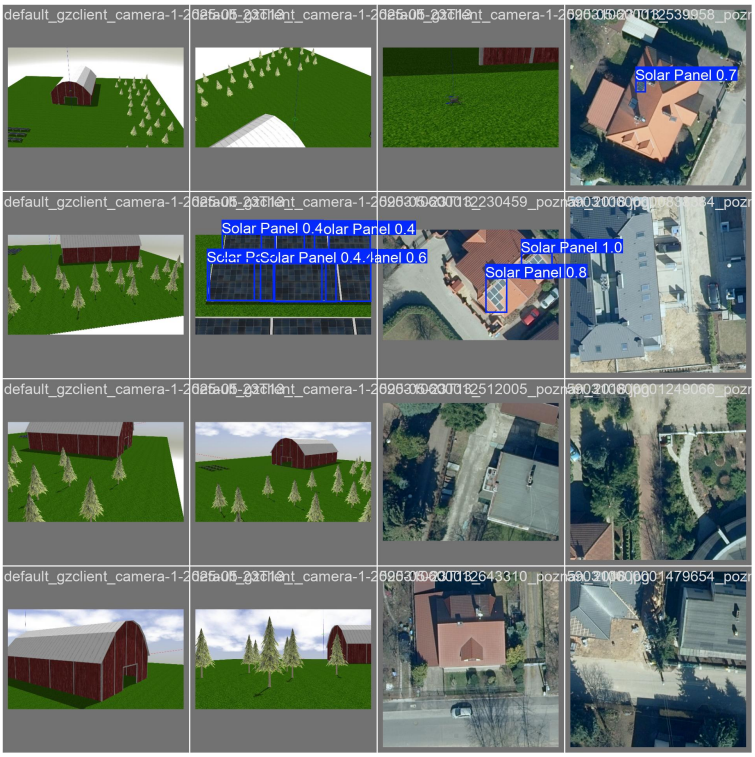


Figure IV.30: Real training dataset predict on 50% real and 50% simulated test dataset.

detections in Figure IV.36.

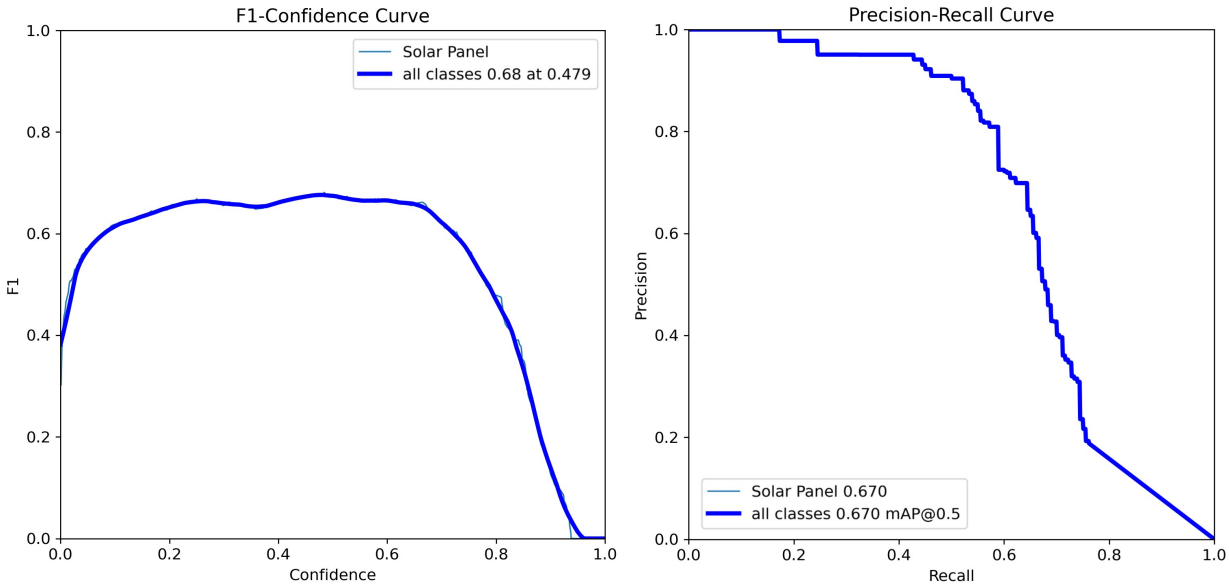


Figure IV.31: F1 and Precision-recall curve 75% real and 25% simulated training dataset tested on 50% real and 50% 50% real and 50% simulated test dataset.

The model with an exact balance between real and simulated data (50/50) achieved the best performance among all configurations. Figure IV.33 shows a consistently high F1-score and highlights the precision-recall curve, and Figure IV.34 displays the detections. This result reinforces that training with a balanced diversity of domains provides superior robustness, making the model more adaptable to

