



**Instituto
Politécnico
Castelo Branco**

Escola Superior
de Tecnologia

Sistema inteligente de visão computacional para inspeção de centrais fotovoltaicas

João Filipe Campos Cabrito

20171417

Orientadores

João Manuel Leitão Pires Caldeira

Vasco Nuno da Gama de Jesus Soares

Trabalho apresentado à Escola Superior de Tecnologia de Castelo Branco do Instituto Politécnico de Castelo Branco, realizado sob a orientação científica do Doutor João Manuel Leitão Pires Caldeira e coorientação do Doutor Vasco Nuno da Gama de Jesus Soares, do Instituto Politécnico de Castelo Branco.

Janeiro e 2026

Composição do júri

Presidente do júri

Doutora, Mónica Isabel Teixeira da Costa

Professor Adjunto do Instituto Politécnico de Castelo Branco

Vogais

Doutor João Manuel Leitão Pires Caldeira

Professor Adjunto do Instituto Politécnico de Castelo Branco

Ângela Cristina Marques de Oliveira

Professor Adjunto do Instituto Politécnico de Castelo Branco

Agradecimentos

Agradeço ao Professor Doutor João Manuel Leitão Pires Caldeira, meu orientador, e ao Professor Doutor Vasco Nuno da Gama de Jesus Soares, meu coorientador, por terem proposto este tema e por me terem dado liberdade para o desenvolver numa direção alinhada com o meu interesse pela visão computacional. Agradeço também a disponibilidade, o acompanhamento e o rigor das sugestões, que foram decisivos para transformar a ideia inicial num trabalho com método e resultados.

Agradeço à minha família e às pessoas mais próximas pelo apoio e pela paciência ao longo deste processo, especialmente nos períodos de maior exigência.

Resumo

As centrais fotovoltaicas são essenciais na transição energética, mas o seu desempenho é sensível ao estado superficial dos módulos. A acumulação de sujidade, como poeiras e excrementos de aves reduz a captação de radiação e pode aumentar a necessidade de manutenção. A inspeção manual é pouco escalável e variável, sobretudo quando se pretende monitorização frequente. Neste contexto, este trabalho propôs o desenvolvimento e a avaliação de um sistema de inspeção visual automática, baseado em visão computacional e aprendizagem profunda, para identificar padrões de sujidade em imagens RGB de painéis fotovoltaicos e apoiar limpeza seletiva e manutenção preventiva.

O trabalho iniciou-se com uma revisão sistemática da literatura segundo a metodologia PRISMA, permitindo enquadrar o estado da arte e apoiar decisões de desenho experimental. Com base nessa análise, foram definidos requisitos e estruturado um conjunto de dados anotado com classes orientadas ao problema, incluindo situações de superfície limpa, sujidade difusa (poeira) e sujidade localizada (excrementos). Foram ainda consideradas técnicas de aumento de dados para melhorar a robustez face a variações de iluminação e condições ambientais.

Numa fase seguinte o trabalho avançou para uma componente experimental comparando abordagens com perfis distintos: variantes da família YOLO, adequadas a deteção eficiente, e uma abordagem baseada em transformers (DINOv2), com foco na segmentação e produção de máscaras mais informativas. A avaliação recorreu a métricas como *mAP*, precisão, *recall* (revocação), *F1-score* e velocidade de inferência, tendo sido complementadas por análise qualitativa das inferências, de forma a interpretar padrões de erro e estabilidade em diferentes cenários.

Os resultados confirmaram a viabilidade de uma solução automatizada, com melhor desempenho quando as ocorrências mostraram visualmente dominantes e contrastantes, e maior sensibilidade em situações de sujidade difusa, reflexos, sombras ou coexistência de classes. No conjunto, o trabalho sustentou a seleção do modelo mais adequado ao objetivo do sistema e estabeleceu uma base para evolução futura, com reforço e normalização do dataset, testes adicionais de generalização e integração do output em indicadores acionáveis para suporte à decisão em operação e manutenção.

Palavras-chave

Visão Computacional; Aprendizagem Profunda; Módulos Fotovoltaicos; Deteção e Segmentação; Sujidade

Abstract

Photovoltaic (PV) power plants play a key role in the energy transition, yet their performance is highly sensitive to the surface condition of PV modules. The accumulation of soiling—such as dust and bird droppings—reduces irradiance capture and may increase maintenance needs. Manual inspection is difficult to scale and often inconsistent, particularly when frequent monitoring is required. In this context, this work proposes the development and evaluation of an automatic visual inspection system, based on computer vision and deep learning, to identify soiling patterns in RGB images of photovoltaic panels and to support selective cleaning and preventive maintenance.

The study began with a systematic literature review following the PRISMA methodology, which framed the state of the art and informed the experimental design choices. Based on this analysis, system requirements were defined and an annotated dataset was structured with problem-oriented classes, including clean surfaces, diffuse soiling (dust), and localized soiling (bird droppings). Data augmentation techniques were also considered to improve robustness under varying illumination and environmental conditions.

The experimental phase compared approaches with distinct profiles: YOLO-family variants, suitable for efficient object detection, and a transformer-based approach (DINOv2), focusing on segmentation and the generation of more informative masks. Evaluation employed metrics such as mean Average Precision (mAP), precision, recall, F1-score, and inference speed, complemented by a qualitative analysis of model outputs to interpret error patterns and stability across different scenarios.

The results confirm the feasibility of an automated solution, with best performance when soiling instances are visually dominant and high-contrast, and increased sensitivity to challenging conditions such as diffuse soiling, reflections, shadows, or class co-occurrence. Overall, the work supports the selection of the model most suited to the system's objectives and establishes a foundation for future improvements, including dataset reinforcement and normalization, additional generalization tests, and the integration of model outputs into actionable indicators for operation and maintenance decision support.

Keywords

Computer Vision; Deep Learning; Photovoltaic Modules; Detection and Segmentation; Soiling

Índice geral

1. Introdução	1
1.1 Âmbito e Definição do Problema	1
1.2 Objetivo	3
1.3 Contribuições.....	5
1.4 Organização do Documento	6
2. Revisão da Literatura	8
2.1 Metodologia de Seleção de Artigos	8
2.2 Processo de Seleção dos Estudos	9
2.3 Caracterização dos Estudos Incluídos	11
2.4 Técnicas de Melhoria nas Abordagens	16
2.5 Conjuntos de Dados Utilizados nos Estudos Analisados	20
2.6 Comparação Crítica e Escolha do Modelo	23
3. Enquadramento Teórico	28
3.1 Visão Computacional para Inspeção de Centrais Fotovoltaicas	28
3.2 Modelos de Detecção de Objetos	31
4. Avaliação de Desempenho e Seleção do Modelo	39
4.1 Descrição do <i>Dataset</i>	39
4.2 Métricas de Desempenho.....	45
4.3 Cenário de Implementação	50
4.4 Resultados e Discussão	51
5. Desafios e Oportunidades	72
6. Conclusão e Trabalho Futuro.....	74
6.1 Conclusão	74
6.2 Trabalho Futuro.....	74
7. Referências	76

Índice de figuras

Figura 1 - Fluxograma <i>PRISMA</i> do processo de identificação, triagem, elegibilidade e inclusão dos artigos selecionados.	10
Figura 2 - Distribuição temporal dos seis estudos incluídos na revisão sistemática.....	14
Figura 3 - Desempenho médio em termos de <i>mAP</i> dos modelos utilizados nos seis estudos incluídos na revisão sistemática.....	15
Figura 4 - Relação entre precisão (<i>mAP</i>) e velocidade de inferência (FPS) dos modelos utilizados nos seis estudos incluídos na revisão sistemática	16
Figura 5 - Exemplos de técnicas clássicas de data <i>augmentation</i> (<i>flip</i> , rotação, ruído, <i>cutout</i>) aplicadas a uma imagem de defeito. Adaptado de [16]	17
Figura 6 - Processo de construção do <i>dataset</i> “Naïve/Realistic” com recorte de regiões de sujidade e colagem sobre novos fundos de painéis. Adaptado de [17]	18
Figura 7 - Exemplos de imagens de sujidade, máscaras de <i>ground truth</i> e máscaras de transparência associadas. Adaptado de [17]	19
Figura 8 - Distribuição das arquiteturas de <i>deep learning</i> utilizadas nos estudos analisados.....	24
Figura 9 - Estrutura genérica de uma rede neuronal convolucional (<i>CNN</i>), composta por camadas de convolução, pooling e classificação. Adaptado de [7]	32
Figura 10 - Esquema simplificado dos detetores <i>Two-Stage</i> (<i>Faster/Mask-RCNN</i>). Adaptado de [7].....	33
Figura 11 - Classificação geral dos modelos de deteção de objetos e distinção entre abordagens <i>Two-Stage</i> e <i>One-Stage</i> . Adaptado de [7]	34
Figura 12 - Arquitetura detalhada do modelo <i>YOLOv8</i> , composta pelos módulos <i>Backbone</i> , <i>Neck</i> e <i>Head</i> , e incorporando blocos <i>C2f</i> , operações de upsampling e perdas independentes de regressão (<i>BBox Loss</i>) e classificação (<i>Cls Loss</i>). Adaptado [9].....	37
Figura 13 - Arquitetura do modelo <i>SSD</i> melhorado com integração de módulos de atenção e deteção multi-escala. Adaptado de [8].....	38
Figura 14 - Exemplos de anotações no <i>Roboflow</i> para a classe <i>dust-solar-panel</i>	41
Figura 15 - Exemplos de anotações no <i>Roboflow</i> para a classe <i>clean-solar-panel</i>	42
Figura 16 - Exemplos de anotações no <i>Roboflow</i> para a classe <i>bird-drop</i>	42
Figura 17 - Evolução do <i>mAP</i> em validação (<i>YOLOv8x</i>)	52
Figura 18 - Evolução do <i>mAP</i> em validação (<i>YOLOv11x</i>).....	53
Figura 19 - Evolução do <i>mAP</i> (box e mask) em validação (<i>YOLOv11-seg</i>).....	54
Figura 20 - Evolução das curvas de aprendizagem (<i>DINOv2</i>)	56
Figura 21 - <i>IoU</i> por classe em validação (<i>DINOv2</i>)	57
Figura 22 - <i>YOLOv8x</i> inferência <i>clean-solar-panel</i>	58
Figura 23 - <i>YOLOv8x</i> inferência <i>bird-drop</i>	59
Figura 24 - <i>YOLOv8x</i> inferência <i>dust-solar-panel</i>	60

Figura 25 - YOLOv11x inferência clean-solar-panel	61
Figura 26 - YOLOv11x inferência bird-drop.....	62
Figura 27 - YOLOv11x inferência dust-solar-panel	63
Figura 28 - YOLOv11-seg inferência clean-solar-panel	64
Figura 29 - YOLOv11-seg inferência bird-drop.....	65
Figura 30 - YOLOv11-seg inferência dust-solar-panel.....	66
Figura 31 - DINOv2 inferência clean-solar-panel	67
Figura 32 - DINOv2 inferência bird-drop.....	68
Figura 33 - DINOv2 inferência dust-solar-panel.....	69

Lista de tabelas

Tabela 1 - Cronograma de tarefas.....	5
Tabela 2 - Resumo dos estudos incluídos na revisão do estado da arte	12
Tabela 3 - Distribuição de imagens no split treino, validação e teste	44
Tabela 4 - Distribuição de anotações por classe e por split no dataset final	45
Tabela 5 - Desempenho global em validação (deteção).....	52
Tabela 6 - Desempenho global em validação (segmentação) — YOLOv11-seg	54
Tabela 7 - Desempenho em validação (segmentação) — DINOv2.....	55

Lista de equações

Equação 1 - Fórmula para o cálculo de Precision.....	47
Equação 2 - Fórmula para o cálculo de Recall	47
Equação 3 - Fórmula para o cálculo de Average Precision.....	48
Equação 4 - Fórmula para o cálculo de mAP.....	48
Equação 5 - Fórmula para o cálculo do mAP50-95.....	49

Lista de abreviaturas, siglas e acrónimos

AP	Average Precision
API	Application Programming Interface
BBox	Bounding Box
CNN	Convolutional Neural Network (Rede Neuronal Convolucional)
CSPDarknet	Cross Stage Partial Darknet
DL	Deep Learning (Aprendizagem profunda)
Faster R-CNN	Faster Region-based Convolutional Neural Network
FP	False Positive
FPN	Feature Pyramid Network
GPU	Graphics Processing Unit
HSV	Hue, Saturation, Value
IEEE	Institute of Electrical and Electronics Engineers
IoU	Intersection over Union
IR	Infrared (Infravermelho)
LR	Learning Rate
mAP	mean Average Precision
Mask R-CNN	Mask Region-based Convolutional Neural Network
mIoU	mean Intersection over Union
NMS	Non-Maximum Suppression
O&M	Operação e Manutenção
PRISMA	Preferred Reporting Items for Systematic Reviews and Meta-Analyses
PV	Photovoltaic (Fotovoltaico)
R-CNN	Region-based Convolutional Neural Network
RGB	Red, Green, Blue
ROI	Region of Interest
RoIAlign	Region of Interest Align
RoIPool	Region of Interest Pooling
RPN	Region Proposal Network
Soft-NMS	Soft Non-Maximum Suppression
SSD	Single Shot Detector
TP	True Positive
UAV	Unmanned Aerial Vehicle (drone)

UPAC	Unidade de Produção para Autoconsumo
VGG-16	Visual Geometry Group 16-layer
ViT	Vision Transformer
VRAM	Video Random Access Memory
YAML	YAML Ain't Markup Language
YOLO	You Only Look Once

1. Introdução

Este primeiro capítulo estabeleceu o enquadramento geral do trabalho, apresentando a temática principal e o contexto onde este se insere, bem como as motivações que conduziram à sua realização. Foram definidos o problema a investigar, os objetivos que orientaram o seu desenvolvimento e a metodologia concebida para os alcançar de forma sistemática e coerente. Adicionalmente, foi clarificado o âmbito do estudo e a sua relevância no domínio em que se insere. Por fim, descreveu-se a organização do documento, de modo a facilitar a compreensão da estrutura e da progressão dos conteúdos.

1.1 Âmbito e Definição do Problema

A aceleração da transição energética tem colocado a energia solar fotovoltaica no centro das estratégias de descarbonização em Portugal, com forte crescimento recente da capacidade instalada e da produção elétrica a partir de fontes renováveis. Em 2024, a produção de eletricidade a partir de fontes renováveis atingiu cerca de 36,7 TWh, garantindo aproximadamente 71 % do consumo nacional de eletricidade, o valor mais elevado de sempre no sistema elétrico português [1]. A mesma fonte indica que a produção fotovoltaica cresceu cerca de 37 % face a 2023 e já representa perto de 10 % do consumo elétrico nacional, refletindo a forte expansão desta tecnologia. Em paralelo, as estatísticas oficiais mostram que a potência instalada em fotovoltaico (PV) é a que mais tem crescido no sistema elétrico, tendo atingido 5,6 GW em 2023 e continuando a aumentar em 2024 [2].

Para além das grandes centrais ligadas à rede, o segmento do autoconsumo tem assumido um papel relevante na disseminação da energia solar. O relatório mais recente da Entidade Reguladora dos Serviços Energéticos (ERSE) sobre autoconsumo indica que, no final de 2024, existiam em Portugal continental cerca de 237 mil autoconsumidores com aproximadamente 1,8 GW de potência instalada em unidades de produção para autoconsumo (UPAC), maioritariamente baseadas em módulos PV [3]. Esta massificação, tanto em larga escala como em contexto distribuído, torna a operação e manutenção (O&M) dos sistemas PV um desafio central para a fiabilidade e a sustentabilidade económica do setor.

Nesta expansão, a fiabilidade dos módulos PV é crítica. Entre os fatores que degradam o desempenho, a sujidade (*soiling*) – acumulação de poeiras, partículas e excrementos – destaca-se como uma das causas mais influentes após a irradiância, sendo responsável por perdas anuais típicas de 3–5% (podendo ser superiores consoante o clima e a limpeza), com impactos económicos relevantes para a exploração das centrais [4]. Estas perdas traduzem-se em redução direta da energia disponível para injeção na rede e na necessidade de intensificar estratégias de manutenção para recuperar o desempenho dos sistemas.

Os métodos tradicionais de inspeção (observação visual, medições elétricas pontuais e campanhas de manutenção calendarizadas) são morosos, caros e pouco escaláveis em grandes centrais. Em contraste, práticas modernas de O&M recorrem à termografia infravermelha – frequentemente com *drones* – para detetar anomalias (p. ex., hot spots, delaminações) e mapear áreas com perdas de desempenho, permitindo intervenções direcionadas e redução do tempo de inatividade. Estas abordagens estão já disseminadas no contexto português em empresas e projetos de inspeção, bem como em estudos técnicos nacionais [5]. No entanto, continuam a depender significativamente da interpretação humana das imagens obtidas e são realizadas de forma episódica, o que limita a capacidade de monitorização contínua do estado dos módulos.

Paralelamente, a digitalização dos processos de manutenção exige alinhar a inspeção com referenciais técnicos e regulatórios. A norma internacional IEC 62446-1 estabelece requisitos para ensaios, documentação e manutenção de sistemas PV ligados à rede – aplicáveis a comissionamento e testes periódicos – enquanto a regulamentação nacional do autoconsumo (DL n.º 15/2022) enquadra a operação de instalações e comunidades de energia. A integração de inspeção visual automática baseada em visão computacional e deep learning (DL) com estas práticas e normas pode aumentar a cobertura, reduzir custos e padronizar a qualidade da manutenção [6].

Deste modo, o problema aqui abordado consistiu em desenvolver um sistema inteligente de inspeção automática para identificar e classificar sujidade e outras anomalias superficiais nos módulos PV a partir de imagens *RGB*, apoiando a decisão de limpeza e a manutenção preventiva. Pretendeu-se, com isso, mitigar perdas por *soiling*, otimizar planos de limpeza e aumentar a disponibilidade dos ativos, com relevância direta para o contexto português de rápida expansão da geração solar.

Com a crescente disponibilidade de dados visuais e o avanço das técnicas de *visão computacional* (CV), tem-se verificado uma transição gradual das abordagens convencionais, baseadas em inspeções manuais e termografia isolada, para métodos automáticos suportados por algoritmos de DL. Modelos de deteção de objetos baseados em redes neuronais convolucionais (*CNN*), como as arquiteturas da família *YOLO* e *SSD*, têm demonstrado elevado potencial na inspeção de módulos PV, permitindo identificar padrões de sujidade e defeitos com maior rapidez e precisão. Estudos recentes comprovam a eficácia do *YOLOv5* na deteção de poeira, folhas e excrementos em painéis solares, apresentando desempenhos superiores em métricas de *Mean Average Precision* (*mAP*) e tempos de inferência reduzidos [7]. De forma complementar, métodos baseados em *SSD* aperfeiçoados com mecanismos de atenção e *transfer learning* têm mostrado melhorias significativas na identificação de anomalias estruturais em módulos PV, mantendo elevada eficiência computacional [8].

No contexto português, o crescimento acelerado da energia solar e a aposta na digitalização da manutenção reforçam a pertinência destas soluções inteligentes. Segundo dados da Associação Portuguesa de Energias Renováveis (APREN, REN – Redes Energéticas Nacionais), 2024 registou um novo máximo histórico de produção elétrica renovável, com a energia solar a representar uma fatia crescente da potência instalada nacional, confirmando a relevância da inspeção automatizada para a sustentabilidade e eficiência operacional das centrais fotovoltaicas [1].

Esta evolução tecnológica abriu caminho ao desenvolvimento de sistemas autónomos de diagnóstico visual, aplicáveis tanto em imagens *RGB* obtidas por *drones* como em câmaras fixas instaladas nas centrais, permitindo uma monitorização contínua e mais fiável do desempenho dos painéis.

1.2 Objetivo

O objetivo central deste projeto foi propor e avaliar uma solução automatizada de inspeção de centrais fotovoltaicas baseada em técnicas de visão computacional e aprendizagem profunda. Pretendeu-se desenvolver um sistema inteligente capaz de detetar e classificar automaticamente o tipo e o grau de sujidade presente nos painéis — como poeira ou excrementos de aves — de forma a apoiar o planeamento de ações de limpeza particularizada e a manutenção preventiva dos módulos. A ambição não foi apenas identificar se um painel está “sujo” ou “limpo”, mas fornecer informação estruturada que permita priorizar intervenções e quantificar o impacto potencial da sujidade no desempenho.

Do ponto de vista técnico, o sistema assentou na análise de imagens *RGB* dos módulos PV, recorrendo a modelos de deteção de objetos baseados em *CNN*, com especial enfoque em arquiteturas recentes da família *YOLO*, pela sua capacidade de combinar rapidez de inferência com elevada precisão. O projeto envolveu a construção ou adaptação de um conjunto de dados anotado para o contexto PV, o treino e afinação de um modelo de deteção ajustado às características visuais dos painéis (textura, reflexos, variação de iluminação) e a definição de métricas objetivas de avaliação — *mAP*, precisão, *recall* e tempos médios de inferência — de forma a quantificar rigorosamente o desempenho da solução proposta e a compará-lo com valores de referência na literatura.

Em termos funcionais, o objetivo passou pela integração deste modelo num protótipo de sistema de suporte à decisão, capaz de processar imagens de forma automática e devolver, para cada painel analisado, a localização e o tipo de sujidade ou anomalia superficial detetada. Esta abordagem visou reduzir a intervenção humana em tarefas repetitivas de inspeção, aumentar a fiabilidade e consistência das deteções e contribuir para a melhoria do desempenho energético global das instalações, assegurando simultaneamente uma gestão mais eficiente dos recursos afetos à O&M. Pretendeu-se que o protótipo permita, por exemplo,

identificar conjuntos de painéis com sujidade crítica, gerar listagens de intervenção e apoiar a definição de prioridades em planos de O&M.

Numa perspetiva aplicada, pretendeu-se ainda que a solução desenvolvida seja demonstrável em contexto próximo do real, através da integração do modelo num ambiente de teste com interface gráfica ou plataforma digital simples, que permita visualizar as deteções, gerar alertas e apoiar o registo do estado dos painéis ao longo do tempo. O objetivo não foi apenas comprovar a viabilidade técnica do modelo de deteção em ambiente controlado, mas também evidenciar o seu potencial de integração em fluxos de trabalho de O&M existentes, abrindo caminho a futuras extensões em cenários de inspeção contínua com *drones*, câmaras fixas ou sistemas embarcados em unidades móveis.