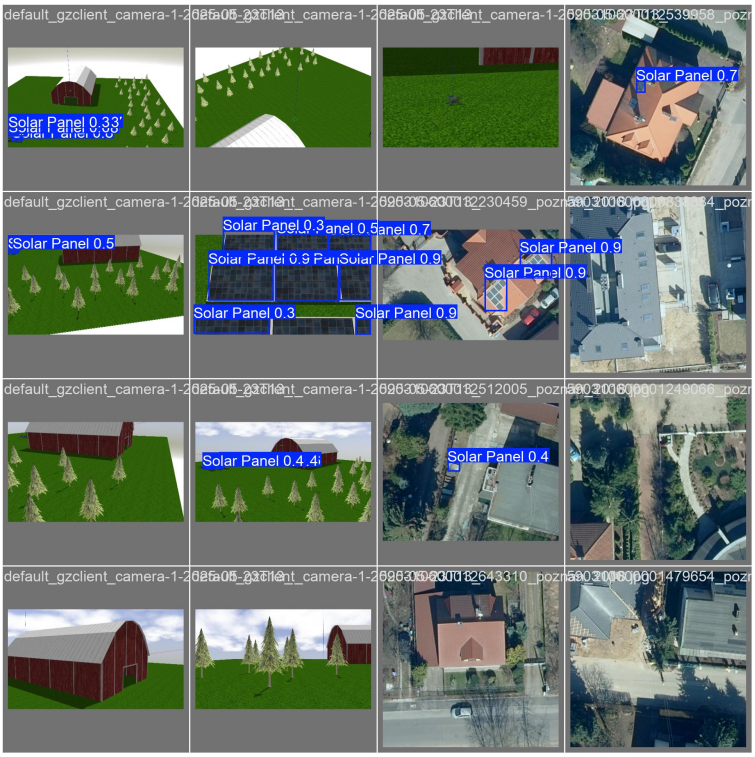


Figure IV.32: 75% real and 25% simulated training dataset predict on 50% real and 50% simulated test dataset.

varied contexts.

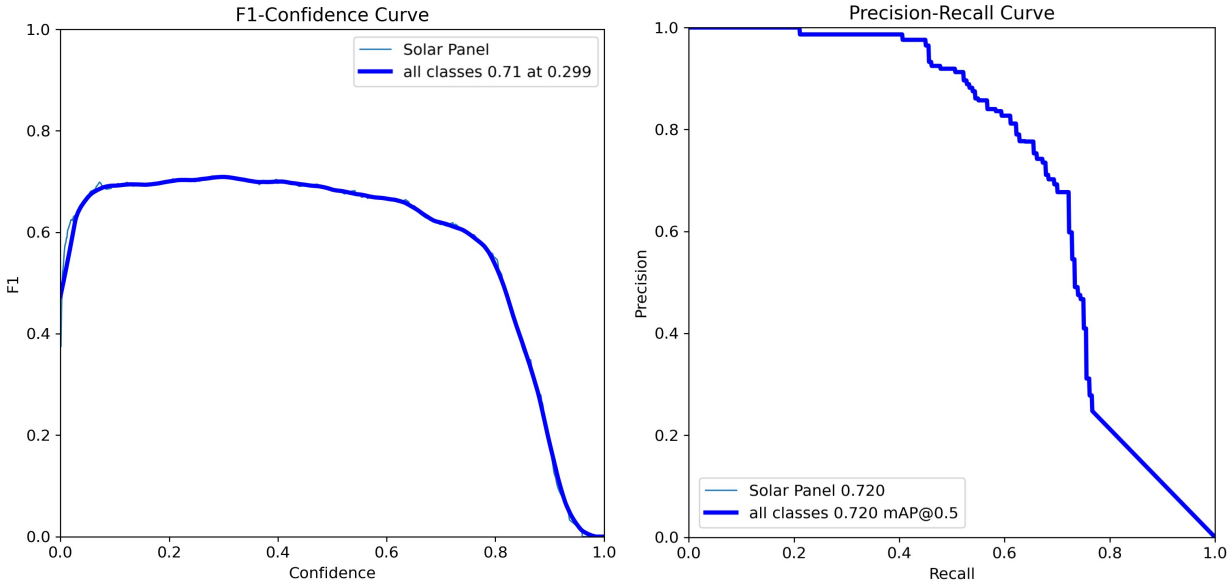


Figure IV.33: F1 and Precision-recall curve 50% real and 50% simulated training dataset tested on 50% real and 50% simulated test dataset.

On the other hand, the models with a predominance of simulated data (66 real images + 200 simulated images) and those trained exclusively with simulated data showed lower performance compared to the balanced model. As shown in Figures IV.35 and IV.37, the F1-score is reasonable but lower, and

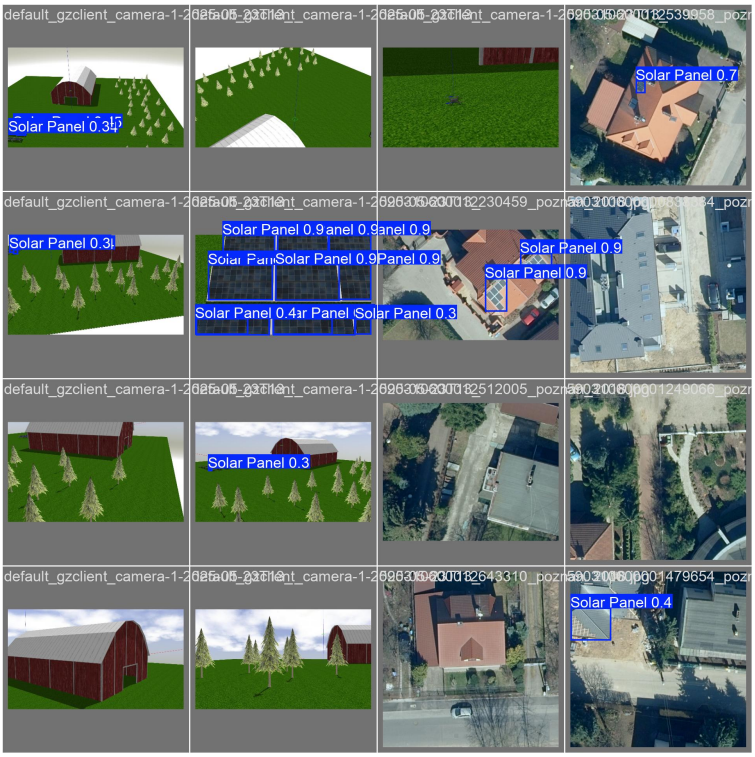


Figure IV.34: 50% real and 50% simulated training dataset predict on 50% real and 50% simulated test dataset.

the precision-recall curve reveal a drop in precision, which may indicate a higher number of false positives. The detections can also be seen in Figures IV.36 and IV.38.

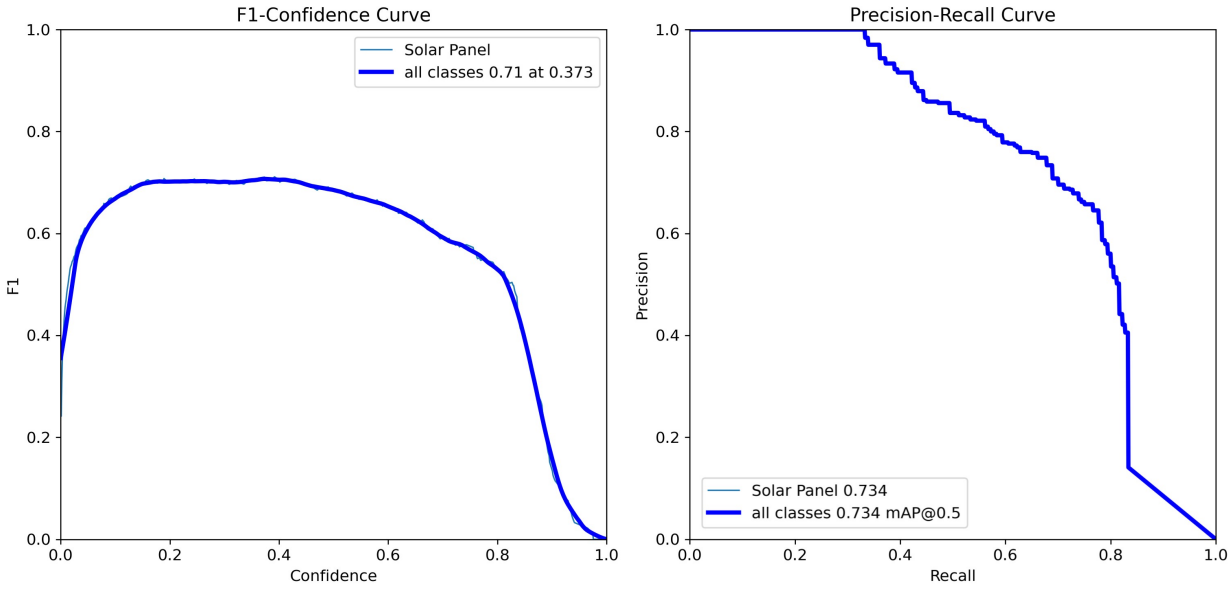


Figure IV.35: F1 and Precision-recall curve 25% real and 75% simulated training dataset tested on 50% real and 50% simulated test dataset.

Table IV.4 summarizes the confusion matrix results obtained from tests using the hybrid dataset composed of 50% real and 50% simulated data. The model trained with 50% real and 50% simulated

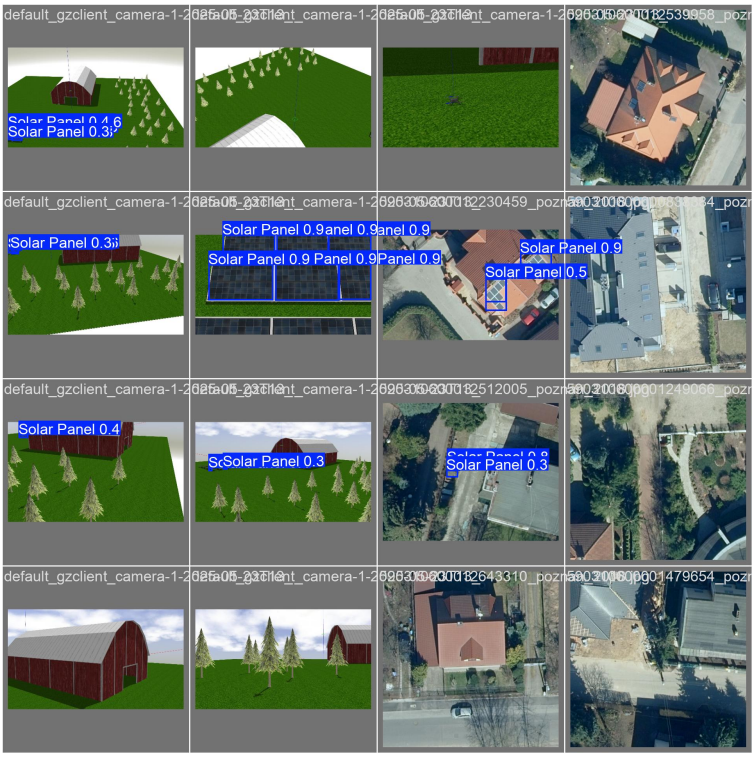


Figure IV.36: 25% real and 75% simulated training dataset predict on 50% real and 50% simulated test dataset.