Em termos de delimitação, o projeto focou-se exclusivamente na análise de imagens *RGB* e na deteção de sujidade e anomalias superficiais visíveis, não abrangendo, por exemplo, a análise termográfica de hot spots nem a modelação elétrica detalhada do impacto dos defeitos. O sistema proposto foi concebido como prova de conceito e protótipo demonstrador, não como produto final pronto a ser industrializado, mas procurando desde já respeitar constrangimentos práticos relevantes, como a necessidade de tempos de processamento compatíveis com utilização em campo e a possibilidade de adaptação a diferentes centrais fotovoltaicas.

Para atingir este objetivo principal, identificaram-se as seguintes etapas no âmbito do desenvolvimento de Projeto I:

- Estudo do trabalho relacionado;
- Análise de requisitos;
- Preparação de dados e desenvolvimento experimental;
- Demonstração e testes funcionais;

Para a concretização das etapas definidas, foi delineado o plano de trabalhos e respetivo cronograma, composto pelas seguintes tarefas como apresentado na **Tabela 1**:

Tarefa 1 – Estudo do trabalho relacionado (Meses 1 a 3):

Revisão do estado da arte sobre inspeção visual de módulos PV e deteção/segmentação de sujidade com aprendizagem profunda, incluindo seleção de métricas e metodologias de avaliação. Realização de revisão sistemática segundo *PRISMA* e síntese crítica dos resultados para suportar decisões técnicas.

Tarefa 2 – Análise de requisitos (Meses 2 e 4):

Definição dos requisitos funcionais e não funcionais do sistema de inspeção automática (entrada de imagens, classificação/deteção/segmentação, visualização de resultados e registo), bem como critérios de desempenho e

restrições operacionais (robustez a iluminação, custo computacional, escalabilidade). Delimitação do âmbito e das classes consideradas.

Tarefa 3 – Preparação de dados e desenvolvimento experimental (Meses 3 a 5):

Construção/organização do *dataset* anotado e preparação do *pipeline* de treino e validação, incluindo aumento de dados e implementação e configuração dos modelos selecionados (YOLO para detecção/segmentação e abordagem baseada em *transformers*, *DINOv2*), com execução de treinos/experiências controladas.

Tarefa 4 – Demonstração e testes funcionais (Meses 5 a 6):

Avaliação quantitativa e qualitativa dos modelos com métricas (*mAP*, precisão, *recall*, *F1-score* e velocidade de inferência), análise de erros e discussão de limitações (variação de iluminação, reflexos, coexistência de classes, qualidade de anotação). Seleção fundamentada da abordagem mais adequada ao objetivo do sistema e definição de trabalho futuro para integração num protótipo demonstrador.

Tabela 1 - Cronograma de tarefas

	Mês 1	Mês 2	Mês 3	Mês 4	Mês 5	Mês 6
Tarefa 1						
Tarefa 2						
Tarefa 3						
Tarefa 4						

1.3 Contribuições

O desenvolvimento deste trabalho deu origem a um conjunto de contribuições tecnológicas que se articularam diretamente com os objetivos definidos na secção anterior. Em primeiro lugar, foi realizada uma revisão sistemática do estado da arte no domínio da inspeção automática de painéis PV com recurso a técnicas de visão computacional e aprendizagem profunda, permitindo mapear as principais abordagens existentes, os modelos mais utilizados e as métricas de avaliação adotadas. Em segundo lugar, a análise crítica desses estudos permitiu identificar e sistematizar limitações relevantes nas soluções atuais de detecção de sujidade em módulos solares, nomeadamente ao nível da generalização a diferentes tipos de sujidade, da robustez a variações de iluminação e enquadramento, e da escassez de conjuntos de dados anotados em contexto real. Em terceiro lugar, foi concebido e implementado um protótipo de sistema inteligente de inspeção visual, baseado num modelo de detecção de objetos de última geração, capaz de detetar e localizar diferentes tipos de sujidade em imagens *RGB* de painéis PV. Esta contribuição incluiu a construção e preparação de um conjunto de dados anotado, a definição

do *pipeline* de pré-processamento e *data augmentation*, bem como a integração do modelo num protótipo demonstrador com capacidade de visualização das deteções. Por fim, procedeu-se à avaliação experimental do desempenho do sistema proposto, recorrendo a métricas como *mAP*, precisão, *recall* e tempos de inferência, e discutindo os resultados à luz dos valores reportados na literatura. No seu conjunto, estas contribuições procuraram não apenas demonstrar a viabilidade técnica da abordagem proposta, mas também evidenciar o seu potencial para apoiar estratégias de limpeza seletiva e manutenção preventiva em centrais fotovoltaicas.

1.4 Organização do Documento

O presente relatório encontra-se organizado em seis capítulos que seguem uma lógica progressiva, desde o enquadramento conceptual do problema até à avaliação experimental e à seleção do modelo, terminando com uma reflexão crítica e conclusões. No primeiro capítulo é apresentada a introdução ao tema, enquadrando o impacto da sujidade em módulos PV e a necessidade de inspeção automática, sendo explicitados o problema, os objetivos do trabalho, as contribuições previstas e o plano global de desenvolvimento. O segundo capítulo apresenta a revisão sistemática da literatura, conduzida segundo a metodologia PRISMA, na qual são definidas as questões de investigação, os critérios de inclusão e exclusão, a estratégia de pesquisa e o processo de seleção dos estudos; são ainda caracterizados os artigos incluídos, analisados os datasets utilizados e discutidas comparativamente as abordagens reportadas, consolidando o estado da arte e os critérios que sustentam as opções tomadas nas fases experimentais. O terceiro capítulo é dedicado ao enquadramento teórico e reúne os fundamentos de visão computacional e aprendizagem profunda relevantes para este estudo, incluindo o papel das redes neuronais convolucionais, as principais abordagens de deteção de objetos (two-stage e one-stage), com destaque para modelos da família YOLO e arquiteturas alternativas, bem como o pipeline típico de treino e inferência, desde a preparação e normalização de dados até técnicas de *data augmentation* e estratégias de pós-processamento. O quarto capítulo constitui o núcleo de avaliação de desempenho e seleção do modelo, descrevendo de forma auditável a construção e caracterização do dataset (incluindo preparação, anotação e organização dos dados), as métricas de avaliação adotadas e a sua interpretação operacional, o cenário de implementação onde ocorre a comparação (ambiente, configurações, pipeline de treino e validação e condições mantidas constantes entre modelos), e, por fim, os resultados e discussão, onde se apresentam e interpretam comparações quantitativas e qualitativas, suportando a decisão final do modelo selecionado para o objetivo proposto. O quinto capítulo reúne os principais desafios e oportunidades identificados ao longo do trabalho, discutindo limitações do dataset, dificuldades típicas em condições reais (variações de iluminação, reflexos, ruído, dimensão de artefactos e generalização) e oportunidades de

melhoria técnica e metodológica. Por fim, o sexto capítulo apresenta as conclusões, sintetiza os contributos do estudo, explicita as limitações mais relevantes e define direções de trabalho futuro, com foco no reforço da robustez, na extensão experimental e na eventual aproximação a cenários de O&M em contexto real.

2. Revisão da Literatura

Este capítulo apresenta o estado da arte sobre a inspeção automática de centrais fotovoltaicas, com ênfase nas técnicas de visão computacional e DL aplicadas à detecção de sujidade e anomalias superficiais nos painéis solares.

A acumulação de poeiras, folhas e excrementos de aves (*soiling*) nos módulos solares é uma causa influente de perdas de rendimento, que tipicamente se situam entre 3–5% anualmente, podendo causar danos permanentes. Em contraste com os métodos de inspeção tradicionais (que são demorados e caros), a visão computacional e o DL permitem a análise automática de imagens para detetar anomalias, sujidade e falhas estruturais.

Foram analisados estudos publicados entre 2021 e 2025, redigidos em inglês, que abordam a detecção visual de poeira, sujidade ou sombras na superfície dos módulos, utilizando métodos de análise de imagem baseados em DL.

A revisão foi conduzida de acordo com as diretrizes da metodologia *Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA)*. Esta metodologia é reconhecida pela sua aplicação em revisões sistemáticas que exigem rigor e transparência na seleção de publicações. Esta abordagem sistemática, que envolveu a identificação, triagem e elegibilidade de estudos (resultando na inclusão de seis artigos), permitiu identificar as principais tendências e soluções mais recentes.

A aplicação concreta da metodologia *PRISMA* neste trabalho organiza-se da seguinte forma: na secção 2.1 apresentam-se as questões de investigação, os critérios de inclusão e a estratégia de pesquisa; na secção 2.2 descreve-se o processo de seleção dos estudos, seguindo as etapas de identificação, triagem, elegibilidade e inclusão; na secção 2.3 procede-se à caracterização global dos trabalhos incluídos; nas secções 2.4 e 2.5 analisa-se em detalhe o *pipeline* de processamento de dados e os conjuntos de dados utilizados; por fim, na secção 2.6 realiza-se uma comparação crítica das abordagens e justifica-se a escolha do modelo base a adotar no sistema proposto.

2.1 Metodologia de Seleção de Artigos

Com o objetivo de orientar a revisão da literatura, foram formuladas questões de investigação relacionadas com a inspeção automática de painéis PV e a aplicação de técnicas de visão computacional. Estas questões serviram de base à pesquisa e análise dos estudos selecionados:

Questão 1: Que métodos de visão computacional e aprendizagem profunda têm sido aplicados na detecção de sujidade em painéis PV?

Questão 2: Quais os modelos de detecção de objetos mais utilizados e que resultados apresentam em termos de desempenho e precisão?

Questão 3: Que técnicas complementares — como pré-processamento, *data augmentation* e pós-processamento — contribuem para melhorar a eficácia das deteções?

Adicionalmente foram definidos os seguintes critérios de inclusão:

Critério 1: Estudos publicados entre 2021 e 2025.

Critério 2: Estudos redigidos em inglês.

Critério 3: Trabalhos com texto integral disponível.

Critério 4: Estudos focados na deteção visual de anomalias físicas (pó, sujidade, folhas, sombras, etc.) em painéis PV.

Critério 5: Aplicação de técnicas de visão computacional ou aprendizagem profunda para análise de imagens.

A pesquisa foi efetuada utilizando a expressão realizada durante o mês de outubro de 2025:

(“photovoltaic” OR “solar panel”) AND (“soiling” OR “dust” OR “shading” OR “anomaly”) AND (“deep learning” OR “computer vision” OR “image processing” OR “object detection”)

2.2 Processo de Seleção dos Estudos

A pesquisa inicial permitiu identificar um total de 329 registos, distribuídos entre as bases de dados *IEEE Xplore* (251 publicações) e *ACM Digital Library* (78 publicações). Todos estes registos foram submetidos ao processo de filtragem definido pela metodologia *PRISMA*, que estrutura a seleção em etapas sucessivas de identificação, triagem, elegibilidade e inclusão, assegurando transparência e rigor na construção da revisão sistemática.

Numa fase preliminar, procedeu-se à verificação dos critérios mínimos de inclusão. Foram removidos 35 artigos publicados antes de 2021, de forma a garantir que a revisão incidia exclusivamente sobre avanços recentes na deteção visual de anomalias em módulos PV. Não se registaram duplicados entre as bases consultadas, nem publicações excluídas por falta de acesso integral ou por não estarem redigidas em inglês. Concluída esta primeira filtragem, permaneceram 294 artigos para a etapa seguinte.

A fase de triagem consistiu numa leitura cuidada de títulos e resumos. Nesta análise, foram excluídas 270 publicações cujo conteúdo não se enquadrava no propósito da investigação. Entre os motivos de exclusão encontravam-se estudos centrados em análise elétrica de falhas, previsão de produção energética, otimização operacional de sistemas PV ou abordagens de carácter estritamente teórico, sem aplicação prática de visão computacional ou algoritmos de

aprendizagem profunda. Após esta etapa, subsistiram 24 artigos considerados potencialmente relevantes e, por isso, selecionados para leitura integral.

A avaliação detalhada destes 24 estudos permitiu aplicar critérios adicionais de elegibilidade, direcionados para a relevância metodológica, atualidade tecnológica e alinhamento com os objetivos definidos. O processo conduziu à exclusão de 18 estudos: um deles por não empregar técnicas de visão computacional ou DL na deteção de anomalias; cinco por recorrerem a metodologias desatualizadas ou apresentarem resultados pouco expressivos face ao estado da arte; e doze por revelarem redundância metodológica, replicando abordagens já presentes noutros artigos incluídos.

Concluída a aplicação de todos os critérios, foram selecionados seis estudos para integrar a revisão sistemática. Estes constituíram o núcleo da análise e destacaram-se por apresentarem abordagens recentes baseadas em CNN e modelos da família *YOLO*, evidenciando melhorias significativas na deteção automática de sujidade, poeira, sombras e irregularidades estruturais na superfície dos módulos PV. O fluxograma *PRISMA* apresentado na **Figura 1** sintetiza graficamente o processo de afunilamento progressivo, desde o conjunto inicial de 329 publicações até à amostra final de seis estudos incluídos.

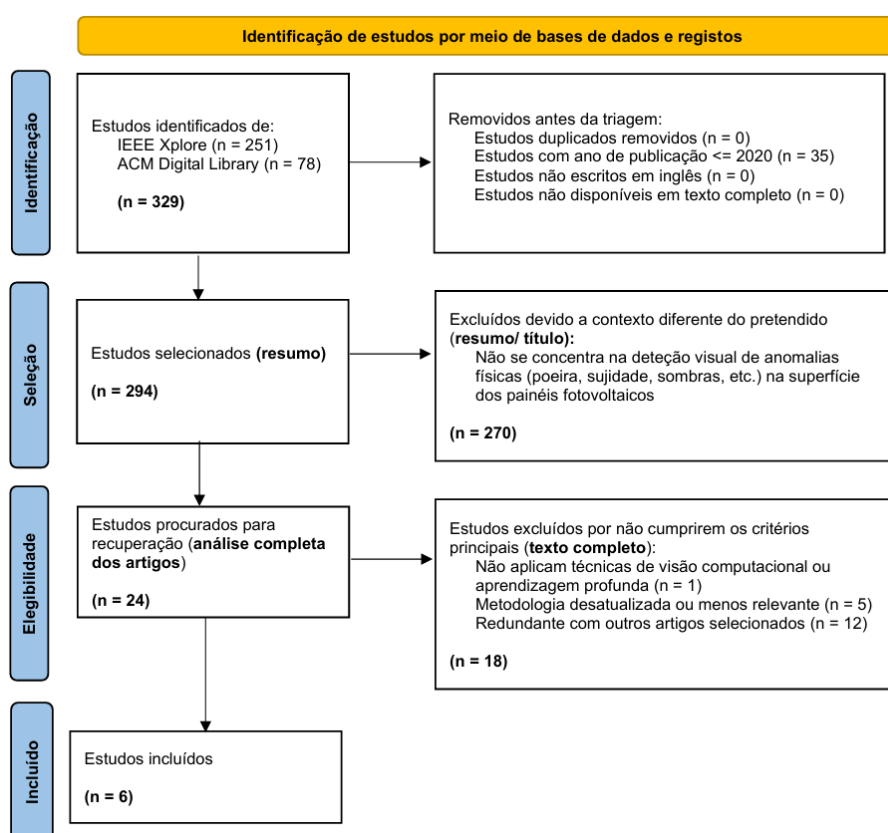


Figura 1 - Fluxograma *PRISMA* do processo de identificação, triagem, elegibilidade e inclusão dos artigos selecionados.

2.3 Caracterização dos Estudos Incluídos

Concluído o processo de seleção *PRISMA* e identificados os seis estudos que cumpriram integralmente os critérios definidos, tornou-se necessário proceder à sua caracterização detalhada. Esta etapa foi fundamental para compreender o contributo específico de cada publicação, clarificar as opções metodológicas adotadas e situar as abordagens dentro do panorama atual da inspeção automática de módulos PV. Nesta síntese, destaca-se que, apesar da diversidade de contextos experimentais, todos os trabalhos analisados recorreram a técnicas de visão computacional baseadas em aprendizagem profunda, com predominância clara de arquiteturas de deteção da família *YOLO* — um indicador da maturidade e consolidação deste paradigma na análise automática de imagens *RGB* em ambiente PV. Esta convergência não foi casual: resultou da combinação entre elevada velocidade de inferência, boa precisão mesmo em condições adversas e facilidade de adaptação a plataformas embarcadas ou de baixo consumo energético.

Embora partilhassem esta base comum, os seis artigos diferenciaram-se significativamente no tipo de anomalias abordadas, nos conjuntos de dados utilizados, nos procedimentos de pré-processamento e *data augmentation* e nas variantes arquiteturais aplicadas às redes de deteção. Alguns estudos [7] [9] [10] concentraram-se exclusivamente na deteção de poeira, sujidade superficial e partículas difusas, enquanto outros abrangeram um espectro mais amplo de anomalias, incluindo sombras, folhas, manchas heterogéneas, objetos estranhos e irregularidades estruturais. Houve também trabalhos [11] [12] que combinaram visão *RGB* com informação térmica, explorando imagens infravermelhas para identificar hotspots e defeitos latentes não detetáveis no espectro visível. Estas diferenças metodológicas refletiram igualmente a diversidade dos *datasets* utilizados: desde coleções próprias obtidas em parques solares reais até bases de dados sintéticas ou ampliadas através de técnicas de geração e fusão de imagens. A variabilidade das condições de aquisição — iluminação, ângulo, distância ao módulo — reforçou a necessidade de modelos robustos e *pipelines* de processamento cuidadosamente calibrados.

Nos pontos seguintes apresentou-se de forma estruturada a caracterização dos estudos incluídos, ressaltando para cada um deles o modelo utilizado, o tipo de anomalia analisada, o *dataset*, as técnicas de pré-processamento e pós-processamento e as métricas de desempenho reportadas. Esta sistematização constituiu a base para a análise crítica que se segue, permitindo identificar padrões transversais, limitações metodológicas recorrentes e oportunidades de evolução tecnológica relevantes para o desenvolvimento de sistemas de inspeção fotovoltaica mais eficazes, fiáveis e escaláveis. A articulação destes contributos como descrito na **Tabela 2** forneceu ainda um enquadramento empírico sólido para sustentar a escolha das arquiteturas exploradas posteriormente neste relatório,

garantindo que o desenvolvimento proposto se apoiou nas tendências mais consistentes e relevantes da literatura recente.

Tabela 2 - Resumo dos estudos incluídos na revisão do estado da arte

Ref.	Ano	Arquiteturas	Finalidade do Estudo	Técnicas Utilizadas (Pré / Aug / Pós)	Resultados
[7]	2023	YOLOv5 (CNN single-stage)	Detetar e classificar sujidade (poeira, folhas, excrementos) em painéis PV a partir de imagens RGB	Pré: normalização e <i>resize</i> Aug: <i>Mosaic</i> , rotações, ajuste <i>HSV</i> / espelhamentos Pós: NMS	<i>mAP@0.5</i> entre 94–96% precisão ~93% <i>recall</i> ~92% <i>throughput</i> ~45 FPS
[12]	2024	YOLOv8 + UAV/edge (pipeline embarcado)	Inspeção autónoma com quadrotor, desde aquisição até deteção on-board e registo de anomalias	Pré: estabilização/normatização de frames Aug: rotações/escala para voo; <i>motion blur</i> sintético (quando aplicável) Pós: NMS; <i>tracking</i> simples para sequência	taxa de deteção > 90% ~30 FPS em GPU embarcada latência baixa adequada para operação aérea
[9]	2025	YOLOv8 (backbone leve; head desacoplado, anchor-free)	Sistema inteligente para deteção de sujidade superficial otimizado para operação em campo/edge	Pré: normalização; <i>resize</i> 640×640 Aug: <i>Mosaic</i> , <i>Copy-Paste</i> , <i>Color Jitter</i> Pós: NMS com calibração de confiança/IOU	<i>mAP</i> superior ao YOLOv7 (~95%) precisão ~95% <i>recall</i> ~94% ~45 FPS robusto a iluminação variável
[13]	2024	CNN-based pipeline (classificação/deteção; conforme descrito)	Implementar sistema completo de deteção de defeitos em PV com <i>deep learning</i>	Pré: limpeza de ruído, balanceamento de classes Aug: rotações, recortes; variação de iluminação Pós: regras de consolidação por classe / limiares	precisão ≥ 95% falsos positivos reduzidos estabilidade elevada sob ruído e variação de iluminação
[14]	2025	DINOv2 / ViT + atenção multi-cabeça	Detetar poeira/manchas difusas explorando <i>transformers</i> para robustez a iluminação/textura	Pré: equalização adaptativa; normalização Aug: <i>Hue Shift</i> , ruído, <i>flip</i> Pós: <i>Soft-NMS</i> + fusão contextual	<i>mAP</i> ~95% precisão ~94% maior robustez a sombra/brilho latência superior às arquiteturas YOLO
[10]	2023	YOLOv8 (all versions)	Identificar “partículas desnecessárias” /sujidade fina na superfície de painéis, visando limpeza seletiva.	Pré: padronização de brilho/contraste Aug: <i>Flip</i> , <i>RandomCrop</i> , variações de cor Pós: NMS/filtragem por classe	precisão 93–96% FPS ~45 boa generalização a diferentes lotes e condições ambientais

Apesar de apresentarem metodologias distintas, os seis estudos incluídos convergem na adoção de técnicas baseadas em CNN e, em particular, em variantes da família YOLO, que se afirmam como a arquitetura dominante na deteção automática de sujidade e anomalias superficiais em módulos PV. Esta prevalência não decorre apenas da popularidade dos modelos, mas sobretudo da sua capacidade de manter precisão elevada em situações operacionais exigentes: reflexos intensos, heterogeneidade de cores, ângulos de captura não padronizados

e variações bruscas de luminosidade típicas de parques solares a céu aberto. Modelos mais recentes — como *YOLOv8*, *PP-YOLO* e *pipelines* híbridos que incorporam mecanismos de atenção multicanal — alcançaram um equilíbrio particularmente eficaz entre velocidade de inferência, sensibilidade a objetos pequenos e estabilidade em contextos ruidosos, revelando-se adequados tanto para sistemas embarcados como para inspeções aéreas com *UAVs*.

Paralelamente, abordagens baseadas em arquiteturas transformadoras, como *DINOv2* e *ViT*, evidenciaram vantagens específicas na detecção de padrões subtis ou difusos, como manchas ligeiras, sujidade de textura homogênea ou irregularidades de baixa saliência. A atenção global destes modelos favoreceu uma leitura mais contextualizada da imagem, permitindo detetar anomalias que escapam aos detetores convencionais. No entanto, esta sofisticação apresentou custos: maior latência, dependência mais acentuada de *hardware* robusto e maior complexidade de treino. Estas limitações tornaram os *transformers* menos competitivos em cenários de produção, onde a velocidade de processamento e a eficiência energética são fatores determinantes.

A literatura analisada reforçou ainda uma fragilidade transversal: a falta de *datasets* extensos, diversificados e representativos das condições reais de operação. A maioria dos estudos baseou-se em coleções reduzidas, captadas em contextos controlados ou com variação limitada, o que condiciona a capacidade de generalização dos modelos. As estratégias de *data augmentation* procuraram mitigar esta lacuna, mas não substituíram a necessidade de bases de dados mais amplas, anotadas com rigor e provenientes de diferentes geografias, inclinações de módulos, prestadores de O&M e condições meteorológicas. Acresce a ausência de protocolos normalizados de avaliação, que dificultou a comparação direta entre estudos e impediu a identificação clara de *benchmarks* no domínio da inspeção fotovoltaica.

A partir desta caracterização, emergiram tendências claras e oportunidades de investigação: a integração multimodal (*RGB*, infravermelhos e eventualmente profundidade), o desenvolvimento de *pipelines* mais leves e energeticamente eficientes, e a necessidade de estratégias de treino que melhorem a resiliência dos modelos a iluminação extrema, poeira heterogênea e artefactos de captura. Estas observações fundamentaram a análise crítica apresentada na secção seguinte e sustentaram a seleção das abordagens exploradas neste trabalho, garantindo que o desenvolvimento proposto se ancorou nas soluções mais promissoras da literatura recente.

De forma a sintetizar temporalmente esta caracterização, a **Figura 2** apresenta a distribuição dos seis estudos incluídos pela respetiva data de publicação. Observa-se uma concentração quase exclusiva de trabalhos a partir de 2023, com particular incidência em 2024 e 2025, o que confirma que a detecção automática de sujidade e anomalias superficiais em módulos PV é um tema emergente, associado às gerações mais recentes de modelos de DL. Esta concentração temporal foi

relevante por duas razões: em primeiro lugar, indicou que as soluções analisadas incorporam já arquiteturas modernas (como *YOLOv5*, *YOLOv8* ou *DINOv2*), o que aproximou a revisão do estado da arte atual; em segundo lugar, mostrou que ainda não existiam séries históricas longas nem *benchmarks* consolidados, o que limitou a possibilidade de meta-análises formais, mas abriu espaço para propostas inovadoras que contribuam para estabilizar práticas e métricas neste domínio. Assim, os resultados discutidos nas secções seguintes devem ser lidos como um retrato de uma área em rápida consolidação, mais do que como um campo tecnologicamente saturado.

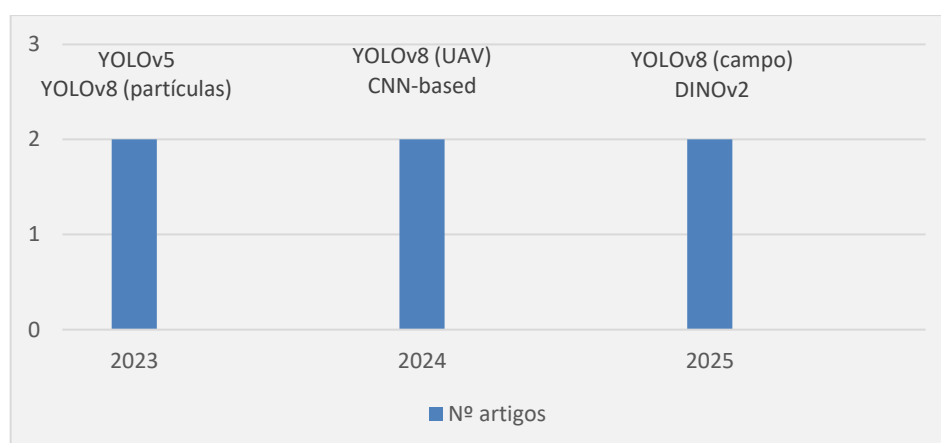


Figura 2 - Distribuição temporal dos seis estudos incluídos na revisão sistemática

Para além da distribuição temporal, importou sintetizar o desempenho dos modelos avaliados nos diferentes estudos, em linha com a questão de investigação relativa às arquiteturas mais eficazes. A **Figura 3** representa o desempenho médio em termos de *mAP* das principais famílias de modelos utilizadas — *YOLOv5*, *YOLOv8*, variantes *YOLO* genéricas e *DINOv2/ViT* — agregando os valores reportados nos seis artigos. Apesar de as métricas não serem diretamente comparáveis em sentido estrito, uma vez que cada estudo utiliza *datasets* e cenários experimentais distintos, o padrão global foi claro: o *YOLOv8* apresentou os valores médios de *mAP* mais elevados, seguido de perto pelo *YOLOv5* e pelo *DINOv2*, enquanto as variantes *YOLO* não especificadas surgiram ligeiramente abaixo. Esta ordenação indicou que a evolução dentro da própria família *YOLO* (das versões anteriores para o *YOLOv8*) se traduziu não apenas em ganhos marginais, mas em melhorias consistentes de desempenho em diferentes contextos de inspeção. Em paralelo, a proximidade do *DINOv2* em termos de *mAP* sugeriu que abordagens baseadas em *transformers* são competitivas em precisão, mas a sua adoção generalizada dependerá, como se discute adiante, do compromisso entre custo computacional e requisitos de operação em campo.

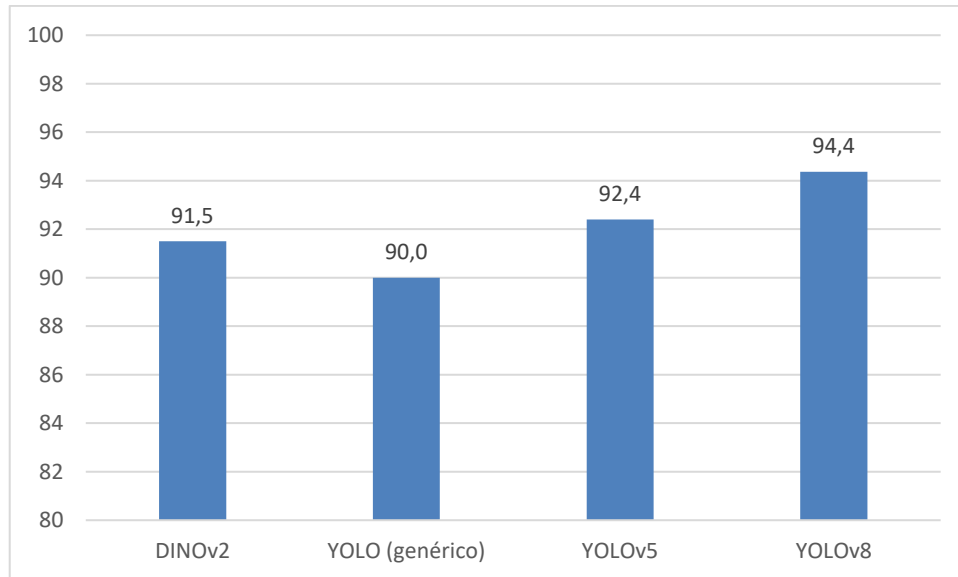


Figura 3 - Desempenho médio em termos de *mAP* dos modelos utilizados nos seis estudos incluídos na revisão sistemática

A **Figura 4** complementa esta análise ao representar explicitamente o compromisso entre precisão e velocidade de inferência para os modelos considerados, relacionando o mAP médio com a taxa de processamento em FPS. Cada ponto corresponde a um modelo em contexto experimental distinto, permitindo observar não apenas “quem acerta mais”, mas também “quem acerta mais depressa”, dimensão crítica em cenários de inspeção contínua com drones, câmaras fixas ou sistemas embarcados. Os pontos associados às variantes YOLOv8 situam-se, de forma consistente, na região superior direita do gráfico, isto é, na fronteira de eficiente do conjunto: combinam valores elevados de mAP com FPS superiores aos restantes modelos, o que significa que não existe outro modelo na amostra que os supere simultaneamente em precisão e velocidade. Em contraste, arquiteturas como o DINOv2 tendem a apresentar precisão competitiva, mas com menores taxas de processamento, evidenciando um custo computacional mais elevado e uma adequação limitada a dispositivos com recursos restritos. Esta visualização reforçou que a escolha de arquitetura, em contexto PV, não pode ser feita com base apenas na métrica de mAP isolada; é necessário considerar o enquadramento operacional, nomeadamente a necessidade de processar grandes volumes de imagens em tempo quase real e com consumo energético reduzido.

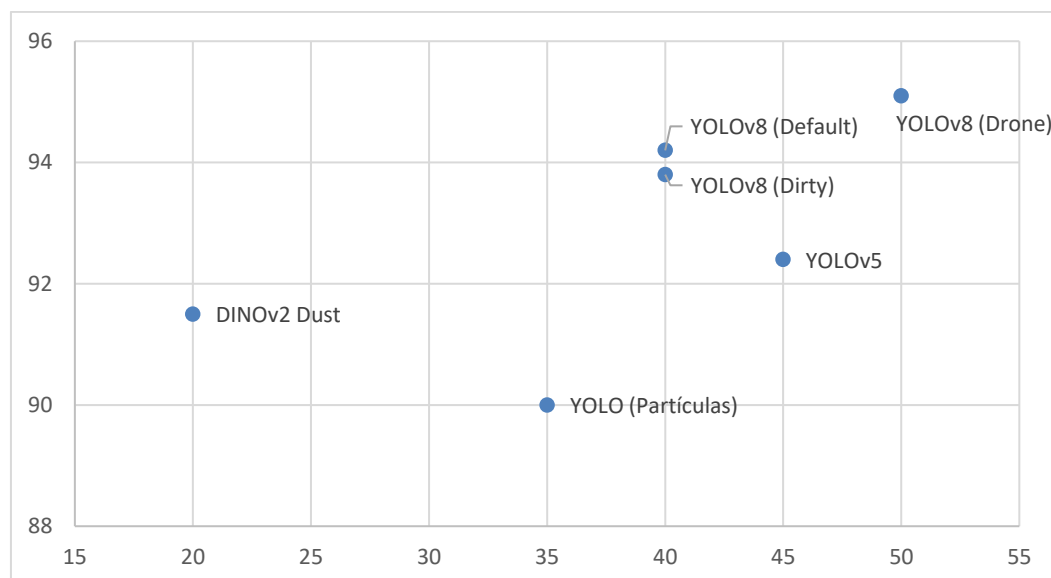


Figura 4 - Relação entre precisão (mAP) e velocidade de inferência (FPS) dos modelos utilizados nos seis estudos incluídos na revisão sistemática

2.4 Técnicas de Melhoria nas Abordagens

O desempenho dos modelos de deteção e classificação depende fortemente da qualidade e da diversidade dos dados. As etapas de pré-processamento, *data augmentation* e pós-processamento são, por isso, determinantes para a robustez do sistema e aparecem de forma sistemática nos estudos analisados. Em termos gerais, o pré-processamento procura estabilizar a distribuição dos dados de entrada; a *augmentation* aumenta artificialmente a variabilidade do conjunto de treino; e o pós-processamento refina as predições do detetor, reduzindo falsos positivos e resolvendo sobreposições de caixas.

No pré-processamento, operações como normalização de brilho/contraste, correção de distorções óticas, redimensionamento e remoção de ruído ajudam a mitigar variações de iluminação, reflexos e sombras típicas de imagens recolhidas por *drones* ou câmaras fixas em ambiente exterior. Em [7] é aplicado um esquema simples, mas eficaz de normalização e *resize* das imagens para a resolução de treino, assegurando consistência na entrada do modelo. Já em [12] o pré-processamento inclui estabilização e normalização de *frames* captadas em voo, reduzindo o impacto de vibrações e variações bruscas de exposição. Em abordagens mais clássicas, como [13], o pré-processamento integra também limpeza de ruído e balanceamento de classes, de forma a mitigar desequilíbrios entre exemplos de defeitos e de normalidade. Em cenários PV, este tratamento prévio é essencial para reduzir o erro de generalização e facilitar o treino estável das redes, tal como ilustrado pela utilização sistemática de *preprocessing layers* em *frameworks* como *TensorFlow/Keras* [15].

A *data augmentation* amplia artificialmente o conjunto de treino para melhorar a generalização às condições reais. Para inspeção de painéis PV, é habitual aplicar

rotações, espelhamentos (*flip*), recortes aleatórios, *jitter* de cor e adição controlada de ruído, simulando alterações de perspectiva, poeiras difusas, sombras parciais e variações de luz. Em modelos da família YOLO, práticas consolidadas incluem *Mosaic* (composição de quatro imagens), *MixUp* ou *Copy-Paste* (combinação de objetos/instâncias) e ajustes *Hue*, *Saturation*, *Value* (*HSV*). Isto foi visível em [7], onde o *Mosaic* e o ajuste *HSV* são usados para aumentar a diversidade de padrões de sujidade sem degradar a semântica dos objetos, e em [9], que combina *Mosaic*, *Copy-Paste* e variações de cor (*Color Jitter*) para robustez a diferentes condições de iluminação e fundo. Em [12] surgem ainda rotações e escalas orientadas ao voo, bem como *motion blur* sintético para simular o arrastamento associado ao movimento do UAV. No caso de sujidade difusa e pouco contrastada, como em [14], combinam-se *jitter* de cor, *hue shift* e ruído com operações de *flip*, reforçando a sensibilidade do modelo a variações subtis de textura; de forma semelhante, o artigo [10] recorre a *RandomCrop*, *flip* e variações de cor para melhorar a deteção de partículas pequenas em diferentes contextos de brilho e sujidade.

A **Figura 5** ilustra alguns exemplos de *data augmentation* tradicional aplicados a imagens de defeitos em células solares – rotações, espelhamento horizontal, injeção de ruído do tipo *salt-and-pepper*, *random erasing* e *cutout* – mostrando como, a partir de uma única imagem original, se gera um conjunto mais variado de exemplos para treino do modelo.

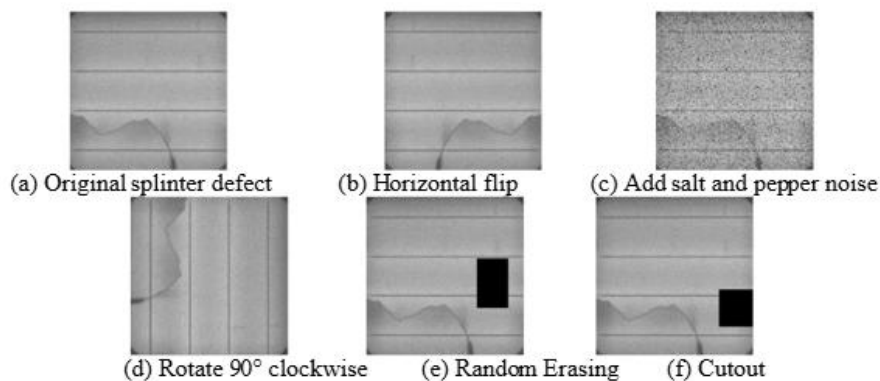


Figura 5 - Exemplos de técnicas clássicas de *data augmentation* (*flip*, rotação, ruído, *cutout*) aplicadas a uma imagem de defeito. Adaptado de [16]

Por fim, o pós-processamento refina as predições do detetor para reduzir falsos positivos e consolidar resultados. A técnica base é a NMS, que filtra caixas redundantes mantendo as mais prováveis; é precisamente a abordagem seguida em [7], [12] e [10], onde a NMS é ajustada com limiares de confiança e *IoU* adaptados ao tipo de sujidade e ao compromisso desejado entre *recall* e precisão. Em contextos mais exigentes, como em [14], recorre-se a variantes como *Soft-NMS* e fusão contextual de caixas, atenuando as pontuações de caixas sobrepostas em vez de as descartar de imediato, o que melhora o *mAP* em cenários com objetos

próximos ou manchas contíguas. Noutros trabalhos, como [13], são ainda aplicadas regras de consolidação por classe ou limiares específicos por tipo de defeito, reduzindo a incidência de falsos positivos em classes raras.

Estas técnicas de pré-processamento, *augmentation* e pós-processamento, sintetizadas na **Tabela 2** comparativa dos seis estudos seleccionados, mostraram que o desempenho reportado não dependeu apenas da escolha do modelo (YOLOv5, YOLOv8, DINOv2, SSD melhorado, etc.), mas de todo o *pipeline* de processamento de imagem. Esta constatação foi particularmente relevante para o presente trabalho: a definição do *pipeline* de dados — desde a normalização e *resize* das imagens até à configuração de *Mosaic*, *Color Jitter* e *NMS/Soft-NMS* — foi um fator determinante para reproduzir, e idealmente aproximar, os níveis de *mAP* e robustez observados no estado da arte.

Para além das operações geométricas e fotométricas clássicas, alguns trabalhos recentes exploram estratégias de geração sintética de sujidade específicas para o domínio PV. No artigo [17] é proposta a criação de subconjuntos de dados “*Naïve*” e “*Realistic*”, em que segmentos de sujidade recortados a partir de imagens reais são colados sobre diferentes fundos de painéis, recorrendo a máscaras binárias e a esquemas de *Copy-Paste* condicionados (**Figura 6**). Este processo permite controlar a distribuição espacial e a densidade das partículas de pó, produzindo milhares de exemplos a partir de um número reduzido de imagens anotadas.

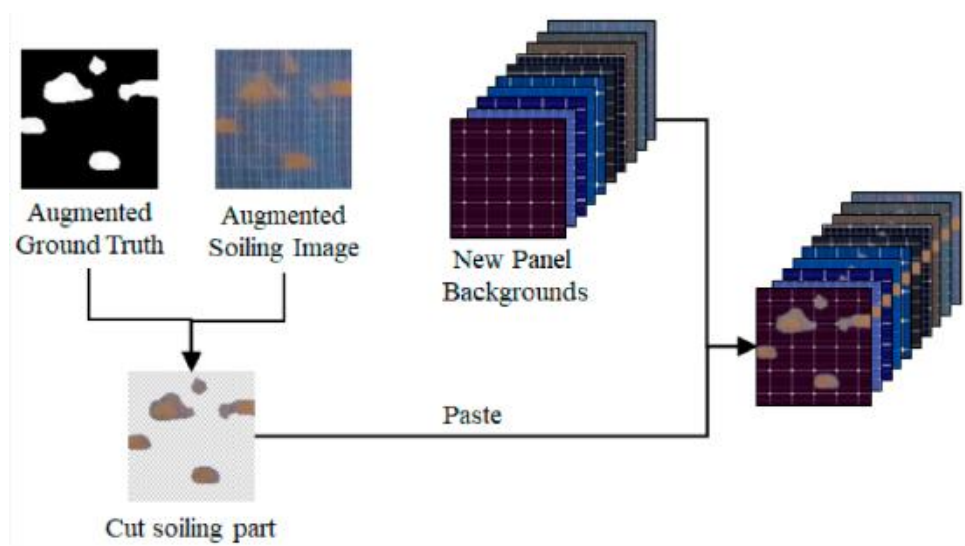


Figura 6 - Processo de construção do *dataset* “*Naïve/Realistic*” com recorte de regiões de sujidade e colagem sobre novos fundos de painéis. Adaptado de [17]

Numa segunda fase, os autores introduzem ainda máscaras de transparência derivadas da diferença *RGB* entre imagens limpas e sujas do mesmo painel, de forma a modelar melhor variações subtis de intensidade associadas a sujidade

translúcida (**Figura 7**). A partir de pares de imagens alinhadas, é calculado um mapa de diferença que realça apenas as regiões efetivamente alteradas pela deposição de poeiras, resíduos ou filmes finos quase imperceptíveis a olho nu. Esse mapa é posteriormente normalizado e transformado numa máscara de opacidade, que funciona como um canal adicional de informação durante a geração de novas amostras, modulando a contribuição da sujeidade em cada pixel.

Esta estratégia permitiu distinguir, de forma mais fina, entre padrões estruturais do próprio módulo (linhas de interconexão, texturas do vidro, reflexos especulares) e variações de luminância introduzidas pela sujeidade, reduzindo a probabilidade de o modelo confundir padrões do painel com defeitos superficiais. Em termos práticos, as máscaras de transparência funcionam como um “guia” para o gerador de dados sintéticos, assegurando que a sujeidade adicionada às imagens é colocada em regiões plausíveis e com intensidade realista. Tal como discutido pelos autores, este refinamento contribui para uma melhor generalização do modelo em cenários reais, onde a sujeidade raramente aparece como manchas opacas bem delimitadas, mas antes como gradientes suaves e heterogêneos de deposição de partículas.