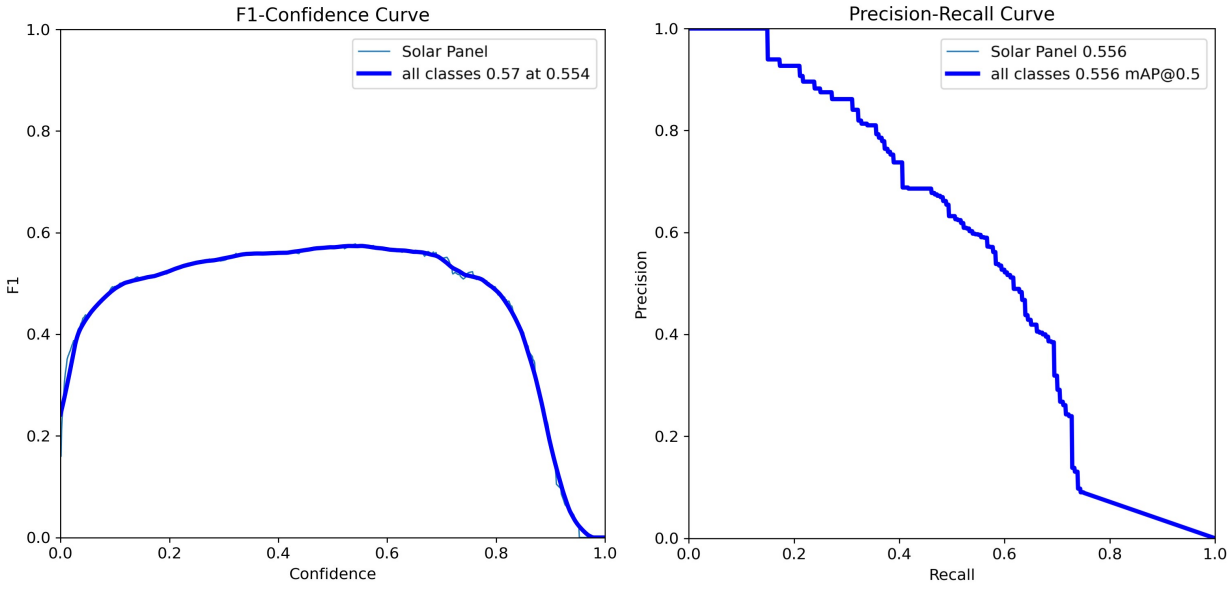


Figure IV.37: F1 and Precision-recall curve simulated training dataset tested on 50% real and 50% simulated dataset.

data showed the best balance between true positives (125) and false negatives (55), with a moderate number of false positives (30). The model with 75% simulated data achieved the highest number of true positives (133), but also had a higher number of false positives (53), compromising its precision. On the other hand, the model trained exclusively with real data performed significantly worse, with only 32 true positives and 148 false negatives, highlighting its poor generalization to mixed environments.

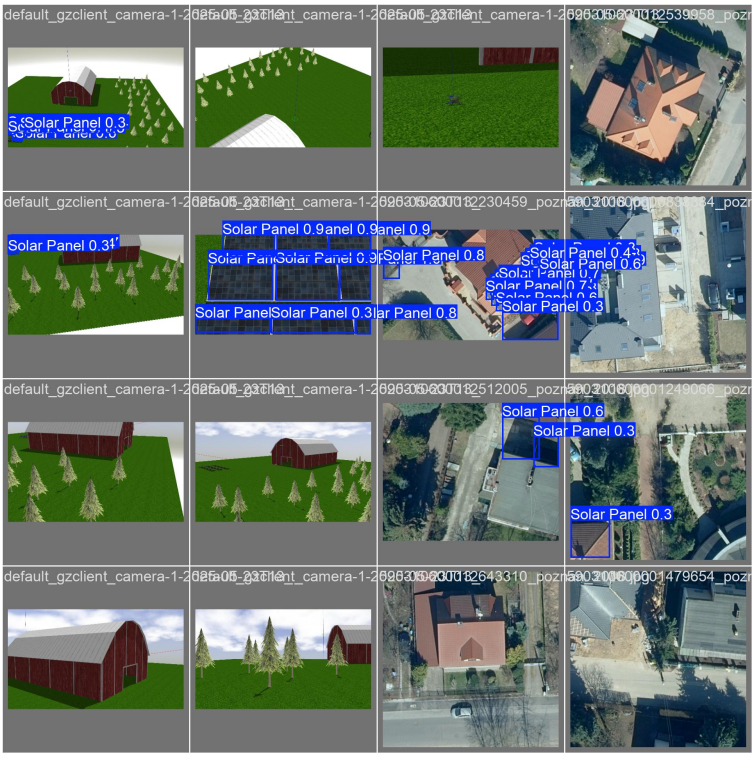


Figure IV.38: Simulated training dataset predict on 50% real and 50% simulated test dataset.

These results reinforce that a balanced training approach (50/50) is the most effective for scenarios that combine real and simulated data.

Table IV.4: Comparison of confusion matrices for YOLOv8m predictions on 50% real and 50% simulated test datasets.