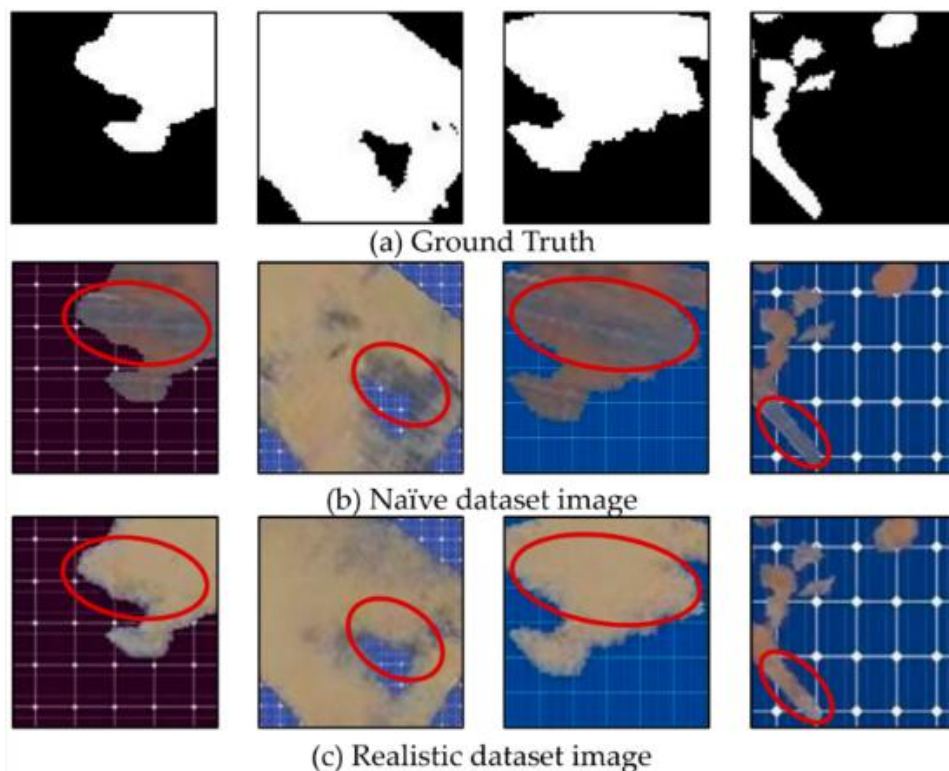


Figura 7 - Exemplos de imagens de sujeidade, máscaras de *ground truth* e máscaras de transparência associadas. Adaptado de [17]

Em síntese, o estado da arte mostrou que a qualidade do *pipeline* de dados é tão determinante como a escolha da arquitetura de detecção. Operações de pré-processamento bem calibradas reduzem o ruído e estabilizam a distribuição das

imagens; esquemas de *data augmentation* clássicos (rotações, *flips*, *jitter* de cor, *cutout*) aumentam a variabilidade do treino sem introduzir artefactos visuais graves; e abordagens mais avançadas, como a geração sintética de sujidade com máscaras de transparência, permitem expandir significativamente os *datasets* a partir de um número limitado de imagens anotadas, desde que cuidadosamente desenhadas para preservar o realismo. Em paralelo, técnicas de pós-processamento como a NMS e as suas variantes refinam as predições e controlam o compromisso entre *recall* e precisão.

Para o presente trabalho, estas conclusões tiveram implicações diretas: a definição do *pipeline* de pré-processamento e *augmentation* — incluindo a eventual utilização de estratégias de síntese de sujidade inspiradas nos estudos analisados — foi feita de forma criteriosa e alinhada com as limitações de dados disponíveis e com os requisitos de tempo de processamento em contexto real. Esta perspetiva é retomada nas secções 2.5 e 2.6, articulando a análise dos conjuntos de dados com a comparação crítica das arquiteturas e a escolha do modelo base a adotar no sistema proposto.

2.5 Conjuntos de Dados Utilizados nos Estudos Analisados

A qualidade e a representatividade dos conjuntos de dados são determinantes para o desempenho dos modelos de visão computacional. Nos seis estudos selecionados verifica-se uma clara predominância de imagens *RGB* da superfície dos módulos PV, recolhidas em centrais reais ou em ambiente semi-controlado, anotadas sobretudo com caixas delimitadoras *BBoxes* e, nalguns casos, com rótulos adicionais de tipo ou grau de sujidade. Em todos os casos, os *datasets* foram construídos especificamente para o estudo, o que gerou resultados relevantes, mas dificultou a comparação direta entre trabalhos e a sua replicação noutras instalações.

No artigo [7] os autores construíram um conjunto de imagens *RGB* captadas em centrais fotovoltaicas reais com diferentes combinações de poeira, folhas e excrementos de aves sobre os módulos. Cada instância de sujidade foi anotada com *BBoxes* e a respetiva classe, permitindo treinar o YOLOv5 como detetor multi-classe. As imagens apresentaram variação moderada de iluminação e perspetiva (recolhas em diferentes horários e ângulos de câmara), mas com distâncias relativamente curtas ao painel, o que facilitou a leitura detalhada da textura do vidro e limitou o impacto de *motion blur*.

Em [12] o cenário foi mais desafiante: o *dataset* resultou de capturas vídeo realizadas por um UAV em voo, com ângulos oblíquos, variações significativas de altitude e *backgrounds* complexos. As imagens extraídas incluíram painéis, estruturas de suporte, solo e vegetação, e a sujidade competiu visualmente com sombras e reflexos fortes. As anotações foram novamente feitas por *BBoxes* sobre regiões de sujidade ou defeito, mas a distribuição espacial dos objetos e a grande

variação de escala tornaram o problema mais exigente do ponto de vista da deteção e justificaram o uso de *augmentation* agressiva (rotações, *motion blur*, variações de escala).

Em [9] os autores seguiram uma abordagem intermédia: combinou imagens obtidas em campo com capturas em ambiente semi-controlado, procurando cobrir diferentes níveis de sujidade superficial. O *dataset* foi estruturado por classes que distinguiam, por exemplo, sujidade leve, moderada e severa, bem como tipos específicos de partículas (folhas, poeira acumulada, excrementos). Este desenho permitiu não só detetar a presença de sujidade, mas também inferir um grau aproximado de severidade, o que foi particularmente útil para priorizar intervenções de limpeza. Verificou-se, no entanto, um desequilíbrio natural entre imagens “limpas” e imagens com sujidade pronunciada, que foi compensado com esquemas de data *augmentation* descritos na secção 3.4.

Em [13] o objetivo foi mais abrangente: o conjunto de dados incluiu imagens de módulos com defeitos elétricos e estruturais (fissuras, quebras de célula, manchas de degradação) para além de situações de sujidade. A recolha combinou painéis reais em serviço com condições de ensaio em laboratório, recorrendo a iluminação controlada e, em alguns casos, a informação complementar (por exemplo, imagens térmicas). As imagens foram, em grande parte, rotuladas para classificação de defeitos (módulo normal, célula fissurada, *hotspot*, etc.), tendo depois sido usadas numa *pipeline* em que uma primeira etapa identificou a região relevante e uma segunda classificou o tipo de anomalia. Este desenho aproximou-se de um cenário de diagnóstico global do módulo, indo além da mera inspeção de *soiling*.

O artigo [14] explorou um tipo de conjunto de dados mais subtil: o foco esteve em manchas difusas de poeira e padrões de textura de baixo contraste, muitas vezes difíceis de distinguir a olho nu. As imagens foram captadas em condições de iluminação variáveis e incluíram tanto painéis uniformemente cobertos de poeira fina como casos com acumulações localizadas. As anotações combinaram rótulos globais de nível de sujidade do painel com regiões de interesse associadas às áreas mais críticas, dependendo da configuração de treino. Este tipo de *dataset* foi particularmente adequado a modelos com mecanismos de atenção, capazes de explorar relações de contexto e nuances texturais em larga escala.

Em [10] o *dataset* incidiu em partículas pequenas e localizadas — folhas, resíduos sólidos, detritos diversos — frequentemente dispersas sobre a superfície do módulo. As imagens foram captadas tanto em plantas reais como em cenários controlados, colocando objetos específicos sobre os painéis para simular diferentes condições de contaminação. As anotações privilegiaram *BBoxes* de pequena dimensão e classes que distinguem tipos de partículas “indesejadas”, avaliando a capacidade do modelo em detetar objetos de pequena escala em contextos de fundo relativamente uniforme.

Apesar das diferenças de foco, estes estudos partilharam características estruturais semelhantes:

- Dimensão moderada dos conjuntos de dados (tipicamente algumas centenas a poucos milhares de imagens ou recortes anotados);
- Forte dependência de dados proprietários, recolhidos pelos próprios autores e, em muitos casos, não disponibilizados publicamente;
- Heterogeneidade na forma de anotação (detecção com *BBoxes*, classificação global, recortes *patch-based*), que tornou difícil comparar de forma direta os valores de *mAP*, precisão e *recall* reportados.

Este cenário contrastou com abordagens, como a apresentada em [17] (ver **Figura 6** e **Figura 7**), onde o conjunto de dados foi parcialmente gerado de forma sintética: máscaras de sujidade extraídas de imagens reais foram coladas sobre fundos de painéis limpos, produzindo versões “*Naïve*” e “*Realistic*” das mesmas cenas. Embora este trabalho não tenha feito parte do núcleo de seis artigos selecionados, ilustrou uma tendência emergente para recorrer a dados gerados ou fortemente aumentados como forma de contornar a escassez de amostras anotadas, à custa de um esforço adicional para garantir realismo visual e evitar artefactos.

No contexto do presente projeto, seguiu-se uma abordagem distinta da maioria dos trabalhos revistos, privilegiando a utilização de conjuntos de dados públicos já disponíveis na comunidade como base para construção de um *dataset* alinhado com o caso de uso. Em particular, foram selecionados *datasets* disponibilizados na plataforma *Roboflow* [18], com imagens *RGB* de módulos PV em diferentes condições visuais, e procedeu-se à sua integração num único conjunto coerente. Esta opção permitiu tirar partido de dados já anotados e validados pela comunidade, mas evitando a dependência de um único *dataset* “pronto” cuja taxonomia e critérios de anotação não coincidissem com os objetivos do sistema. O *dataset* final foi estabilizado numa taxonomia operacional de três classes (*clean*, *dusty* e *bird-drop*), adequada à lógica de inspeção em O&M e suficientemente consistente para treino supervisionado.

A opção por recorrer a *datasets* públicos teve várias vantagens: aumentou a reprodutibilidade do trabalho, facilitou a comparação com estudos futuros e reduziu o esforço de recolha e anotação manual de imagens. No entanto, levantou também desafios importantes, nomeadamente a heterogeneidade entre fontes (condições de captura, enquadramento, iluminação e estilo de anotação) e o potencial desfasamento entre o domínio dos dados e o contexto específico onde se pretende aplicar o sistema. Por essa razão, a utilização destes *datasets* públicos foi acompanhada por um processo de harmonização e curadoria e, sempre que necessário, por enriquecimento com imagens adicionais provenientes de outras fontes, anotadas de acordo com o mesmo esquema de classes — com especial atenção à classe *bird-drop*, inicialmente menos representada — de forma a reduzir desequilíbrios e mitigar o desfasamento entre domínios.

Em síntese, a análise dos conjuntos de dados descritos na literatura, em articulação com a identificação de recursos públicos disponíveis, conduziu a uma orientação clara para este projeto. Em vez de construir de raiz um *dataset* totalmente proprietário — processo moroso, caro e difícil de reproduzir — optou-se por assentar o desenvolvimento e a avaliação do modelo numa base aberta, construída a partir de *datasets* do *Roboflow* e refinada através de fusão, anotação e validação na própria plataforma. Esta decisão é complementada por um segundo eixo igualmente importante: o *pipeline* de pré-processamento e *data augmentation* (secção 3.4) foi desenhado para compensar limitações típicas destes conjuntos (dimensão, desequilíbrio de classes e heterogeneidade visual), explorando normalização e aumentos geométricos/fotométricos de forma controlada. O objetivo final foi que o protótipo resultante não fosse apenas capaz de obter boas métricas no *dataset* de treino, mas que conseguisse generalizar com robustez para condições próximas das que se verificam em centrais fotovoltaicas reais, onde iluminação, perspectiva e padrões de sujidade são inevitavelmente mais variáveis.

2.6 Comparação Crítica e Escolha do Modelo

A análise dos seis estudos seleccionados evidenciou a consolidação do DL como abordagem dominante na inspeção automática de centrais fotovoltaicas. Entre as arquiteturas avaliadas — YOLOv5, YOLOv8, DINOv2, CNN clássicas e variações baseadas em atenção — observou-se uma evolução clara no equilíbrio entre precisão, velocidade e adaptabilidade. Como sintetizado na **Figura 3**, as variantes YOLOv8 apresentaram, em média, os valores de *mAP* mais elevados entre os modelos analisados, com ganhos consistentes face ao YOLOv5 e às abordagens baseadas em CNN ou DINOv2.

O YOLOv8, presente em dois dos estudos analisados [12], [9], surgiu como a arquitetura mais completa, integrando avanços significativos em relação às versões anteriores, como o *head* desacoplado, *anchor-free detection* e melhor agregação multi-escala. Estas melhorias refletiram-se em ganhos médios de 2–3 % em *mAP* e reduções de latência até 30 %, tornando-o particularmente adequado para aplicações em tempo real ou sistemas embarcados, como *drones* de inspeção [12].

O modelo DINOv2 [14] destacou-se pela robustez a variações de iluminação e textura, graças ao uso de *transformers* e mecanismos de atenção multicanal. Apesar de alcançar elevada precisão ($mAP \approx 94\%$), apresentou maior complexidade computacional e menor velocidade de inferência, fatores que limitaram a sua adoção em sistemas de campo. Por outro lado, a abordagem CNN proposta em [13] — centrada na deteção de falhas elétricas e estruturais — demonstrou a aplicabilidade do DL a sistemas completos de diagnóstico, mas sem a capacidade de processamento em tempo real das soluções baseadas em YOLO. Finalmente, o estudo [10] evidenciou o potencial do YOLO para deteção de partículas e sujidade fina, mostrando boa generalização e reforçando a relevância desta família de modelos para inspeção detalhada de módulos PV.

De forma global, os resultados permitiram concluir que as arquiteturas YOLOv8 combinaram elevada precisão, baixa latência e ampla documentação técnica, tendo sido sustentadas por uma comunidade ativa de desenvolvimento (*Ultralytics*). A escolha deste modelo para o presente trabalho fundamentou-se, assim, na sua versatilidade, facilidade de treino e implementação, e comprovada eficácia em contextos reais de inspeção fotovoltaica, quer em plataformas fixas, quer embarcadas em UAVs.

A **Figura 8** sintetiza graficamente esta tendência, apresentando a distribuição das arquiteturas de DL utilizadas nos seis estudos analisados. Verifica-se que cinco dos seis trabalhos recorrem a variantes da família YOLO — incluindo YOLOv5, YOLOv8 e versões não explicitamente identificadas — enquanto apenas um explora uma abordagem baseada em *transformers* (*DINOv2/ViT*). Esta assimetria não deve ser lida apenas como efeito de “moda” tecnológica: resultou da combinação entre maturidade das implementações, disponibilidade de ferramentas e documentação (por exemplo, a *framework Ultralytics*) e, sobretudo, da capacidade destas arquiteturas conciliarem *mAP* elevado com latências compatíveis com operação em campo, tal como evidenciado na **Figura 3** e **Figura 4**. Em contrapartida, o *DINOv2* surgiu como linha exploratória promissora, com ganhos específicos em cenários de textura difusa e baixa saliência, mas ainda distante de uma adoção generalizada em ambientes industriais de inspeção fotovoltaica. Assim, a opção por uma arquitetura YOLOv8 neste trabalho não resultou apenas da leitura dos resultados quantitativos isolados, mas também da análise da trajetória de adoção das diferentes famílias de modelos na literatura recente.

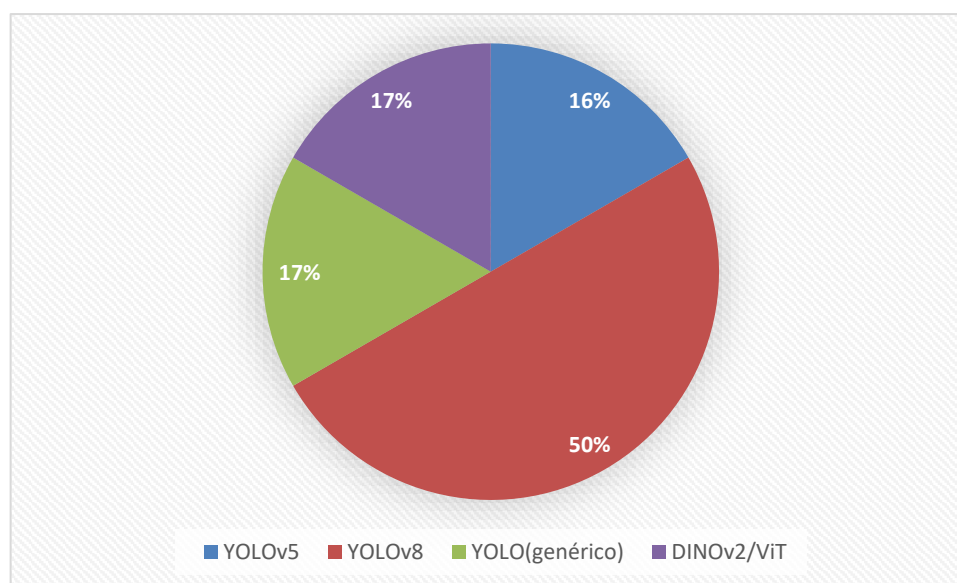


Figura 8 - Distribuição das arquiteturas de *deep learning* utilizadas nos estudos analisados

O panorama apresentado na **Figura 8** reforçou esta leitura quantitativa: cerca de metade dos estudos analisados recorreram explicitamente ao YOLOv8, enquanto o restante dividiu-se entre variantes anteriores da família YOLO (como o YOLOv5 ou designações genéricas “YOLO-based”) e abordagens alternativas baseadas em *transformers*, como o DINOv2/ViT. Esta distribuição não é neutra: traduziu uma tendência clara da comunidade para migrar de modelos mais simples, como CNNs clássicas, para arquiteturas *one-stage* modernas, otimizadas para execução em tempo real e com *pipelines* de treino já bastante estabilizados.

Do ponto de vista da escolha tecnológica para este trabalho, importou sublinhar que a decisão não assentou apenas no valor absoluto de *mAP* ou na popularidade momentânea de um modelo, mas no compromisso entre três dimensões: desempenho, custo computacional e maturidade do ecossistema. Os resultados dos seis estudos mostraram que o YOLOv8 atingiu níveis de precisão comparáveis — e, em vários casos, superiores — ao DINOv2, ao mesmo tempo que manteve tempos de inferência significativamente inferiores, sobretudo nas variantes “n” e “s” mais leves. Em contraste, o DINOv2 apresentou excelente robustez a variações de iluminação e textura, mas à custa de uma arquitetura mais pesada e de requisitos de *hardware* pouco compatíveis com sistemas de inspeção em campo, especialmente quando se pensa em *drones* ou dispositivos *edge* com recursos limitados.

Outro fator decisivo foi a maturidade da ferramenta. O YOLOv8 esteve integrado no ecossistema *Ultralytics* [19], com documentação extensa, scripts de treino, validação, exportação e *tracking* já consolidados, bem como suporte nativo para tarefas de deteção, segmentação e classificação numa *Application Programming Interface* (API) unificada [20]. Esta maturidade reduziu o esforço de engenharia necessário para construir o protótipo, facilitou a reprodutibilidade dos resultados e diminuiu o risco de bloqueios técnicos durante o desenvolvimento.

Apesar disso, é importante reconhecer que, à data de redação deste relatório, existem já iterações mais recentes da família YOLO. Em particular, o YOLOv11, apresentado pela *Ultralytics* em 2024, que introduz um *backbone* e *neck* melhorados, otimizações no *pipeline* de treino e uma gestão mais eficiente de parâmetros, alcançando, em benchmarks de referência como o COCO (*Common Objects in Context*), *mAP* superior ao YOLOv8m com cerca de 22 % menos parâmetros [21]. Paralelamente, modelos como o YOLOv9 exploram conceitos arquiteturais inovadores (PGI, GELAN) orientados sobretudo para cenários de investigação e maximização de desempenho em *GPUs* de alta gama [22]. Há ainda uma linha de desenvolvimento em curso para versões futuras, como o YOLO26, apontada para finais de 2025, com o objetivo declarado de ser mais pequena, mais rápida e *end-to-end*, mantendo o suporte às tarefas já disponíveis no YOLOv11 [23].

À luz deste contexto, a opção tecnológica adotada neste trabalho assentou numa estratégia deliberadamente pragmática, mas não conformista. O YOLOv8 foi

assumido como referência operacional porque combinou maturidade técnica com previsibilidade de engenharia: foi um detetor amplamente validado em múltiplos domínios — incluindo inspeção visual em contexto PV — e esteve ancorado num ecossistema estável de treino, validação, exportação e colocação em produção, com *tooling* consistente e documentação consolidada. Isto não foi um detalhe “de conforto”: num protótipo que tem de evoluir rapidamente, a estabilidade do *pipeline* reduziu o risco de bloqueios técnicos e deslocou o esforço para onde interessa — construir o *dataset*, treinar, medir, errar cedo e corrigir — em vez de desperdiçar semanas a contornar incompatibilidades, scripts frágeis ou infraestruturas pouco reproduzíveis [20]. Em paralelo, foi explorado o YOLOv11, enquanto geração mais recente desta linha, beneficiando de melhorias no *backbone* e no *neck*, de um *pipeline* mais eficiente e do suporte alargado a tarefas além da deteção, mantendo compatibilidade com a mesma API e com a mesma lógica de dados [21]. Esta abordagem em dois níveis evita tanto o conservadorismo cego como o risco de apostar tudo numa novidade: o objetivo é comparar, de forma limpa e justa, o compromisso entre precisão e custo computacional num cenário realista de inspeção de sujidade e anomalias em módulos PV.

Do ponto de vista evolutivo, importou reconhecer que esta escolha não fecha o sistema — prepara-o. Em particular, para situações em que a geometria da sujidade (bordos difusos, manchas irregulares, acumulações com contorno pouco definido) seja mais informativa do que uma caixa delimitadora, foi considerada a variante YOLOv11-seg como opção nativa de segmentação, permitindo avaliar se máscaras oferecem ganho real face à deteção clássica, e como esse ganho se posiciona perante abordagens baseadas em *transformers*, como as exploradas com DINOv2 em cenários de poeira subtil e textura de baixo contraste [14]. Esta avaliação, contudo, dependia criticamente da qualidade e adequação dos dados: por isso, o *pipeline* foi pensado desde já para treinar e comparar modelos sobre um *dataset* próprio, anotado na plataforma *Roboflow*, construído a partir da fusão de dois *datasets* públicos aí disponibilizados e depois enriquecido com novas imagens e novas anotações ajustadas ao caso de uso — um processo que foi detalhado no Capítulo 4, juntamente com os testes comparativos entre YOLOv8, YOLOv11/YOLOv11-seg e DINOv2. Finalmente, como linha de evolução futura, manteve-se em aberto a integração de arquiteturas de próxima geração como o YOLOv26, apontado como um passo no sentido de inferência *end-to-end* e desenho orientado a eficiência, particularmente relevante para cenários *edge* [24]. Esta progressão foi coerente com a filosofia do trabalho: começar com um núcleo robusto, validar com método e só depois escalar ambição com evidência.

Em síntese, o DL tem demonstrado elevado potencial na inspeção de módulos PV. Em particular, a literatura aponta para modelos de deteção de objetos baseados em CNN — com destaque para a família YOLO — como soluções recorrentes em inspeção visual. Entre estes, YOLOv5 e YOLOv8 surgem frequentemente como escolhas eficazes na deteção de sujidade e objetos como

poeira, folhas e excrementos, o que sustenta a sua consideração como base para o sistema desenvolvido e a avaliação comparativa conduzida no capítulo seguinte.

3. Enquadramento Teórico

A inspeção automatizada de centrais fotovoltaicas assenta na convergência de várias áreas científicas, entre as quais se destacam a visão computacional e a aprendizagem profunda (DL). A compreensão dos conceitos fundamentais associados a estas áreas é indispensável para enquadrar as decisões técnicas tomadas ao longo deste trabalho, desde a definição do problema até à escolha dos modelos e do respetivo *pipeline* de processamento de dados.

Neste capítulo apresentam-se os princípios teóricos que sustentam a abordagem proposta. Começa-se por enquadrar o papel da visão computacional na inspeção de sistemas PV, discutindo a evolução desde métodos clássicos de processamento de imagem até soluções baseadas em redes neurais convolucionais. Em seguida, aprofunda-se o estudo dos modelos de deteção de objetos mais relevantes para este contexto, com especial enfoque nas abordagens two-stage e one-stage que dominam o estado da arte atual. Este enquadramento teórico estabelece a base conceptual necessária para interpretar a revisão sistemática da literatura apresentada no Capítulo 2 e para enquadrar, nos capítulos seguintes, a arquitetura e a implementação do sistema de inspeção desenvolvido.

3.1 Visão Computacional para Inspeção de Centrais Fotovoltaicas

A visão computacional tem-se afirmado como uma ferramenta essencial na inspeção e monitorização de sistemas PV, permitindo a análise automática de imagens para detetar anomalias, sujidade e falhas estruturais que comprometem a eficiência dos módulos [11]. Em centrais de grande dimensão, compostas por dezenas ou centenas de milhares de painéis, a inspeção manual torna-se morosa, subjetiva e economicamente pouco viável. Neste contexto, a possibilidade de adquirir imagens de forma sistemática, através de câmaras instaladas em *drones*, robots terrestres ou estruturas fixas, e de as processar automaticamente com algoritmos de visão computacional, representa um salto qualitativo na forma como a O&M é conduzida.

A aquisição de dados visuais pode ser feita em diferentes modalidades: imagens *RGB* em luz visível, imagens térmicas obtidas por câmaras infravermelhas, ou combinações de ambas. As imagens *RGB* permitem identificar fenómenos como acumulação de sujidade, sombras parciais, presença de folhas ou excrementos de aves, enquanto as imagens térmicas evidenciam padrões de aquecimento anómalos associados a hot spots, falhas de bypass ou defeitos internos nas células. Em muitos sistemas, estas fontes de informação são complementares e podem ser integradas num *pipeline* comum de análise, aumentando a capacidade de deteção e diagnóstico.

As abordagens iniciais neste domínio baseavam-se sobretudo em técnicas clássicas de processamento de imagem — como segmentação, limiarização adaptativa, filtragem morfológica e detecção de contornos — usadas para isolar regiões de interesse com base em características de baixo nível, como intensidade, cor ou textura. Por exemplo, era comum aplicar filtros de realce de bordas para detetar fissuras, ou limiares de intensidade para separar zonas potencialmente sujas do fundo do painel. Embora estas técnicas pudessem produzir bons resultados em ambientes controlados, apresentavam limitações significativas em termos de generalização: pequenas variações na iluminação, no ângulo de captação, no tipo de módulo ou na presença de reflexos solares podiam degradar fortemente o desempenho. Na prática, a aplicação a larga escala exigia reajustes manuais constantes dos parâmetros de segmentação e limiarização, tornando o processo pouco robusto e difícil de automatizar em centrais com condições ambientais dinâmicas.

Com o avanço das técnicas de aprendizagem profunda DL, os sistemas de inspeção evoluíram para modelos baseados em CNN, capazes de extrair automaticamente características de alto nível a partir das imagens [25]. Em vez de dependerem de descritores manuais cuidadosamente desenhados, estes modelos aprendem, a partir de dados anotados, padrões visuais associados a diferentes tipos de defeitos. Inicialmente, muitos trabalhos concentraram-se em modelos de classificação de imagens, onde o objetivo era determinar se um painel apresentava ou não uma determinada anomalia. Apesar de representarem um avanço em relação aos métodos clássicos, estes classificadores puros não forneciam informação sobre a localização exata da anomalia, limitando a sua utilidade operacional.

Para ultrapassar essa limitação, começaram a surgir sistemas baseados em CNN capazes de realizar não apenas classificação, mas também detecção e localização de defeitos em imagens de sistemas PV. Alguns destes sistemas implementam *pipelines* completos de inspeção, desde a aquisição de imagens até à fase de decisão, integrando módulos de pré-processamento, redes convolucionais para extração de características e camadas especializadas para detecção de padrões anómalos [12]. Estes trabalhos demonstram ganhos claros face às abordagens convencionais, nomeadamente uma maior robustez face a variações de iluminação e enquadramento, bem como uma redução significativa do esforço manual de inspeção. Passou a ser possível detetar com fiabilidade sujidade, fissuras, delaminações e hot spots com níveis de precisão que se aproximam, ou em alguns casos superam, a inspeção visual por especialistas.

Em paralelo, a área da visão computacional foi fortemente impulsionada pelo desenvolvimento de modelos de detecção de objetos genéricos, capazes de localizar e classificar múltiplos elementos dentro de uma mesma imagem em tempo quase real. Modelos como as diferentes versões da família YOLO e variações otimizadas de SSD, originalmente propostos para detecção de objetos em cenários genéricos, começaram a ser adaptados ao contexto PV para identificar, por exemplo, regiões

com sujidade, objetos estranhos sobre os módulos ou áreas sujeitas a sombreamento parcial. Trabalhos recentes, como [9], evidenciam o desempenho do *YOLOv8* em tarefas de deteção de defeitos e sujidade, alcançando valores elevados de *mAP* com tempos de inferência reduzidos, o que confirma o potencial destas abordagens para aplicações em tempo real.

Para além da sujidade superficial e das anomalias visíveis a olho nu, há também um interesse crescente em utilizar visão computacional para apoiar o diagnóstico de falhas elétricas e estruturais. O trabalho de [13], por exemplo, apresenta uma abordagem baseada em DL para deteção de falhas em sistemas PV, integrando análise automática de imagens com algoritmos de classificação supervisionada. Estas soluções tendem a ser integradas em arquiteturas mais abrangentes, onde a visão computacional é um dos módulos de um sistema de manutenção inteligente, articulando-se com dados elétricos, históricos de produção e registos de intervenção para apoiar decisões de O&M.

De forma geral, pode afirmar-se que os sistemas modernos de visão computacional para inspeção de centrais fotovoltaicas seguem um *pipeline* típico que inclui:

1. Aquisição de imagens (*RGB*, térmicas ou multimodais),
2. Pré-processamento para correção geométrica e radiométrica,
3. Aplicação de técnicas de data *augmentation* para aumentar a diversidade dos dados de treino,
4. Utilização de modelos de deteção ou segmentação para identificar e localizar anomalias, e
5. Um módulo de pós-processamento e agregação de resultados que disponibiliza informação útil para as equipas de O&M.

Estes avanços tecnológicos têm impulsionado o desenvolvimento de sistemas automáticos de inspeção capazes de reduzir substancialmente a intervenção humana, otimizar a deteção de anomalias e contribuir para uma manutenção mais eficiente e preventiva das centrais fotovoltaicas. No contexto nacional, onde a energia solar apresenta um crescimento acelerado e se assumiu como um dos pilares da transição energética [1] [26], a integração de soluções baseadas em visão computacional e inteligência artificial torna-se um fator chave para garantir a sustentabilidade e a competitividade económica das instalações. Nos pontos seguintes aprofunda-se a discussão sobre os modelos de deteção de objetos que materializam estas capacidades, com particular enfoque nas abordagens *two-stage* e *one-stage* que foram posteriormente consideradas na implementação desenvolvida neste trabalho.

3.2 Modelos de Detecção de Objetos

A detecção de objetos é uma das tarefas centrais da visão computacional moderna. O seu objetivo é identificar, dentro de uma imagem, as posições e as classes dos elementos de interesse, normalmente através de caixas delimitadoras (*bounding boxes*) e de pontuações de confiança associadas. Em sistemas PV, estes objetos podem corresponder a zonas de sujidade, folhas, excrementos, sombras localizadas ou até fissuras e delaminações visíveis, cuja presença conduz à degradação do desempenho dos módulos e, em casos extremos, a falhas permanentes. Ao contrário de uma simples classificação de imagem, onde se responde apenas “este painel está sujo ou limpo”, a detecção permite localizar onde se encontram os problemas e que tipo de anomalia está presente, informação essencial para planear intervenções de manutenção de forma seletiva.

Do ponto de vista algorítmico, a detecção de objetos pode ser vista como uma extensão conjunta de duas tarefas: localização (estimar coordenadas de caixas) e classificação (atribuir uma etiqueta a cada caixa). A qualidade destas predições é tipicamente avaliada com métricas como a *Intersection-over-Union* (IoU) entre caixas preditas e verdade-terreno, e com medidas agregadas como o *mAP*, que combinam precisão e *recall* em diferentes limiares de IoU. Estas métricas são precisamente as utilizadas nos estudos analisados no contexto da inspeção de painéis PV, o que permitiu comparar de forma objetiva arquiteturas alternativas e justificar a escolha de modelos mais recentes, como o YOLOv8, face a abordagens anteriores.

A maioria das arquiteturas modernas de detecção assenta em *CNN*, cuja estrutura genérica se encontra ilustrada na **Figura 9**. Numa perspetiva simplificada, as camadas convolucionais extraem padrões visuais progressivamente mais abstratos (bordos, texturas, formas), as camadas de *pooling* reduzem a resolução espacial concentrando a informação relevante, e as camadas totalmente ligadas (*fully connected*) realizam a decisão final de classificação. Na detecção de objetos, este “esqueleto” é estendido com cabeças específicas para regressão das coordenadas das caixas e estimativa das probabilidades de cada classe, resultando em arquiteturas capazes de, a partir de uma única imagem, devolver simultaneamente múltiplas predições de objetos.

Além disso, arquiteturas modernas introduzem componentes adicionais – como blocos de atenção, pirâmides de características multi-escala e *necks* especializados – que permitem lidar com a grande variabilidade de tamanho, contraste e posição dos objetos. No caso particular da inspeção de módulos PV, estas extensões são fundamentais para conseguir detetar tanto manchas pequenas de sujidade difusa como áreas extensas de sombras, mantendo robustez a reflexos intensos, padrões repetitivos das células e variações de iluminação entre diferentes horários de aquisição.

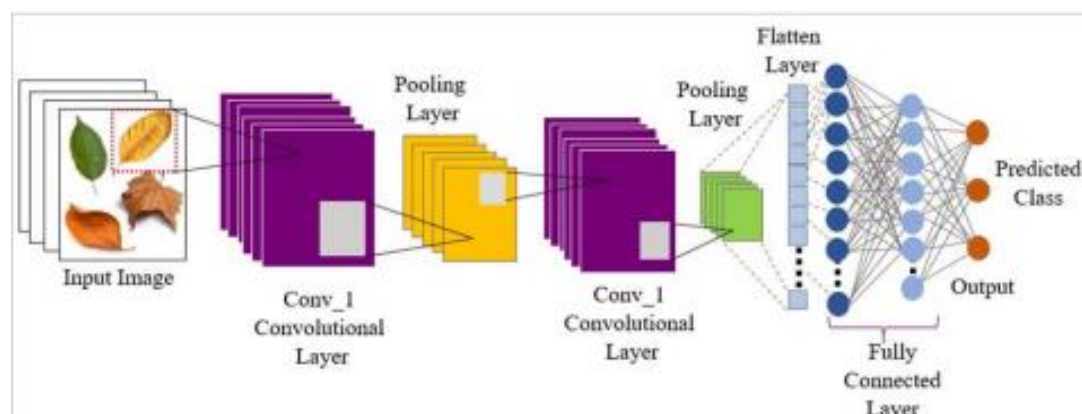


Figura 9 - Estrutura genérica de uma rede neuronal convolucional (CNN), composta por camadas de convolução, pooling e classificação. Adaptado de [7]

Com a evolução do DL, surgiram arquiteturas que combinam CNNs com mecanismos de atenção, fusão multiescala e cabeças de detecção especializadas, otimizadas para lidar com objetos de diferentes dimensões, perspectivas e condições de iluminação. De forma agregada, os principais modelos utilizados na literatura podem ser organizados em três grandes paradigmas, ilustrados na **Figura 10**:

1. *Two-Stage Detectors* — operam em duas fases: numa primeira etapa, é gerado um conjunto de propostas de regiões (*region proposals*) com elevada probabilidade de conter objetos; numa segunda fase, cada região é refinada e classificada de forma mais precisa. Esta família inclui modelos como *Faster Region-based Convolutional Neural Network (Faster R-CNN)* e *Mask R-CNN*, frequentemente usados como *baseline* de alta precisão em contextos de detecção de defeitos finos.
2. *One-Stage Detectors* — eliminam a etapa explícita de propostas e realizam, numa única passagem pela rede, a previsão direta de caixas e classes para um conjunto denso de posições no mapa de características. Aqui enquadram-se modelos como *YOLOv5*, *YOLOv8* e *SSD*, amplamente explorados na literatura de inspeção fotovoltaica devido ao seu compromisso favorável entre *mAP* e tempo de inferência.
3. Modelos baseados em *Transformers (Vision Transformers)* — utilizam mecanismos de atenção global para modelar relações espaciais entre regiões da imagem, captando dependências de longo alcance que complementam a natureza local das convoluções. Abordagens como as baseadas em *DINOv2/ViT* têm sido avaliadas para detecção de padrões de poeira e manchas pouco contrastadas em painéis solares, explorando a capacidade destes modelos para generalizar entre condições de iluminação e textura muito distintas.

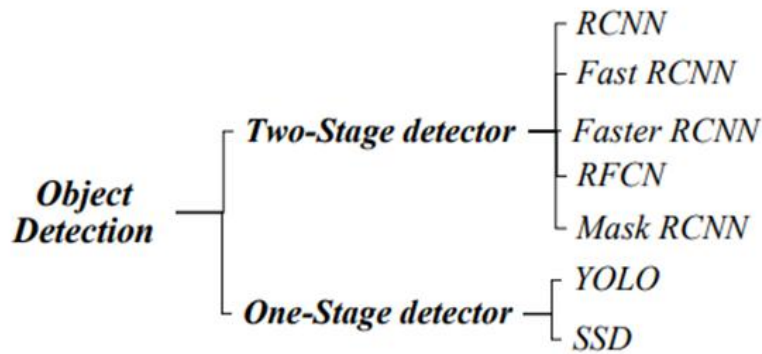


Figura 10 - Esquema simplificado dos detectores *Two-Stage* (*Faster/Mask-R-CNN*). Adaptado de [7]

Independentemente da família, o *pipeline* de detecção pode ser decomposto em três blocos conceituais: um *backbone*, um *neck* e uma *head* de detecção. O *backbone* é responsável por extrair mapas de características a partir da imagem de entrada, utilizando camadas convolucionais sucessivas. Em muitos dos trabalhos analisados recorre-se a variantes otimizadas de redes como *CSPDarknet* (*Cross Stage Partial Darknet*) ou *ResNet*, como se observa em [7], [9]. O *neck* combina informação em múltiplas escalas, recorrendo tipicamente a pirâmides de características (*Feature Pyramid Network (FPN)*, *PAN*) para permitir que o modelo detete simultaneamente objetos pequenos (por exemplo, manchas localizadas ou excrementos) e estruturas maiores, como zonas de sombra ou áreas extensas de sujidade. Por fim, a *head* de detecção produz, para cada localização relevante, as probabilidades de classe e os parâmetros das caixas delimitadoras, podendo assumir formas distintas consoante se trate de arquiteturas *Two-Stage*, *One-Stage* ou baseadas em *Transformers*.

No contexto deste trabalho, esta decomposição em *backbone–neck–head* foi particularmente relevante para compreender as diferenças entre abordagens clássicas (*Two-Stage*), modelos otimizados para tempo real (*YOLO*, *SSD*) e arquiteturas mais recentes baseadas em atenção. No caso das soluções *One-Stage* aplicadas a painéis PV, como em [12] ou [10], as melhorias de desempenho decorrem precisamente da afinação conjunta destes três blocos: *backbones* mais leves, *necks* multiescala mais eficientes e *heads* desacopladas para regressão e classificação. Já em abordagens baseadas em *Transformers*, como [14], o papel do *backbone* é assumido por uma pilha de blocos de atenção global, mantendo-se, no entanto, a mesma lógica modular de extração de características, combinação multiescala e predição final de caixas.

3.2.1 Modelos *Two-Stage* Detection

Os detetores *Two-Stage* foram pioneiros na deteção moderna de objetos e estabeleceram, durante vários anos, a referência em termos de precisão, sobretudo em objetos pequenos, parcialmente ocluídos ou pouco contrastados. A ideia central é decompor o problema em duas fases logicamente separadas: numa primeira etapa propõem-se regiões com elevada probabilidade de conter objetos (*region proposals*); numa segunda etapa essas regiões são refinadas e classificadas com maior detalhe. Esta decomposição permite uma análise fina das regiões de interesse (*Regions of Interest – ROI*) e uma localização espacial muito precisa das caixas de deteção.

Na primeira fase, uma *CNN* inicial (*backbone*) transforma a imagem em mapas de características (*feature maps*). Sobre estes mapas atua uma *Region Proposal Network* (RPN), que varre o espaço com *anchors* de múltiplas escalas e rácios, produzindo propostas de regiões com pontuações de “*objectness*” e ajustes de offsets (x, y, w, h). A qualidade de cada proposta é avaliada através da métrica *Intersection-over-Union* (*IoU*), comparando a sobreposição entre a caixa proposta e a verdade-terreno; propostas com *IoU* elevada são marcadas como positivas, com *IoU* baixa como negativas, e as intermédias são tipicamente ignoradas no treino. Em seguida, aplica-se *Non-Maximum Suppression* (NMS) para eliminar caixas redundantes e manter apenas algumas centenas de *ROIs* com maior probabilidade. Este enquadramento conceptual – e a distinção entre abordagens *Two-Stage* e *One-Stage* – é precisamente o que é ilustrado nos esquemas comparativos de modelos de deteção presentes em vários trabalhos sobre YOLO aplicados a *soiling* (ver **Figura 11**) [7].

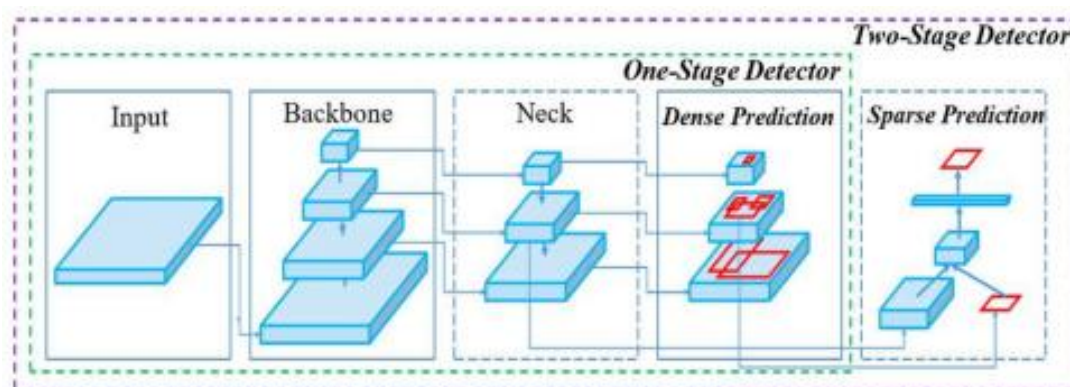


Figura 11 - Classificação geral dos modelos de deteção de objetos e distinção entre abordagens *Two-Stage* e *One-Stage*. Adaptado de [7]

Na segunda fase, cada *ROI* é re-amostrada e alinhada ao mapa de características através de operações como o *ROIAlign*, reduzindo erros de quantização que ocorriam em variantes anteriores (por exemplo, *ROI Pool*). Estas regiões normalizadas alimentam cabeças específicas (*detection heads*) que, para

cada *ROI*, produzem simultaneamente a classe do objeto e o refinamento das coordenadas da caixa por regressão. No caso do *Mask R-CNN*, acrescenta-se ainda uma *mask head* que prevê máscaras binárias por pixel, permitindo segmentar objetos com elevada resolução espacial. Em contexto PV, esta capacidade de segmentação detalhada é particularmente útil para identificar fissuras finas, delaminações ou zonas de hot spot visíveis em imagens de alta resolução, indo além da simples marcação de uma caixa aproximadamente localizada.

O grande trunfo dos modelos *Two-Stage* é, portanto, a elevada precisão espacial e o maior *recall* em objetos pequenos, graças à combinação entre a RPN e o refinamento por *ROI*. Contudo, esta vantagem tem um custo: a latência é superior, devido à segunda fase explícita de processamento e ao número elevado de *ROIs* por imagem, o que penaliza aplicações *edge* e operações com *drones* que exigem *throughput* em tempo (quase) real. Esta troca desempenho/velocidade é claramente evidenciada nos estudos analisados que comparam linhas *Two-Stage* com alternativas *One-Stage*. No trabalho “Enhancing Dust Detection on Photovoltaic Panels with PP-YOLO: A Deep Learning Approach” (a referência que tens associada ao [PP-YOLO/YOLOv5/SSD/Faster-R-CNN]), o *Faster-R-CNN* apresenta boa precisão, mas opera a cerca de 26,7 FPS, ficando significativamente abaixo dos modelos *single-stage* em velocidade: *PP-YOLO* atinge aproximadamente 126 FPS, *YOLOv5* cerca de 96 FPS e *SSD* perto de 76 FPS, mantendo níveis de *mAP* competitivos. Estes resultados evidenciam a limitação prática dos *Two-Stage* para inspeção contínua com UAV ou sistemas embarcados [27].