Dataset	TP	TN	FP	FN
Real	32	0	17	148
75% Real	118	0	40	62
50%	125	0	30	55
75% Simulated	133	0	53	47
Simulated	115	0	25	65

IV.3.4 Discussion and analysis

To better analyze the results of the trained models, the following metrics were used: mAP@0.5, mAP@0.5:0.95, average precision, and average recall. These metrics were obtained by exposing each trained model to the test datasets and were then organized into separate tables, each corresponding to a different test dataset. Table IV.5 presents the metrics for the test dataset with real images, Table IV.6 for the simulated images, and finally, Table IV.7 for the hybrid test dataset.

As shown in Table IV.5, for testing with real images, the best performance was achieved by the model trained with the 75% real dataset (200 real images + 66 simulated images), followed by the model trained with the 50% real and 50% simulated dataset (200 real + 200 simulated images), which

performed similarly to the model trained exclusively on real data (200 real images). This demonstrates that even in scenarios involving only real environments, the addition of simulated images can positively contribute to model performance.

In contrast, Table IV.6, which reports results for testing with simulated images, shows that the model trained with 100% simulated data (200 simulated images) achieved the best performance, followed by the models with 75% and 50% simulated data, which produced similar results. This indicates that in fully simulated environments, the inclusion of real images may introduce uncertainties in detections and reduce overall model performance. In Table IV.7, using the hybrid test dataset (200 real + 200 simulated images), the best results came from the models trained with 75% and 50% simulated images, with similar performance levels, where one outperformed the other in some metrics. This suggests that in hybrid-use scenarios, balancing the training dataset with images that represent all use cases, or prioritizing the inclusion of images with higher definition, contrast, and less noise, could improve model performance.

Another interesting point to mention is the performance of the model trained with the 100% simulated dataset. When tested on the 100% real dataset, as previously discussed and shown in Figures IV.17 and IV.18, its performance was practically null. This is also evident in the metrics from Table IV.5. However, when analyzing the detections on the images, we observe that this model identified solar panels differently from the other models. While other models tended to detect multiple panels within a single bounding box, the model trained exclusively on simulated data detected each panel individually, generating a separate bounding box for each one. This behavior indicates that the model learned strong low-level visual features associated with panel geometry and texture in the simulated domain, but lacked the contextual understanding necessary to aggregate adjacent panels into a single detection in real-world scenarios. This effect highlights the domain gap between simulated and real environments and reinforces the importance of incorporating real data to learn higher-level semantic grouping and spatial context.

Figure IV.39 illustrates this difference in detection patterns between the model trained with 100% simulated data and the model trained with 75% real data, which achieved the best performance on real image tests.



Figure IV.39: Comparison between detections from a 100% simulated training dataset versus a 75% real training dataset.

Table IV.5: Comparison of metrics by model predictions on real test datasets.

Dataset	mAP@0.5	mAP@0.5:0.95	Average Accuracy	Average Recall
Real	0.8805	0.6116	0.8032	0.8148
75% Real	0.9184	0.6971	0.8476	0.8889
50%	0.8881	0.6481	0.8060	0.9234
75% Simulated	0.8198	0.5177	0.7632	0.7164
Simulated	0.0023	0.0004	0.0039	0.1481

Table IV.6: Comparison of metrics by model predictions on simulated test datasets.

Dataset	mAP@0.5	mAP@0.5:0.95	Average Accuracy	Average Recall
Real	0.0519	0.0271	0.2756	0.0654
75% Real	0.6372	0.4210	0.8663	0.5098
50%	0.7060	0.5059	0.7655	0.6188
75% Simulated	0.7269	0.4986	0.7383	0.6863
Simulated	0.8254	0.5506	0.9492	0.6928

Table IV.7: Comparison of metrics by model predictions on 50% real and 50% simulated test datasets.

Dataset	mAP@0.5	mAP@0.5:0.95	Average Accuracy	Average Recall
Real	0.1905	0.1241	0.5370	0.1778
75% Real	0.6697	0.4501	0.7962	0.5889
50%	0.7195	0.5157	0.7762	0.6549
75% Simulated	0.7339	0.4945	0.7410	0.6778
Simulated	0.5557	0.3801	0.5796	0.5667

Chapter V Conclusions and Future Work

V.1 Conclusions

Presented a robust and modular architecture for the autonomous inspection of PV farms through the collaboration of UAVs and UGVs, integrating real-time visual detection, intelligent path planning, and adaptive coordination. The UAV, equipped with the YOLOv8 deep learning model, performed solar panel detection and anomaly localization. At the same time, the UGV provided logistical support and dynamic repositioning capabilities, extending the UAV's operational range and autonomy.

The architecture was structured into a multi-layered control framework, relying on a GCS for centralized decision-making, telemetry supervision, and communication management using PX4 and MAVLink within the ROS middleware. The proposed system demonstrated effective real-time collaboration between UAV and UGV, including autonomous landing and recharge procedures based on battery status and task demands.

Additionally, a comprehensive study was conducted to assess the impact of training data compositionreal, simulated, and hybridon the YOLOv8 model's performance. Results showed that using a balanced mix of real and synthetic data significantly improves model generalization and reliability across simulated, real-world, and hybrid scenarios. Models trained exclusively on one domain showed limitations, especially when applied to unseen environments, while the hybrid-trained models achieved high mAP and F1 Scores.