De forma semelhante, o artigo [8] analisa um *SSD* melhorado em comparação com *Faster-R-CNN* e outras arquiteturas *Two-Stage*, concluindo que, embora estes últimos alcancem bons valores de precisão e *recall*, o custo computacional e a latência tornam-se impeditivos em cenários com grande volume de imagens, típicos de centrais fotovoltaicas de larga escala. Os autores mostram que variações *One-Stage* com mecanismos de atenção e *transfer learning* conseguem preservar uma exatidão elevada com menor tempo de processamento, o que motiva a adoção preferencial destas arquiteturas em sistemas de inspeção automática [8].

3.2.2 Modelos *One-Stage Detection*

Os modelos *One-Stage Detection* surgiram como uma resposta direta às limitações de velocidade e complexidade dos detetores *Two-Stage*. Enquanto arquiteturas tradicionais – como o *Faster R-CNN* ou o *Mask R-CNN* – dividem a tarefa em duas fases sequenciais (proposta e classificação), os modelos *One-Stage* executam toda a deteção num único processo de inferência, tratando a identificação de objetos como uma tarefa de regressão direta sobre o mapa de características.

Em termos práticos, o processo é estruturado em três etapas fundamentais:

1. Extração de características (*Feature Extraction*): um *backbone* convolucional — normalmente uma rede leve e otimizada, como *CSPDarknet* ou *VGG-16* — produz mapas de características a diferentes escalas, captando padrões texturais, bordos e estruturas relevantes na imagem.
2. Agregação multiescala (*Feature Fusion*): a informação dessas várias resoluções é combinada através de módulos como FPN ou *PANet*, permitindo ao modelo detetar simultaneamente objetos grandes (por exemplo, sombras ou áreas extensas de sujidade) e detalhes finos, como manchas localizadas ou excrementos de aves.
3. Predição direta (*Dense Prediction*): para cada célula do mapa, o modelo gera previsões de caixas, confiança e classes, que são posteriormente filtradas por NMS para remover sobreposições redundantes.

Esta filosofia confere aos *One-Stage Detectors* uma vantagem clara em termos de velocidade e simplicidade arquitetónica, tornando-os especialmente adequados para sistemas de monitorização em tempo real, como *drones* de inspeção, robôs móveis ou câmaras fixas em sistemas PV, onde é essencial processar fluxos de imagem contínuos com baixo consumo de energia e latência reduzida.

A nível conceptual, estas arquiteturas podem ser agrupadas em duas grandes famílias — *YOLO (You Only Look Once)* e *SSD (Single Shot MultiBox Detector)* — que partilham a filosofia de deteção direta, mas diferem na forma como estruturam as camadas de extração e fusão de características.

O *YOLOv8* aprofunda esta linha evolutiva, introduzindo *backbones* mais leves, blocos *C2f*, *heads* desacopladas e deteção *anchor-free*. No trabalho [9], é proposta uma arquitetura otimizada para operação em campo, reportando *mAP* na ordem dos 95 %, *recall* ~94 % e boa robustez face a variações de iluminação e perspetiva. De forma complementar, em [12], o modelo é integrado num *pipeline* embarcado em UAV, atingindo taxas de deteção superiores a 90 % e cerca de 30 FPS em *Graphics Processing Unit* (GPU) embarcada, o que confirma a adequação desta família de modelos para inspeções aéreas rápidas com *drones*. Já o estudo [10] explora diferentes variantes da família *YOLO* para identificação de partículas indesejadas, evidenciando boa generalização a diferentes tipos de sujidade e condições ambientais.

A **Figura 12** apresenta, como exemplo, a arquitetura detalhada do *YOLOv8*, um dos modelos mais representativos da família *One-Stage*, onde se observam os módulos principais responsáveis pelo fluxo de deteção (*Backbone*, *Neck* e *Head*), bem como os blocos *C2f* e as perdas independentes de regressão (*BBox Loss*) e classificação (*Cls Loss*), que permitem otimizar, em paralelo, a qualidade das caixas e a separação entre classes.

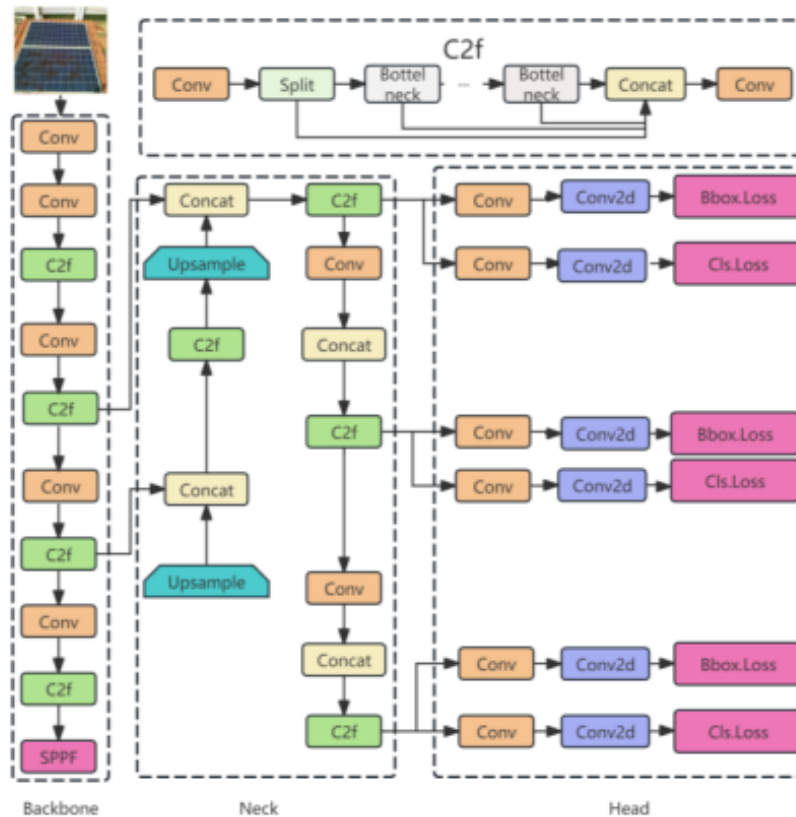


Figura 12 - Arquitetura detalhada do modelo YOLOv8, composta pelos módulos *Backbone*, *Neck* e *Head*, e incorporando blocos C2f, operações de upsampling e perdas independentes de regressão (BBox Loss) e classificação (Cls Loss). Adaptado [9]

Em paralelo, o SSD representa uma alternativa *One-Stage* com filosofia semelhante, mas com uma organização distinta das escalas de detecção. Em vez de concentrar as predições em poucas camadas de alta resolução, o SSD distribui as caixas por vários *feature maps* de diferentes tamanhos, com *default boxes* (*anchors*) adaptadas à dimensão típica dos objetos em cada escala. No [8] é proposta uma versão aprimorada do SSD, onde se integram módulos de atenção entre as camadas convolucionais e se ajustam as escalas das caixas padrão à geometria dos defeitos em painéis solares. Os autores reportam ganhos de *mAP* face ao SSD original, mantendo latências competitivas, o que torna este modelo particularmente interessante para sistemas fixos de monitorização contínua, em que a estabilidade das deteções ao longo do tempo é tão importante como a velocidade bruta.

A **Figura 13** ilustra a arquitetura deste SSD melhorado, onde se destacam os blocos de atenção intercalados e o processo final de NMS, conforme descrito em [8]. Em conjunto, as Figuras, **Figura 12** e **Figura 13**, sintetizam as duas linhas principais de desenvolvimento dos detetores *One-Stage* aplicados à inspeção de centrais fotovoltaicas: por um lado, modelos fortemente otimizados para tempo real e plataformas embarcadas (YOLOv5/YOLOv8); por outro, variantes mais estáveis e modulares, adequadas a infraestruturas fixas de monitorização (SSD melhorado).

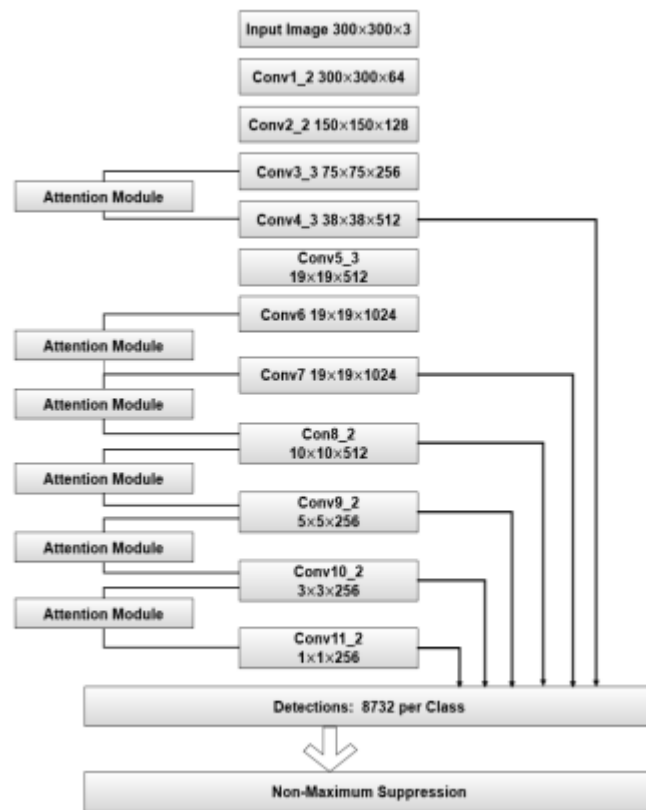


Figura 13 - Arquitetura do modelo SSD melhorado com integração de módulos de atenção e detecção multi-escala. Adaptado de [8]

No conjunto dos seis estudos selecionados na revisão sistemática, observa-se uma tendência clara de consolidação dos modelos *One-Stage*, em particular da família *YOLO*, como solução preferencial quando o objetivo é conciliar elevada precisão, baixa latência e implementação prática em cenários reais de inspeção fotovoltaica. Esta evidência empírica sustentou a opção, no presente trabalho, por uma arquitetura da família *YOLO* como base do sistema de detecção de sujidade desenvolvido, análise que será retomada na comparação crítica da secção 2.6.

4. Avaliação de Desempenho e Seleção do Modelo

Este capítulo tem como objetivo apresentar, de forma estruturada e verificável, a avaliação de desempenho dos modelos considerados para a detecção (e, quando aplicável, segmentação) de sujidade e anomalias visíveis em módulos PV, bem como justificar a seleção do modelo a adotar no protótipo final. Em vez de partir de uma escolha arquitetural “por omissão”, o capítulo organiza-se a partir daquilo que, na prática, determina o sucesso do sistema: o *dataset* e o protocolo experimental usados para treinar, comparar e interpretar resultados.

A avaliação assenta num *dataset* construído especificamente para este caso de uso, obtido através da fusão de dois *datasets* públicos disponibilizados na plataforma *Roboflow* e posteriormente enriquecido com novas imagens e anotações adicionais realizadas no próprio *Roboflow*. Este processo — que será ilustrado com figuras representativas do *workflow* de anotação e exemplos de imagens anotadas — é descrito de modo a tornar explícitos os critérios adotados, as decisões de harmonização de classes e as opções que garantem consistência e reprodutibilidade. Em paralelo, serão apresentados os procedimentos de preparação dos dados e as escolhas de treino que permitem comparar modelos de forma justa, evitando conclusões frágeis por diferenças de configuração ou de *split*.

Com o terreno fixado, o capítulo introduz os modelos avaliados e o racional da comparação: *YOLOv8* como *baseline* operacional pela maturidade do ecossistema *Ultralytics*, *YOLOv11* como alternativa mais recente dentro da mesma família e *YOLOv11-seg* como opção orientada à segmentação quando a geometria da sujidade é mais informativa do que uma *BBox*. Adicionalmente, foi incluída uma abordagem baseada em *transformers* (*DINOv2*) como linha comparativa com características distintas de robustez e custo computacional. A apresentação das métricas e resultados foi suportada por tabelas comparativas e por gráficos extraídos diretamente dos treinos (curvas de *loss*/métricas, *PR curve*, matriz de confusão quando aplicável), complementados por gráficos agregados construídos a partir dos resultados finais, para tornar a comparação legível e auditável. A discussão culmina na decisão do modelo selecionado e nas implicações dessa escolha para o sistema, mantendo ainda em aberto a evolução futura para arquiteturas emergentes como o *YOLO26*, discutidas como continuidade possível, mas fora do núcleo experimental deste capítulo.

4.1 Descrição do *Dataset*

A avaliação apresentada neste capítulo assenta num *dataset* construído especificamente para o problema em causa: detecção de sujidade e anomalias visíveis em módulos PV a partir de imagens *RGB*. A decisão de não treinar “em cima” de um único *dataset* público, tal como encontrado, não foi capricho metodológico — foi uma resposta direta ao desalinhamento típico entre *datasets*

genéricos (classes, perspectivas, condições de captura e critérios de anotação) e o que um sistema de inspeção em contexto real precisa de ver e distinguir. Assim, em vez de adotar um conjunto de dados pré-existente como verdade absoluta, foi seguido um processo controlado de composição, harmonização e enriquecimento, de modo a produzir um *dataset* final consistente, auditável e alinhado com o caso de uso.

O ponto de partida foi a exploração de *datasets* públicos disponibilizados na plataforma *Roboflow* [18], pela vantagem pragmática de concentrar, num único ecossistema, o acesso a *datasets*, ferramentas de anotação e *pipelines* de exportação reproduzíveis. Esta escolha reduziu fricção técnica e, mais importante, permitiu documentar o “como” com evidência visual (*screenshots* do *workflow*, versões do *dataset* e exemplos de anotações), em vez de pedir ao leitor um ato de fé. A plataforma foi também usada como ambiente central de anotação e gestão de versões do *dataset*, assegurando que as transformações aplicadas (adição de imagens, correções, alterações de classes e re-exports) ficam registadas e podem ser reproduzidas.

A definição do problema foi estabilizada logo à partida numa taxonomia simples e operacional, composta por três classes: *clean-solar-panel*, *dust-solar-panel* e *bird-drop*. Esta escolha procurou equilibrar utilidade prática e viabilidade de anotação: por um lado, separou o “painel limpo” do “painel com sujidade relevante” (distinção essencial para decisões de limpeza); por outro, isolou *bird-drop* como categoria específica por representar um padrão visual frequentemente mais localizado e intenso, com comportamento distinto de poeira difusa e com impacto potencialmente desproporcional em áreas pequenas do módulo. Ao fixar estas três classes, o *dataset* passou a ter uma semântica clara: o objetivo não foi “detetar tudo o que parece estranho”, mas sim distinguir estados visuais com significado operacional e suficiente consistência para treino supervisionado [29] [30].

Para além de definir que classes se pretende distinguir, foi crucial explicitar o que constitui uma instância anotada. Neste *dataset*, a unidade de anotação foi o módulo PV (a área do painel), e não “manchas” isoladas de sujidade. A classe (*clean*, *dust* ou *bird-drop*) descreveu o estado visual dominante do módulo observado na imagem. Esta decisão foi deliberada e operacional: em O&M, a ação concreta raramente depende de contornar cada depósito de poeira; depende, isso sim, de identificar módulos cuja condição visual justifica limpeza, reinspeção ou priorização. Ao mesmo tempo, a anotação foi realizada com polígonos (máscaras), e não apenas com *BBoxes*, o que preserva duas vantagens: (i) permitiu derivar caixas delimitadoras para treino de detetores clássicos (YOLO), e (ii) manteve compatibilidade direta com variantes de segmentação (p.ex., YOLOv11-seg), caso se verifique que a geometria do módulo/oclusão é informativa para reduzir falsos positivos em reflexos, sombras e fundos complexos.

As Figuras, **Figura 14**, **Figura 15** e **Figura 16** tornam este critério observável e, sobretudo, verificável: em cada exemplo, vê-se a imagem original com o contorno

poligonal desenhado sobre o módulo e, em paralelo, a visualização da *layer* de segmentação (máscara preenchida) correspondente à classe atribuída. Esta dualidade — anotação “sobre a imagem” e máscara “como layer” — não foi apenas uma questão estética do *Roboflow*; é a forma prática de garantir que a anotação é simultaneamente interpretável por humanos e diretamente consumível por modelos de segmentação. Ao usar máscaras por classe (*dust-solar-panel*, *clean-solar-panel* e *bird-drop*), o *dataset* preserva informação geométrica do módulo (inclinação, oclusões, recortes parciais e limites reais), o que foi particularmente relevante em cenários onde o fundo compete visualmente com o painel (vegetação, estruturas, telhados) e onde reflexos especulares ou sombras duras podem “fabricar” falsos contornos. Foi também aqui que a escolha de anotar o módulo inteiro se revela consequente: o modelo aprende a reconhecer um estado do painel como unidade operacional, e não a perseguir micro-regiões ambíguas de sujeira. Em termos de reprodutibilidade, estas figuras cristalizam aquilo que muitas vezes fica implícito em *datasets* públicos: o que conta como exemplo de cada classe e como esse julgamento foi traduzido numa máscara consistente. Isto reduz o espaço para arbitrariedade e antecipa, desde já, onde os modelos tenderão a falhar: precisamente nos casos-limite em que poeira difusa, *bird-drop* parcial, reflexos e sombra se sobrepõem e forçam o sistema a decidir com base num critério que tem de ser o mesmo em todo o *dataset*.

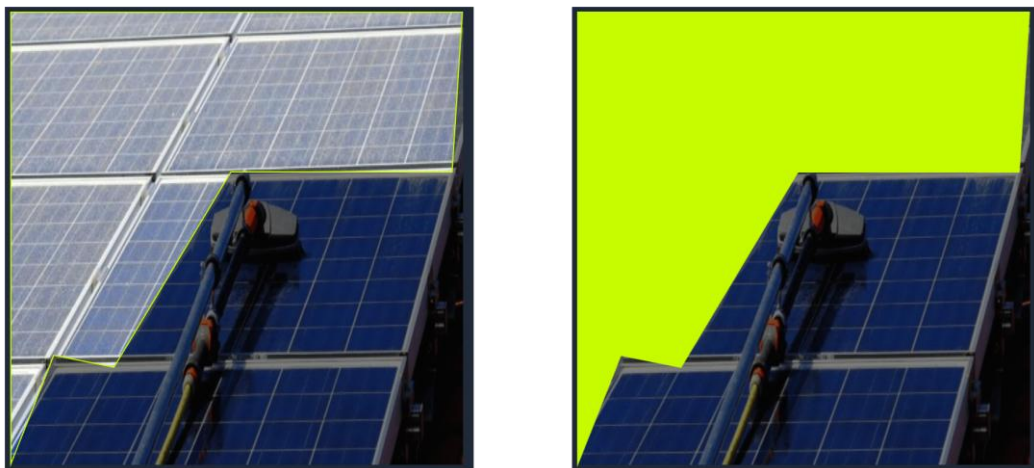


Figura 14 - Exemplos de anotações no *Roboflow* para a classe *dust-solar-panel*

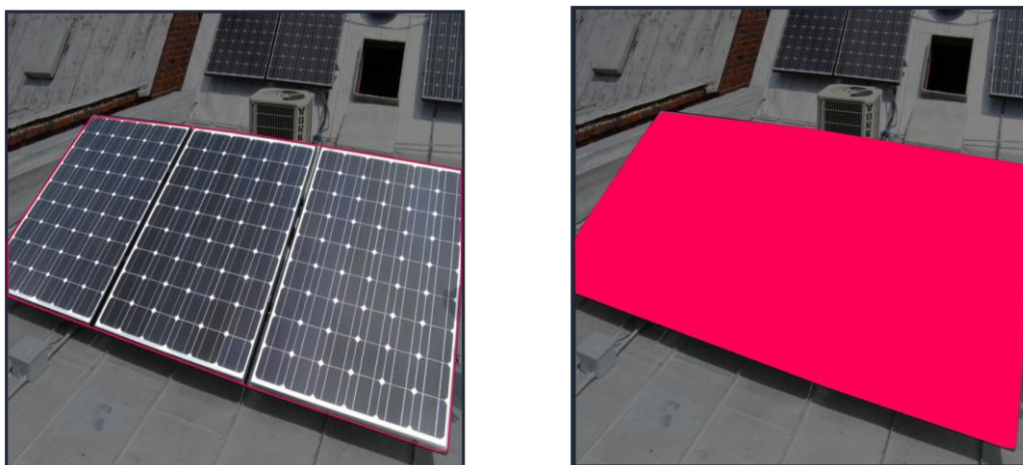


Figura 15 - Exemplos de anotações no *Roboflow* para a classe *clean-solar-panel*

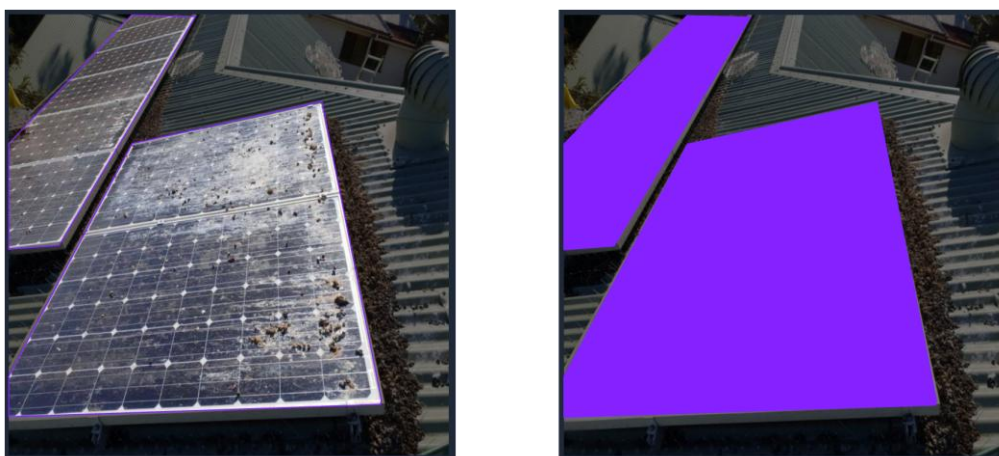


Figura 16 - Exemplos de anotações no *Roboflow* para a classe *bird-drop*

A construção do *dataset* começou com a seleção de dois *datasets* públicos do *Roboflow Universe*. O primeiro, orientado para sujidade em painéis (Solar Panel Dust Detection), forneceu uma base ampla de imagens associadas a condições de limpeza e poeira, contribuindo sobretudo para consolidar o contraste entre *clean* e *dust*. O segundo *dataset*, centrado em dano/segmentação (Solar Panel), foi utilizado de forma seletiva: embora incluía múltiplas classes de defeitos e condições, apenas foram retidas as imagens correspondentes às classes *dust* e *bird-drop*, excluindo explicitamente categorias que não interessavam ao objetivo deste trabalho (por exemplo, classes associadas a defeitos elétricos/físicos e outras condições não relacionadas com sujidade). Esta filtragem não é um pormenor administrativo — é uma decisão metodológica para impedir que o *dataset* final herde ruído conceptual e “puxe” o modelo para um problema diferente daquele que se pretende resolver. Depois de agregadas as imagens provenientes destes dois conjuntos públicos, tornou-se evidente um desequilíbrio: a classe *bird-drop* surgia com menor representatividade, tanto em volume como em diversidade de cenários,

o que criaria um viés previsível no treino (bons resultados médios à custa de falha sistemática na classe rara). Para contrariar esse efeito, foram reunidas imagens adicionais a partir de outras fontes/meios, com o objetivo específico de popular a classe *bird-drop* com variação realista de escala, contraste e contexto (incluindo casos em que os excrementos surgem parcialmente confundidos com reflexos, sombra ou padrões do próprio painel). Esta etapa de enriquecimento é particularmente relevante porque, num problema visual como este, a raridade não é apenas estatística: é também semântica. Um modelo pode “ver” *bird-drop* como um caso extremo de *dusty* se nunca for forçado a aprender o que distingue, de facto, as duas categorias.