From a quantitative perspective, the results obtained in this work demonstrate the effectiveness of the proposed framework and methodologies. The YOLOv8-based detector achieved consistent performance in solar panel detection, achieving an F1-score of 0.91 and low false-positive and false-negative rates, indicating strong reliability for visual inspection tasks. The comparative analysis of different training data compositions showed that models trained with balanced combinations of real and synthetic images, particularly the 50% real and 50% simulated configuration, exhibited superior generalization across real, simulated, and hybrid test sets. These numerical results confirm that the joint use of real and synthetic data significantly enhances model robustness and supports the feasibility of simulation-based data generation as a complementary strategy for developing autonomous robotic inspection systems for photovoltaic farms. Besides, the combination of onboard detection, optimized inspection paths via Genetic Algorithms, and coordinated multi-robot execution proved to be an effective strategy for scalable and intelligent solar panel inspection.

V.2 Future Proposals and Limitations

Although the proposed framework demonstrated robust performance in simulation, several limitations must be addressed before full deployment in real-world solar farm environments. These limitations are mainly associated with sensing reliability under outdoor conditions, communication latency between agents, computational scalability of the GA, and the generalization capacity of the deep learning model when exposed to unseen data.

First, the current study was entirely validated in a simulated ROS with Gazebo environment, which, despite its realism, cannot fully capture the stochastic and unpredictable factors present in real operation, such as wind gusts, lighting variability, or GPS drift. Therefore, the perception and control modules may require fine-tuning once real data are incorporated.

Second, the GA optimization, while efficient for the test scenarios, has a computational cost that scales with both the number of inspection targets and the population size. Although the algorithm achieved convergence within the 10-second runtime limit, scalability to larger solar farms or multi-UAV systems may require additional parallelization strategies or the use of hybrid metaheuristics.

Third, the YOLO detection model, although exhibiting strong performance on hybrid datasets, still faces challenges related to domain shift when tested exclusively on unseen real environments. The precision and recall of detections are sensitive to lighting conditions and the visual texture of panels, suggesting that further data augmentation and domain adaptation techniques are necessary for improved robustness.

Finally, communication between the UAV and UGV currently relies on idealized MAVLink links within ROS simulation. In field applications, packet loss, bandwidth limitations, and latency can impact real-time coordination. Therefore, an evaluation of communication QoS is essential for safe and reliable mission execution.

Although the proposed UAVUGV collaborative framework was extensively validated in a realistic ROSGazebo simulation environment, some limitations inherent to simulation-based studies must be acknowledged. While simulation provides a safe, controllable, and cost-effective platform for developing and validating complex multi-robot systems, it cannot fully reproduce all the uncertainties and dynamics present in real-world deployments.

To overcome these limitations and expand the applicability of the proposed system, several future directions are envisioned:

- **Real-world validation:** Conduct experiments in actual photovoltaic plants to assess detection robustness, navigation accuracy, and endurance under realistic environmental conditions.
- **Thermal anomaly detection:** Integrate infrared cameras and multimodal fusion to enable simultaneous visual and thermal analysis for fault classification.

- **Advanced multi-robot coordination:** Extend the cooperative framework to multiple UAVs and UGVs operating concurrently, leveraging distributed task allocation algorithms and swarm-based GA variants.
- **Hybrid optimization approaches:** Combine GA with local search or reinforcement learning methods to improve convergence speed and adaptiveness in dynamic scenarios.
- **Real-time onboard inference:** Deploy optimized YOLOv11 models on embedded GPUs to enable fully autonomous operation without reliance on the GCS.
- **Dataset expansion and domain adaptation:** Enlarge the training corpus with diverse real-world datasets and synthetic imagery, incorporating domain adaptation and transfer learning to enhance model generalization across different PV farm configurations.

Bibliography

3D Robotics. *IRIS Drone*. 2014. Online. Available at: <https://3dr.com/support/products/iris/>. (Cited on page 26.)

ABUBAKAR, A.; ALMEIDA, C. F. M.; GEMIGNANI, M. Review of artificial intelligence-based failure detection and diagnosis methods for solar photovoltaic systems. *Machines*, MDPI, v. 9, n. 12, p. 328, 2021. (Cited on page 21.)

AGHAEI, M. et al. Image resolution and defects detection in pv inspection by unmanned technologies. In: IEEE. *2016 IEEE Power and Energy Society General Meeting (PESGM)*. [S.l.], 2016. p. 1–5. (Cited on page 21.)

BERGER, G. S. et al. Cooperative heterogeneous robots for autonomous insects trap monitoring system in a precision agriculture scenario. *Agriculture*, MDPI, v. 13, n. 2, p. 239, 2023. (Cited on pages 16, 20, 22 e 23.)

CARRENO, Y. et al. Task allocation strategy for heterogeneous robot teams in offshore missions. In: *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*. [S.l.: s.n.], 2020. p. 222–230. (Cited on pages 20 e 23.)

CASTRO, G. G. d. et al. Adaptive path planning for fusing rapidly exploring random trees and deep reinforcement learning in an agriculture dynamic environment uavs. *Agriculture*, MDPI, v. 13, n. 2, p. 354, 2023. (Cited on pages 15 e 16.)

CASTRO, G. G. de et al. Heterogeneous multi-robot collaboration for coverage path planning in partially known dynamic environments. *Machines*, MDPI, v. 12, n. 3, p. 200, 2024. (Cited on pages 16, 21, 22 e 23.)

CHAKRAA, H. et al. Optimization techniques for multi-robot task allocation problems: Review on the state-of-the-art. *Robotics and Autonomous Systems*, Elsevier, p. 104492, 2023. (Cited on page 20.)

CHEN, S. et al. Hybrid susd-based task allocation for heterogeneous multi-robot teams. In: IEEE. *2023 IEEE International Conference on Robotics and Automation (ICRA)*. [S.l.], 2023. p. 1400–1406. (Cited on page 20.)

CHERAGHI, A. R.; SHAHZAD, S.; GRAFFI, K. Past, present, and future of swarm robotics. In: SPRINGER. *Intelligent Systems and Applications: Proceedings of the 2021 Intelligent Systems Conference (IntelliSys) Volume 3*. [S.l.], 2022. p. 190–233. (Cited on page 15.)

FELSCH, T. et al. Robotized inspection of vertical structures of a solar power plant using ndt techniques. *Robotics*, MDPI, v. 4, n. 2, p. 103–119, 2015. (Cited on pages 21 e 23.)

GREEN, D. R. et al. Chapter 21 - using low-cost uavs for environmental monitoring, mapping, and modelling: Examples from the coastal zone. In: KRISHNAMURTHY, R. et al. (Ed.). *Coastal Management*. Academic Press, 2019. p. 465–501. ISBN 978-0-12-810473-6. Available at: <https://www.sciencedirect.com/science/article/pii-/B9780128104736000224>https://www.sciencedirect.com/science/article/pii/B9780128104736000224. (Cited on page 15.)

GRIMACCIA, F.; LEVA, S.; NICCOLAI, A. Pv plant digital mapping for modules defects detection by unmanned aerial vehicles. *IET Renewable Power Generation*, Wiley Online Library, v. 11, n. 10, p. 1221–1228, 2017. (Cited on page 21.)

HUANG, Y.; ZHANG, Y.; XIAO, H. Multi-robot system task allocation mechanism for smart factory. In: IEEE. *2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*. [S.l.], 2019. p. 587–591. (Cited on page 20.)

ISHII stpete. *Solar Panels YOLOv8 Train and Predict*. 2023. https://www.kaggle.com/code/stpeteishii/solar-panels-yolov8-train-and-predict/input?select=train_test_poznan_2018_2020_second_model. Accessed: 2025-03-19. (Cited on pages xi, 30 e 31.)

KASHKOLI, M. D. et al. A synthetic data approach for object detection in super low resolution images. In: *Proceedings of the 2024 International Conference on Innovation in Artificial Intelligence*. [S.l.: s.n.], 2024. p. 86–91. (Cited on page 16.)

KOENIG, N.; HOWARD, A. Design and use paradigms for gazebo, an open-source multi-robot simulator. In: IEEE. *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*. [S.l.], 2004. v. 3, p. 2149–2154. (Cited on page 18.)

LEE, H.-Y. et al. A distributed dynamic framework to allocate collaborative tasks based on capability matching in heterogeneous multirobot systems. *IEEE Transactions on Cognitive and Developmental Systems*, IEEE, v. 16, n. 1, p. 251–265, 2023. (Cited on pages 20 e 23.)

LI, L. et al. Fast task allocation of heterogeneous robots with temporal logic and inter-task constraints. *IEEE Robotics and Automation Letters*, IEEE, 2023. (Cited on page 20.)

LIDBERG, L. *Object Detection using deep learning and synthetic data*. 2018. (Cited on page 16.)

MANFREDA, S. et al. Assessing the accuracy of digital surface models derived from optical imagery acquired with unmanned aerial systems. *Drones*, v. 3, n. 1, 2019. ISSN 2504-446X. Available at: <https://www.mdpi.com/2504-446X/3/1/15>https://www.mdpi.com/2504-446X/3/1/15. (Cited on page 15.)

MAVLink Contributors. *MAVLink Protocol*. 2022. Online. Available at: <https://mavlink.io/en-/>https://mavlink.io/en/. (Cited on page 26.)

MAYYA, S. et al. Resilient task allocation in heterogeneous multi-robot systems. *IEEE Robotics and Automation Letters*, IEEE, v. 6, n. 2, p. 1327–1334, 2021. (Cited on page 20.)

MERIBOUT, M. et al. Solar panel inspection techniques and prospects. *Measurement*, Elsevier, v. 209, p. 112466, 2023. (Cited on page 21.)

MIRJALILI, S.; MIRJALILI, S. Genetic algorithm. *Evolutionary algorithms and neural networks: theory and applications*, Springer, p. 43–55, 2019. (Cited on page 30.)

MORANDO, L. et al. Thermal and visual tracking of photovoltaic plants for autonomous uav inspection. *Drones*, MDPI, v. 6, n. 11, p. 347, 2022. (Cited on page 15.)

Mornsun Power. *Exploring the 3.3V to 5V Conversion: Methods and Solutions*. 2023. https://www.mornsun-power.com/html/news-detail/blog-posts/213.html. Accessed: 2024-09-29. (Cited on page 21.)

NARVÁEZ, E. et al. Autonomous vtol-uav docking system for heterogeneous multirobot team. *IEEE Transactions on Instrumentation and Measurement*, IEEE, v. 70, p. 1–18, 2020. (Cited on page 15.)

OLAYIWOLA, O.; ELSDEN, M.; DHIMISH, M. Robotics, artificial intelligence, and drones in solar photovoltaic energy applicationssafe autonomy perspective. *Safety*, MDPI, v. 10, n. 1, p. 32, 2024. (Cited on page 15.)