Com todas as imagens reunidas, o *dataset* foi inicialmente organizado e estruturado localmente (normalização de nomes, separação por origem e preparação para importação), sendo depois carregado na plataforma *Roboflow* para anotação. Esta sequência — primeiro consolidar localmente, depois anotar numa plataforma com versionamento — permitiu manter controlo sobre a integridade do conjunto (evitando duplicados evidentes e incongruências básicas) e, ao mesmo tempo, beneficiar da rastreabilidade e do *workflow* de anotação do *Roboflow*. A anotação foi conduzida com foco na consistência: a utilidade do *dataset* não depende apenas de “ter labels”, mas de as labels obedecerem ao mesmo critério em imagens diferentes, sobretudo nos casos-limite (sujidade difusa, reflexos fortes, sombras e zonas de baixo contraste), onde a tentação de anotar “ao sabor do olho” é maior e onde os modelos tendem a falhar mais.

Concluída a consolidação e anotação, o *dataset* final foi particionado em treino, validação e teste, mantendo-se a mesma divisão em todas as experiências deste capítulo para garantir comparabilidade entre modelos e configurações. Para além de assegurar separação entre dados vistos e não vistos, este particionamento impôs disciplina experimental: qualquer diferença observada entre modelos deveria ser atribuível à arquitetura e ao regime de treino, e não a alterações subtis na amostragem. A Tabela 3 sintetiza esta divisão por split e evidencia ainda a presença residual de imagens sem instâncias anotadas (background), que foram mantidas como negativos naturais no conjunto.

Tabela 3 - Distribuição de imagens no *split* treino, validação e teste

<i>split</i>	imagens	% do total	<i>background</i>	% <i>background</i> no <i>split</i>
treino	2427	70%	76	3%
validação	699	20%	6	1%
teste	355	10%	6	2%
total	3481	100%	88	3%

A leitura da **Tabela 3** deve ser feita com duas preocupações. A primeira é a estabilidade estatística: uma validação com dimensão suficiente para acompanhar a evolução do treino sem se tornar demasiado ruidosa, e um teste que seja, de facto, um “último exame” — pequeno o bastante para ser exigente, mas não tão pequeno que transforme cada caso difícil num salto brusco nas métricas.

No final, o dataset incluiu 3 481 imagens, distribuídas por 2 427 imagens de treino (70%), 699 imagens de validação (20%) e 355 imagens de teste (10%). Dentro de cada *split*, existiram ainda imagens marcadas como *background*, num total de 88 imagens ($\approx 3\%$ do dataset): 76 no treino (3% do *split*), 6 na validação (1%) e 6 no teste (2%).

A segunda é a interpretação do *background*: estas imagens não são “sobras”, são um mecanismo de pressão útil para testar a tendência do modelo para alucinar deteções em padrões frequentes do domínio PV, como reflexos especulares, sombras duras e repetição geométrica das células.

Em paralelo, importa caracterizar o dataset ao nível das instâncias anotadas, porque o número de imagens não traduz diretamente a quantidade de supervisão: várias imagens incluem mais do que um módulo anotado, pelo que a densidade de instâncias por classe é o que realmente condiciona a aprendizagem e a leitura de métricas por classe. **Tabela 4** apresenta a distribuição de anotações por classe e por *split*, permitindo avaliar tanto o equilíbrio global do *dataset* como a robustez estatística da avaliação por classe, sobretudo para *bird-drop*, que foi a classe alvo do reforço com imagens adicionais.

Tabela 4 - Distribuição de anotações por classe e por *split* no *dataset* final

classe	treino	validação	teste	total
<i>clean</i>	1251	355	174	1780
<i>dusty</i>	1092	321	158	1571
<i>bird-drop</i>	954	246	140	1340
total	3297	922	472	4691

A leitura conjunta das **Tabela 3** e **Tabela 4** deu contexto ao desempenho que se observou nos treinos, porque clarificou o “peso estatístico” real do problema. O *dataset* teve dimensão suficiente para sustentar uma avaliação estável — 3 481 imagens e 4 691 instâncias anotadas — mas não foi um *dataset* massivo que perdoe escolhas descuidadas: *augmentation*, regularização e *early stopping* deixaram de ser termos decorativos e passaram a ser mecanismos de controlo de sobreajuste com impacto direto na generalização. A própria divisão 70%/20%/10% ajudou a manter a comparação honesta entre modelos, mas colocou uma exigência: as curvas de treino e as métricas por classe tiveram de ser interpretadas à luz do número de instâncias disponíveis em validação e teste, especialmente quando se discutem ganhos pequenos de *mAP* que podem, neste regime, ser sensíveis a ruído de anotação ou variações de configuração. Ao nível das classes, existiu desequilíbrio, mas não um desequilíbrio patológico: *bird-drop* continuou a ser a classe com menor representatividade, porém com volume suficiente para permitir leitura por classe sem cair num regime em que poucos exemplos definem a narrativa. Isto é importante porque, neste domínio, o risco típico é o modelo aprender uma heurística preguiçosa — tratar *bird-drop* como “poeira intensa” — se não for forçado a ver variedade suficiente de casos reais, com diferenças de contraste, escala e confusão com sombras e brilho. Finalmente, as imagens marcadas como *background* funcionaram como negativos naturais do *pipeline*: são um teste silencioso à tendência do modelo para “ver sujidade” em artefactos frequentes do cenário PV, antecipando um tipo de erro que, em campo, tem custo operacional concreto — o falso positivo que desencadeia uma limpeza ou reinspeção sem necessidade.

4.2 Métricas de Desempenho

A avaliação dos modelos neste trabalho foi desenhada para medir, com precisão, aquilo que interessa numa tarefa de inspeção visual de módulos PV: distinguir corretamente o estado do módulo (no *dataset* surgem como *clean-solar-panel*, *dusty-solar-panel* e *bird-drop*) e fazê-lo de forma suficientemente fiável para

suportar decisões operacionais. Para esse efeito, recorreu-se às métricas padrão de deteção e segmentação, tal como reportadas pela infraestrutura *Ultralytics*, analisando resultados globais e por classe [20] [21].

Em termos conceptuais, todas estas métricas assentam na mesma lógica: para cada imagem, o modelo produziu predições (caixas delimitadoras e/ou máscaras), cada uma com uma classe e um valor de confiança, e essas predições foram comparadas com as anotações de referência. A correspondência entre uma predição e a verdade-terreno foi estabelecida através de um critério de sobreposição (IoU) e de um limiar t — neste trabalho, t representou o limiar de IoU usado na avaliação. Só quando a sobreposição é suficiente ($\text{IoU} \geq t$) uma predição pôde ser emparelhada com uma anotação e, por conseguinte, contribuir para os contadores de acertos e erros.

Importa ainda explicitar uma regra de emparelhamento que afeta diretamente os contadores: cada anotação de referência pode ser emparelhada, no máximo, com uma única predição válida. Quando existem várias predições sobre a mesma instância anotada (por exemplo, várias caixas a “cobrir” o mesmo módulo), apenas a melhor correspondência é contabilizada como *true positive* — na prática, a que cumpre $\text{IoU} \geq t$ e surge com maior confiança no processo de varrimento. As restantes duplicações passaram a contar como *false positives* (FP). Esta restrição impediu que um modelo “infle” artificialmente os *TP* produzindo duplicados e ligou-se diretamente ao papel do pós-processamento (NMS e limiares de confiança) na estabilidade das métricas.

Uma predição conta como *True Positive* (TP) quando o modelo identifica um módulo anotado com a classe correta e com sobreposição suficiente para ser considerada uma correspondência válida. Um FP ocorre quando o modelo emite uma deteção que não corresponde a nenhuma anotação válida — por exemplo, porque “vê” um objeto onde não há anotação, porque localiza mal a região e falha $\text{IoU} \geq t$, ou porque atribui a classe errada a uma instância real. Um False Negative (FN) representa o erro oposto: existe um módulo anotado na imagem, mas o modelo não o deteta, ou deteta-o de forma tão imprecisa que não atinge o critério de correspondência.

Aqui convém não confundir dois “limiares” diferentes: t é o limiar de IoU (critério geométrico de correspondência), enquanto o limiar de confiança é o que se varre para construir a curva P–R e calcular *AP/mAP*.

A *Precision* (P) mede a proporção de deteções corretas entre todas as deteções emitidas pelo modelo, quantificando a fiabilidade do que o sistema “afirma” ver. Num contexto PV, uma *precision* baixa traduz-se em alarmes falsos recorrentes — por exemplo, reflexos, sombras ou padrões das células interpretados como sujidade — com impacto operacional direto, porque pode induzir limpezas ou reinspeções desnecessárias. A *precision* é definida pela Equação 1, na qual TP representa o número de deteções corretas (*true positives*) e FP o número de

deteções incorretas FP, de acordo com o critério de correspondência estabelecido pelo limiar de sobreposição (IoU) e pela classe atribuída.

$$P = \frac{TP}{TP + FP}$$

Equação 1 - Fórmula para o cálculo de *Precision*

O *Recall* (R) mede a proporção de ocorrências reais que o modelo conseguiu recuperar, isto é, quantifica a capacidade do sistema para não deixar escapar instâncias anotadas. Em inspeção fotovoltaica, um *recall* baixo não é apenas um “erro estatístico”: significa que casos que deveriam ser sinalizados ficam fora do resultado final e podem não ser considerados em ações de limpeza, reinspeção ou priorização. Este efeito é particularmente relevante em *bird-drop*, onde ocorrências pequenas, parcialmente ocluídas ou confundidas com reflexos e sombras tendem a ser as primeiras a desaparecer quando o limiar de confiança é demasiado exigente. O *recall* é definido pela **Equação 2**, na qual *TP* representa o número de deteções corretas e *FN* o número de instâncias reais não recuperadas — seja por ausência de deteção, seja por deteção com sobreposição insuficiente para cumprir $IoU \geq t$.

$$R = \frac{TP}{TP + FN}$$

Equação 2 - Fórmula para o cálculo de *Recall*

No contexto PV, esta distinção não é académica: falsos positivos podem traduzir-se em limpezas ou reinspeções desnecessárias, enquanto falsos negativos podem ocultar precisamente ocorrências críticas (por exemplo, *bird-drop* discreto ou parcialmente ocluído). Além disso, *precision* e *recall* não são independentes: variar o limiar de confiança do modelo altera o número de deteções aceites e, portanto, desloca o equilíbrio entre FP e FN. Um limiar mais permissivo costuma aumentar *recall* à custa de mais FP; um limiar mais exigente costuma aumentar *precision* à custa de mais FN. É por isso que, para além de olhar para um único ponto, se avalia o comportamento do modelo ao longo de um intervalo de limiares.

É nesta perspetiva que surgiu a curva precisão–revocação (P–R): para uma dada classe, ordenaram-se as predições por confiança e varreu-se o limiar de aceitação, observando como a *precision* evolui à medida que se tenta recuperar uma fração maior das instâncias (*recall*). A *Average Precision* (AP) resumiu essa curva num único valor, correspondendo, em termos práticos, à área sob a curva P–

R, tal como expresso na **Equação 3**. Embora a definição seja contínua, o AP foi obtido na prática a partir de pontos discretos: à medida que o limiar de confiança baixa, foram sendo acumuladas predições, recalculou-se TP e FP traçando-se a curva $P(R)$. A intuição manteve-se e é operacional: quanto mais o modelo consegue manter precisão elevada enquanto aumenta *recall*, maior será o AP — e melhor foi o seu comportamento quando se tenta reduzir falsos negativos sem desencadear uma explosão de falsos positivos.

$$AP = \int_0^1 P_{(R)} dR$$

Equação 3 - Fórmula para o cálculo de *Average Precision*

Nesta expressão, $P(R)$ representa a precisão como função do *recall* ao longo dessa variação do limiar. Embora a definição seja contínua, na prática o AP é calculado a partir de pontos discretos obtidos durante a varredura do limiar (ou, de forma equivalente, ao percorrer a lista ordenada de predições por confiança), acumulando TP e FP e traçando a curva P – R correspondente. A intuição é simples e útil: quanto mais o modelo conseguir manter *precision* elevada enquanto aumenta *recall*, maior foi o AP — e mais robusto é o seu comportamento quando se tenta reduzir falsos negativos sem desencadear uma explosão de falsos positivos.

Como o *dataset* contém várias classes, foi necessário um critério de agregação que não dependa de uma só. A *mAP* foi a média do AP calculado para cada classe c , produzindo um indicador global que permitiu comparar modelos de forma direta, tal como expresso na **Equação 4**, onde n representa o número de classes e k indexa cada classe na soma:

$$mAP = \frac{1}{n} \sum_{k=1}^{k=n} AP_k$$

Equação 4 - Fórmula para o cálculo de *mAP*.

Neste trabalho reportaram-se duas variantes porque responderam a perguntas diferentes. O *mAP@0.50* (*mAP50*) usou um critério de correspondência relativamente permissivo ($\text{IoU} \geq 0.50$), tendo sido útil para perceber se o modelo “encontra” a região correta de forma geral. Já o *mAP@0.50:0.95* (*mAP50–95*) foi

uma medida mais exigente, porque obriga o modelo a manter desempenho quando o critério geométrico se torna progressivamente mais apertado.

O cálculo é feito de forma hierárquica: para cada limiar de sobreposição $t \in \{0.50, 0.55, \dots, 0.95\}$ calcula-se primeiro o $AP_c@t$ por classe, obtém-se depois $mAP@t$ como média desses valores nas n classes e, finalmente, calcula-se a média de $mAP@t$ ao longo dos dez limiares, tal como formalizado na **Equação 5**. Esta construção penaliza localizações “aproximadas” e torna a métrica mais robusta quando se pretende distinguir entre modelos que acertam na presença do objeto, mas diferem na precisão espacial.

$$mAP_{50-95} = \frac{1}{|T|} \sum_{\tau \in T} mAP@_{\tau}$$

$$T = \{0.50, 0.55, \dots, 0.95\}$$

Equação 5 - Fórmula para o cálculo do mAP_{50-95}

Na **Equação 5**, T representa o conjunto de limiares de sobreposição (IoU) considerados na avaliação, e $|T|$ a sua cardinalidade (neste caso, 10 valores entre 0.50 e 0.95, em incrementos de 0.05). O símbolo τ denota cada um desses limiares. Assim, para cada $\tau \in T$ calcula-se o desempenho agregado $mAP@_{\tau}$, por fim, obtém-se $mAP@0.50:0.95$ como a média desses resultados ao longo de todos os limiares. Esta formulação torna a avaliação mais exigente do que $mAP@0.50$: um modelo pode “acertar” na presença e na classe do módulo com uma localização aproximada e ainda assim obter um bom mAP_{50} , mas será penalizado quando se exige maior precisão geométrica, refletindo-se numa descida do mAP_{50-95} . No contexto deste trabalho, esta distinção é relevante porque permite separar melhorias reais de comportamento (detecções mais estáveis e melhor localização do módulo) de ganhos apenas superficiais em regimes permissivos.

Quando se avaliam variantes de segmentação (por exemplo, YOLOv11-seg), o protocolo mantém-se, mas a noção de sobreposição deixa de ser apenas geométrica ao nível da *BBox* e passa a ser calculada também ao nível da máscara prevista. Na prática, isto traduz-se em métricas *mask* (*mask precision*, *mask recall*, *mask mAP₅₀* e *mask mAP₅₀₋₉₅*) reportadas em paralelo com as métricas *box*, permitindo perceber se o modelo melhora apenas a localização grosseira do módulo ou se melhora efetivamente a correspondência espacial da região segmentada. Esta distinção é particularmente relevante neste domínio, porque reflexos especulares e sombras duras podem gerar caixas plausíveis, mas máscaras erradas; nesse cenário, as métricas de máscara funcionam como teste mais exigente à qualidade espacial da predição.

Para assegurar comparabilidade com abordagens fora da família YOLO (como *pipelines* baseados em *transformers* com DINOv2), a avaliação mantém o mesmo

protocolo de medição descrito acima: as predições são convertidas para o mesmo formato de detecção/segmentação e medidas com o mesmo conjunto de limiares de *IoU*, reportando *mAP50* e *mAP50–95* (global e por classe). Desta forma, qualquer diferença observada entre modelos é atribuída ao comportamento do detetor e não a uma mudança no protocolo de avaliação. Complementarmente, e porque o objetivo é operacional, os resultados foram acompanhados por indicadores de custo (tempo de inferência/*throughput* e dimensão do modelo), de modo a explicitar o compromisso entre qualidade e viabilidade de implementação em cenário real de inspeção.

4.3 Cenário de Implementação

O cenário experimental foi executado em *Kaggle Notebooks* [31], utilizando um ambiente com duas *GPUs* NVIDIA Tesla T4, cada uma com 16 GB de VRAM, o que permitiu treinar modelos de maior capacidade e conduzir afinações de forma eficiente. O *dataset* foi disponibilizado em /kaggle/input/... em modo apenas-leitura, enquanto os artefactos resultantes dos treinos — incluindo pesos, *logs*, gráficos e métricas — foram gerados em /kaggle/working/..., garantindo rastreabilidade por execução. Uma consequência prática desta configuração foi a impossibilidade de gravar cache no diretório do *dataset*, surgindo avisos do tipo “cache directory ... is not writable”. Este aspeto não alterou os dados nem o protocolo de avaliação, limitando-se a condicionar a persistência de cache.

Os modelos YOLO foram treinados com a *framework Ultralytics*, em Python 3.12.12 e *PyTorch* 2.8.0 com CUDA (*Compute Unified Device Architecture*) 12.6, com execução em *GPU* e utilização de *Automatic Mixed Precision* (AMP). A configuração do treino, bem como o contexto do ambiente (versões, *GPU* e parâmetros relevantes), ficou registada nos logs, permitindo reconstituir cada experiência. Para assegurar comparabilidade entre abordagens, foi utilizado o mesmo *dataset* para todos os modelos, com os *splits* fixos definidos em data.yaml, e foi mantida a resolução de entrada dos YOLO em *imgsz=640*, evitando que diferenças nos resultados resultassem de alterações nos dados ou nas condições base de treino.

Na tarefa de detecção, foram treinados YOLOv8x e YOLOv11x a partir de pesos pré-treinados. O YOLOv8x foi treinado com *epochs=100*, *batch=16*, *patience=25*, *workers=4* e utilização de duas *GPUs* (*device=[0,1]*), tendo sido identificado como pv_yolo8x_640. Para aumentar robustez face a variações típicas de inspeção visual, foi aplicado um conjunto consistente de *augmentations*, incluindo *scale=0.6*, *degrees=10*, *shear=5*, *flipud=0.2*, *mixup=0.1*, *erasing=0.1*, *mosaic=1.0* e *close_mosaic=15*. O YOLOv11x seguiu a mesma filosofia experimental — pesos pré-treinados, dados e *splits* constantes e resolução fixa — com registo automático dos parâmetros e do comportamento de treino assim como evolução de métricas como precisão, *recall* e *mAP* ao longo das épocas.

Para a tarefa de segmentação, foi utilizado o YOLOv11-seg (YOLOv11-seg.pt), mantendo o mesmo *dataset* e resolução (imgsz=640), mas ajustando a configuração para acomodar o custo adicional da previsão de máscaras. O treino foi conduzido com *epochs*=50 e *batch*=8, preservando *patience*=25, execução em GPU e *augmentations* compatíveis (*degrees*=10, *shear*=5, *flipud*=0.2, *mixup*=0.1), de modo a manter coerência metodológica com os modelos de detecção.

O DINOv2 foi treinado num *notebook* dedicado, com um *pipeline* próprio, refletindo as exigências do modelo. A configuração incluiu *IMG_SIZE*=518, *EPOCHS*=30, *BATCH_SIZE*=6 e *LR*=1e-4, com otimização via *AdamW* e função de perda *CrossEntropyLoss*, sendo gerados artefactos de avaliação e resultados (pesos, registos e ficheiros de análise) para suportar a comparação apresentada no subcapítulo seguinte. No conjunto, este cenário foi definido para tornar a avaliação reprodutível e comparável: dados e *splits* fixos, condições-base explicitadas, e parâmetros de treino documentados, permitindo que a seleção final do modelo resultasse de evidência experimental e não de preferências.