PINTO, M. F. et al. A framework for analyzing fog-cloud computing cooperation applied to information processing of uavs. *Wireless Communications and Mobile Computing*, Hindawi, v. 2019, 2019. (Cited on page 15.)

PINTO, M. F. et al. Case-based reasoning approach applied to surveillance system using an autonomous unmanned aerial vehicle. In: *XXXX*. [S.l.: s.n.], 2017. p. 1324–1329. (Cited on page 15.)

PROROK, A.; HSIEH, M. A.; KUMAR, V. The impact of diversity on optimal control policies for heterogeneous robot swarms. *IEEE Transactions on Robotics*, IEEE, v. 33, n. 2, p. 346–358, 2017. (Cited on page 20.)

PRUTHVIRAJ, U. et al. Solar photovoltaic hotspot inspection using unmanned aerial vehicle thermal images at a solar field in south india. *Remote Sensing*, MDPI, v. 15, n. 7, p. 1914, 2023. (Cited on pages 21, 22 e 23.)

PUGLIESE, L. D. P.; GUERRIERO, F.; MACRINA, G. Using drones for parcels delivery process. *Procedia Manufacturing*, v. 42, p. 488–497, 2020. ISSN 2351-9789. International Conference on Industry 4.0 and Smart Manufacturing (ISM 2019). Available at: <https://www.sciencedirect.com/science/article-/pii/S2351978920305928>https://www.sciencedirect.com/science/article/pii/S2351978920305928. (Cited on page 15.)

RAMÍREZ, I. S.; CHAPARRO, J. R. P.; MÁRQUEZ, F. P. G. Unmanned aerial vehicle integrated real time kinematic in infrared inspection of photovoltaic panels. *Measurement*, Elsevier, v. 188, p. 110536, 2022. (Cited on page 15.)

RAMOS, G. S. et al. Arcog-net: An aerial robot cognitive network architecture for swarm applications development. *IEEE Access*, IEEE, 2024. (Cited on pages 15 e 17.)

RAMOS, G. S. et al. Hybrid methodology based on computational vision and sensor fusion for assisting autonomous uav on offshore messenger cable transfer operation. *Robotica*, Cambridge University Press, v. 40, n. 8, p. 27862814, 2022. (Cited on page 15.)

RAMOS, G. S. et al. Hybrid methodology based on computational vision and sensor fusion for assisting autonomous uav on offshore messenger cable transfer operation. *Robotica*, Cambridge University Press, v. 40, n. 8, p. 2786–2814, 2022. (Cited on page 16.)

ROLDÁN, J. J. et al. Heterogeneous multi-robot system for mapping environmental variables of greenhouses. *Sensors*, MDPI, v. 16, n. 7, p. 1018, 2016. (Cited on pages 20, 22 e 23.)

SHI, Y. et al. Adaptive informative sampling with environment partitioning for heterogeneous multi-robot systems. In: IEEE. *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. [S.l.], 2020. p. 11718–11723. (Cited on page 15.)

SILVA, Y. *Annotated images of simulated solar panels on the gazebo.* IEEE Dataport, 2025. Available at: <https://dx.doi.org/10.21227/s6hs-8r48>https://dx.doi.org/10.21227/s6hs-8r48. (Cited on pages xi, 31 e 40.)

SILVA, Y. M. D. et al. Computer vision based path following for autonomous unmanned aerial systems in unburied pipeline onshore inspection. *Drones*, MDPI, v. 6, n. 12, p. 410, 2022. (Cited on page 17.)

SOUSA, L. C. et al. Obstacle avoidance technique for mobile robots at autonomous human-robot collaborative warehouse environments. *Sensors*, MDPI, v. 25, n. 8, p. 2387, 2025. (Cited on page 17.)

Stanford Artificial Intelligence Laboratory et al., 2018. *Robotic Operating System*. Available at: <https://www.ros.org>https://www.ros.org. (Cited on page 18.)

TAHIR, A. et al. Swarms of unmanned aerial vehicles a survey. *Journal of Industrial Information Integration*, v. 16, p. 100106, 2019. ISSN 2452-414X. (Cited on page 15.)

ULTRALYTICS. *YOLOv8 Performance Metrics*. 2024. https://docs.ultralytics.com/models/yolov8/#performance-metrics. Accessed: 2025-03-19. (Cited on page 40.)

VENKATNIKHIL, A.; RAVICHANDRAN, S.; KUMAR, N. Rover robot for solar panel cleaning and monitoring. In: IEEE. *2022 IEEE North Karnataka Subsection Flagship International Conference (NKCon)*. [S.l.], 2022. p. 1–5. (Cited on pages 21 e 23.)

XIANG, C.; QI, C. R.; LI, B. Generating 3d adversarial point clouds. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. [S.l.: s.n.], 2019. p. 9136–9144. (Cited on page 16.)

YAHYA, Z. et al. Applied imagery pattern recognition for photovoltaic modules inspection: A review on methods, challenges and future development. *Sustainable Energy Technologies and Assessments*, Elsevier, v. 52, p. 102071, 2022. (Cited on page 21.)

ZEFRI, Y. et al. Thermal infrared and visual inspection of photovoltaic installations by uav photogrammetryapplication case: Morocco. *Drones*, MDPI, v. 2, n. 4, p. 41, 2018. (Cited on pages 21, 22 e 23.)