4.4 Resultados e Discussão

Neste subcapítulo apresentaram-se e discutiram-se os resultados obtidos na avaliação dos modelos considerados, com base no conjunto de validação definido no ficheiro *data.yaml* e no *dataset* descrito no subcapítulo 4.1, mantendo-se constantes as condições experimentais estabelecidas no subcapítulo 4.3. A análise seguiu as métricas introduzidas no subcapítulo 4.2, privilegiando *mAP@0.50* e *mAP@0.50:0.95*, bem como precisão e *recall*, por traduzirem de forma direta o compromisso entre deteções corretas e erros com impacto operacional. Os resultados foram organizados em duas dimensões complementares: uma componente quantitativa, apresentada em tabelas de síntese por modelo, e uma componente qualitativa, suportada por curvas de treino e exemplos de inferência, de forma a caracterizar não apenas o desempenho global, mas também a estabilidade do treino e os padrões típicos de acerto e falha nas três classes consideradas.

Seguidamente apresentaram-se os resultados quantitativos por modelo, com o objetivo de estabelecer uma base objetiva para a comparação antes da análise qualitativa. Nos modelos YOLO, os valores reportados corresponderam à melhor época no conjunto de validação, selecionada com base no *mAP@0.50:0.95*, por ser uma métrica mais exigente e representativa do desempenho global. Considerando que deteção e segmentação responderam a tarefas diferentes e produziram saídas distintas (caixas vs máscaras), os resultados foram organizados em tabelas separadas, de forma a evitar comparações artificiais: a **Tabela 5** resumiu os modelos de deteção, a **Tabela 6** apresentou o desempenho do modelo YOLO de segmentação e a **Tabela 7** reuniu os resultados do DINOv2, incluindo métricas globais e, quando aplicável, indicadores por classe.

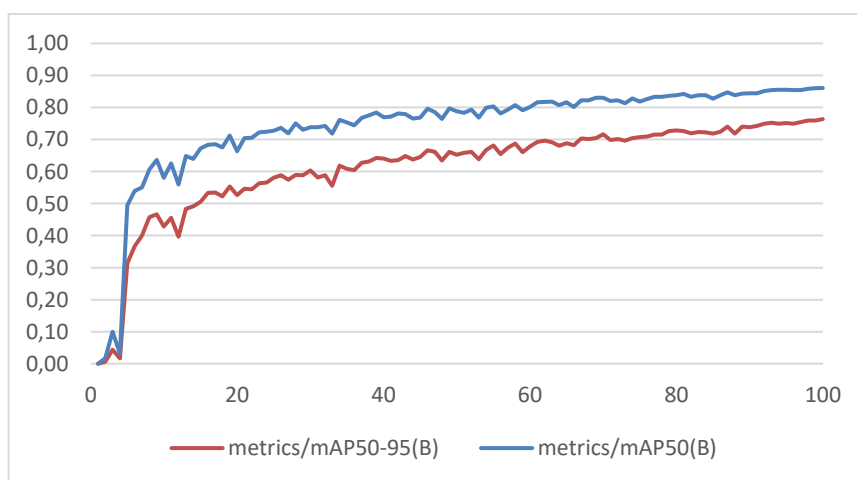
Tabela 5 - Desempenho global em validação (detecção)

Modelo	Melhor época	Precisao (P)	Recall (R)	$mAP@0.50$	$map@0.50:0.95$
YOLOv8x	100	0,861	0,781	0,861	0,764
YOLOv11x	100	0,85	0,768	0,853	0,752

A **Tabela 5** mostrou que ambos os modelos de detecção apresentaram desempenhos próximos, atingindo o melhor resultado na época 100, o que sugeriu convergência estável no conjunto de validação. Ainda assim, o YOLOv8x obteve valores ligeiramente superiores em todas as métricas, destacando-se no $mAP@0.50:0.95$ (0,764 face a 0,752), indicador mais exigente por avaliar a qualidade das detecções sob diferentes limiares de IoU. Esta vantagem, embora moderada, foi consistente e apontou para um melhor compromisso global entre precisão e capacidade de generalização. De seguida, analisou-se o desempenho na tarefa de segmentação, onde a avaliação passou a incidir sobre máscaras, com métricas específicas para esse tipo de saída.

Importa ainda notar que, após o fecho do *mosaic* nas últimas épocas, verificou-se melhoria adicional em $mAP@0.50:0.95$, indicando que o treino continuou a refinar o modelo no regime final.

Para complementar a comparação baseada nos valores finais da **Tabela 5**, as Figuras **Figura 17** e **Figura 18** apresentaram a evolução do desempenho em validação ao longo das épocas para o YOLOv8x e o YOLOv11x, respetivamente, reportando $mAP@0.50$ e $mAP@0.50:0.95$. A análise destas curvas permitiu verificar se a vantagem observada no final do treino resulta de um comportamento consistente e estável, bem como observar o impacto do regime final de treino, após o fecho do *mosaic*.

**Figura 17** - Evolução do mAP em validação (YOLOv8x)

Em ambos os casos verificou-se uma subida acentuada das métricas nas primeiras épocas, seguida de um crescimento progressivamente mais lento, indicando convergência e ganhos marginais na fase final. O $mAP@0.50$ estabilizou mais cedo, enquanto o $mAP@0.50:0.95$ continuou a melhorar de forma gradual, refletindo afinação da qualidade da localização sob critérios de IoU mais exigentes. No período final, as curvas mantiveram-se estáveis e sem oscilações bruscas, o que sugeriu ausência de instabilidade no treino e reforçou que as diferenças reportadas na Tabela 5 foram sustentadas por uma evolução consistente e não por uma época isolada.

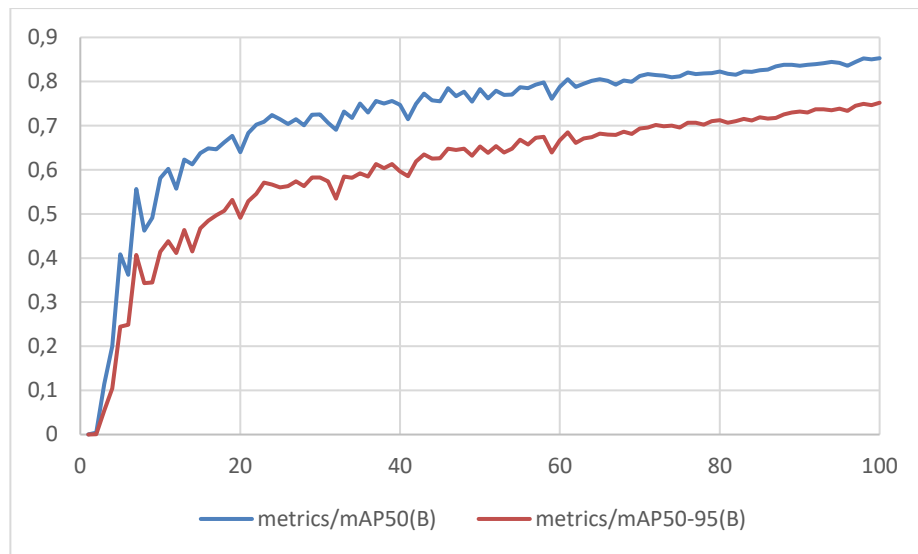


Figura 18 - Evolução do mAP em validação (YOLOv11x)

A comparação entre as Figuras, **Figura 17** e **Figura 18** mostrou comportamentos de convergência semelhantes: após um crescimento rápido nas primeiras épocas, as curvas entraram numa fase de estabilização, com ganhos marginais sobretudo no $mAP@0.50:0.95$, o que foi compatível com uma afinação progressiva da qualidade de localização. A ausência de oscilações acentuadas na fase final sugere um treino estável no conjunto de validação, reforçando a leitura de que a vantagem do YOLOv8x na Tabela 5 resulta de um padrão consistente ao longo do treino. Com esta base consolidada para deteção por *BBoxes*, passou-se à avaliação do modelo orientado à segmentação, onde o critério deixou de ser apenas “detetar e localizar” e passou a incluir a qualidade espacial das máscaras previstas.

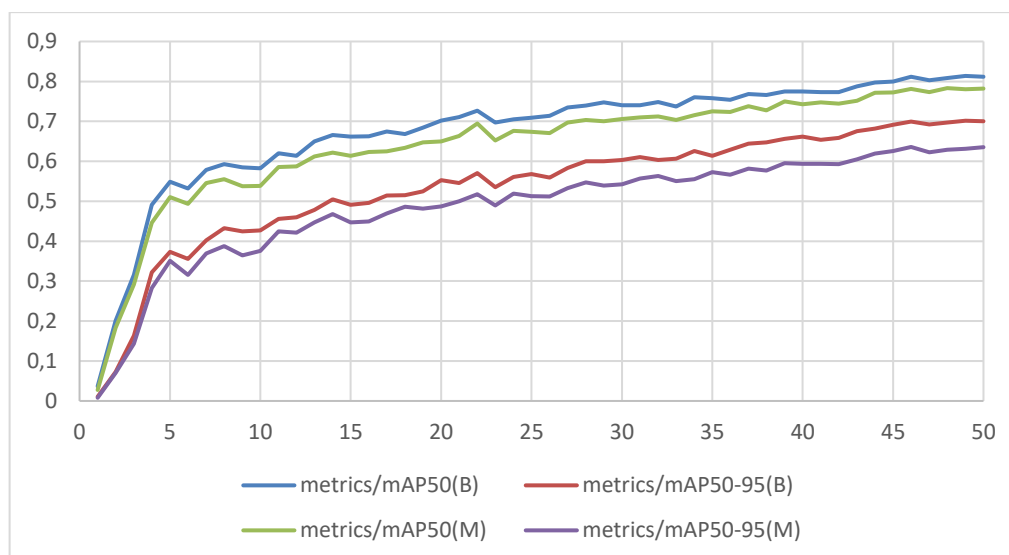
De seguida, a **Tabela 6** apresenta os resultados do modelo YOLOv11-seg, distinguindo o desempenho baseado em *BBoxes* e em máscaras, de forma a caracterizar o impacto da segmentação na qualidade das previsões.

Tabela 6 - Desempenho global em validação (segmentação) — YOLOv11-seg

Modelo	Saída	Melhor época	Precisão (P)	Recall (R)	$mAP@0.50$	$map@0.50:0.95$
YOLOv11-seg	<i>Box</i>	46	0,825	0,736	0,812	0,7
YOLOv11-seg	<i>Mask</i>	46	0,809	0,722	0,782	0,636

A **Tabela 6** evidenciou a diferença entre avaliar o modelo apenas pela localização (*BBoxes*) e avaliar a segmentação propriamente dita. Observou-se que as métricas baseadas em caixas se mantiveram relativamente elevadas ($mAP@0.50:0.95$ de 0,700), enquanto as métricas de máscara desceram ($mAP@0.50:0.95$ de 0,636), refletindo o aumento de dificuldade inerente à previsão precisa de contornos e áreas. Este comportamento era esperado: a segmentação exigia não só detetar a presença da classe, mas delimitar a sua forma com maior rigor espacial, o que tendia a penalizar métricas mais exigentes.

Para além da comparação agregada apresentada na **Tabela 6**, importou observar como estas métricas evoluíram ao longo do treino. A Figura 19 apresentou a evolução do desempenho em validação para o YOLOv11-seg, distinguindo as métricas calculadas com base em *BBoxes* das métricas calculadas sobre máscaras, permitindo analisar a convergência do modelo e perceber se a diferença entre *box* e *mask* se manteve estável ou se se acentou na fase final do treino.

**Figura 19** - Evolução do mAP (box e mask) em validação (YOLOv11-seg)

A Figura 19 confirmou que a segmentação impõe uma exigência adicional face à deteção: embora as métricas associadas a caixas estabilizassem em valores elevados, as métricas de máscara mantiveram-se sistematicamente abaixo, com

uma margem que não desapareceu na fase final do treino. Este padrão indicou que o modelo conseguiu identificar e localizar as classes de forma consistente, mas encontrou maior dificuldade em reproduzir contornos e áreas com a mesma precisão, o que é particularmente relevante quando existem transições pouco definidas entre “limpo” e “poeira” ou quando a sujidade é pequena e difusa, como podia acontecer em *bird-drop*. Esta evidência motiva a comparação com uma abordagem alternativa de segmentação baseada em *transformers*, de forma a avaliar se a representação do *DINOv2* oferece vantagens na delimitação espacial.

Para complementar esta análise com uma abordagem de segmentação alternativa, a **Tabela 7** apresenta os resultados do *DINOv2*, incluindo métricas globais e valores de IoU por classe, permitindo observar de forma mais fina onde se concentram os acertos e as fragilidades do modelo.

Tabela 7 - Desempenho em validação (segmentação) — *DINOv2*

Modelo	Melhor época	<i>Pixel Accuracy</i> (val)	<i>mIoU</i> (val)	IoU <i>background</i>	IoU <i>clean</i>	IoU <i>dust</i>	IoU <i>bird-drop</i>
<i>DINOv2</i>	28	0,905	0,816	0,842	0,722	0,867	0,832

No caso do *DINOv2*, observou-se um desempenho global competitivo em segmentação, com *pixel accuracy* de 0,905 e *mIoU* de 0,816 na melhor época (28). A análise por classe revelou um comportamento assimétrico: as classes *dust-solar-panel* e *bird-drop* apresentam IoU elevados (0,867 e 0,832), enquanto a classe *clean-solar-panel* surgiu com um valor inferior (0,722). Este padrão sugeriu que o modelo é particularmente eficaz a isolar regiões localizadas de sujidade, mas enfrenta maior dificuldade em delimitar de forma consistente áreas extensas e visualmente homogêneas, mais sensíveis a variações de iluminação e reflexos. Em termos operacionais, isto indicou que o *DINOv2* é forte para mapear sujidade, mas a interpretação da classe *clean* deve ser feita com cautela quando o objetivo é quantificação fina da área limpa.

Para além dos valores reportados na **Tabela 7**, importou analisar a dinâmica de aprendizagem do *DINOv2*, de forma a aferir a estabilidade do treino e a capacidade de generalização. A **Figura 20** apresentou a evolução da *loss* e do *mIoU* em treino e validação ao longo das épocas, permitindo verificar se o ganho observado no treino se traduz numa melhoria sustentada no conjunto de validação, ou se existe divergência entre ambos.

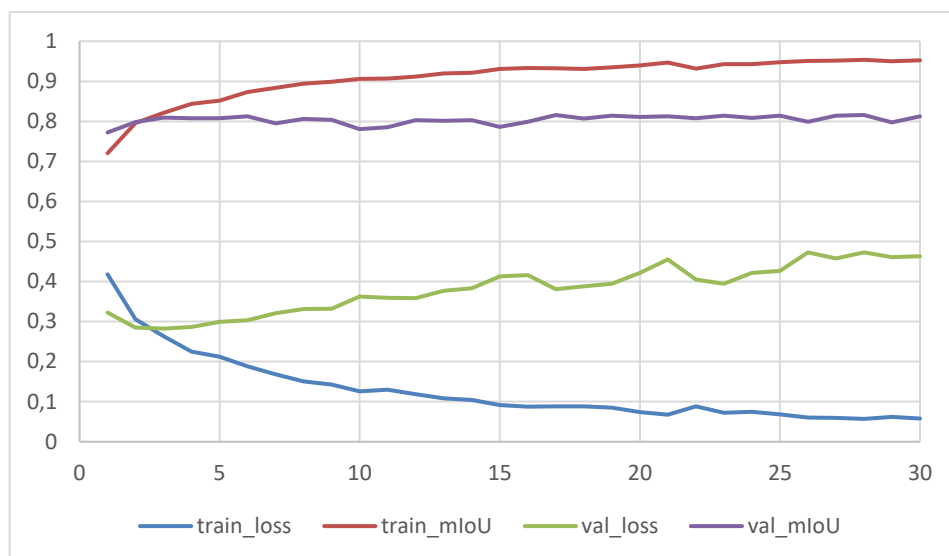


Figura 20 - Evolução das curvas de aprendizagem (*DINOv2*)

A **Figura 20** evidenciou uma redução contínua da *loss* e um aumento do *mIoU* em treino, indicando aprendizagem consistente. Em validação, o *mIoU* sobe rapidamente nas primeiras épocas e tende a estabilizar num patamar, enquanto a *loss* de validação apresenta uma tendência de crescimento gradual. Este comportamento é compatível com sobreajuste nas épocas finais: o modelo continua a otimizar o desempenho em treino, mas os ganhos em validação tornam-se marginais, reforçando a opção de selecionar a época ótima com base no máximo de *mIoU* em validação, tal como refletido na **Tabela 7**.

Para detalhar onde se concentram as fragilidades observadas na **Tabela 7**, a **Figura 21** apresenta a evolução do *IoU* por classe em validação ao longo do treino. Esta leitura é particularmente relevante para compreender se a classe *clean-solar-panel* é sistematicamente mais difícil ou se a diferença resulta de flutuações pontuais entre épocas.

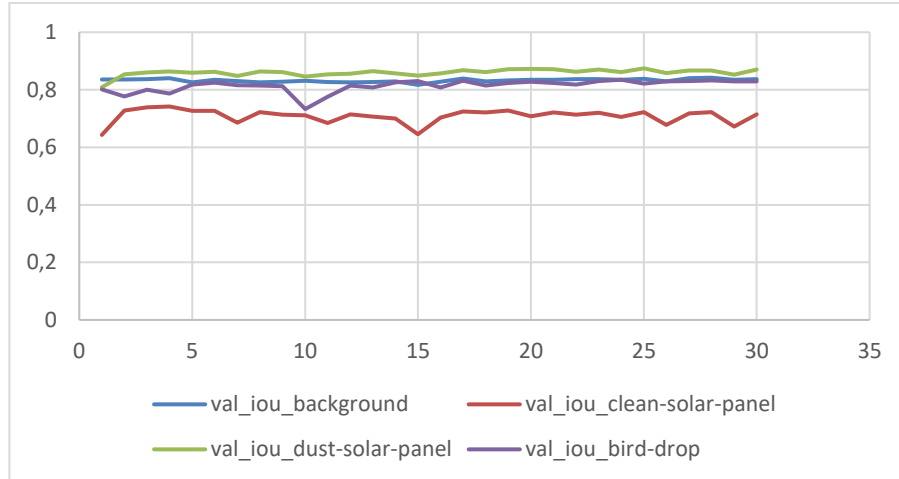


Figura 21 - IoU por classe em validação (DINOv2)

A **Figura 21** confirmou uma assimetria persistente entre classes, com desempenho mais elevado e estável nas classes associadas a sujidade (*dust-solar-panel* e *bird-drop*) e maior variabilidade na classe *clean-solar-panel*. Esta diferença é consistente com a natureza visual das classes: a delimitação de regiões limpas tende a depender de áreas extensas e homogêneas, onde pequenas alterações de iluminação ou reflexos podem deslocar a fronteira de segmentação. Assim, embora o *DINOv2* apresente bom potencial para mapear sujidade, a utilização da classe *clean* para estimativas finas de área deve ser interpretada com prudência, sendo recomendável validar qualitativamente o comportamento do modelo em cenários com reflexos e sombras.

Estabelecida a comparação quantitativa, segue-se a análise qualitativa, centrada na dinâmica do treino e em exemplos de inferência. As curvas de treino permitem observar a convergência, a estabilidade das métricas e eventuais sinais de sobreajuste, enquanto as predições em imagens reais ajudam a identificar padrões típicos de erro (por exemplo, confusões por sombras/reflexos ou dificuldade em detetar *bird-drop* de pequena dimensão), que não são totalmente capturados por métricas agregadas.

Para tornar esta análise qualitativa verificável, recorre-se a um conjunto de exemplos de inferência selecionados a partir do mesmo conjunto de validação, cobrindo as três classes e incluindo situações típicas do domínio PV, como variações de iluminação, reflexos, sombras e ocorrências de *bird-drop* com dimensão reduzida. Em todos os casos, as previsões são apresentadas com o mesmo critério de visualização e sob condições equivalentes, de modo a que as diferenças observadas sejam atribuíveis ao comportamento do modelo e não a alterações no procedimento.

As figuras seguintes organizaram-se por tarefa e por classe: primeiro foram mostrados exemplos de deteção (*YOLOv8x* e *YOLOv11x*), depois exemplos de segmentação (*YOLOv11-seg* e *DINOv2*). Esta organização permitiu ler os

resultados com coerência: nos modelos de detecção avaliou-se sobretudo a consistência da localização e a propensão para falsos positivos/negativos; nos modelos de segmentação, para além da detecção, observou-se a qualidade do contorno e a estabilidade da área segmentada, aspetos diretamente relacionados com as métricas de máscara e com a utilidade operacional da solução.

Neste enquadramento, as Figuras, **Figura 22** a **Figura 33** apresentam exemplos de inferência no conjunto de validação, organizados por modelo e mantendo a mesma sequência de classes em cada conjunto. Para cada modelo foram apresentados três exemplos, correspondentes às classes *clean-solar-panel*, *bird-drop* e *dust-solar-panel*, permitindo observar, de forma consistente, a propensão para falsos positivos e falsos negativos na detecção e, quando aplicável, a qualidade espacial das máscaras na segmentação. Em particular, as Figuras, **Figura 22**, **Figura 23** e **Figura 24** correspondem ao *YOLOv8x*, as Figuras, **Figura 25**, **Figura 26** e **Figura 27** ao *YOLOv11x*, as Figuras, **Figura 28**, **Figura 29** e **Figura 30** ao *YOLOv11x-seg* e as Figuras, **Figura 31**, **Figura 32** e **Figura 33** ao *DINOv2*.



Figura 22 - YOLOv8x inferência clean-solar-panel

Na **Figura 22** observou-se que o *YOLOv8x* identificou de forma consistente a classe *clean-solar-panel* em diferentes contextos visuais, incluindo painéis isolados e cenários com múltiplos painéis no campo de visão. As confianças associadas variaram entre valores elevados (por exemplo, 0,88 e 0,92) e valores mais moderados (por exemplo, 0,50–0,69), sugerindo maior incerteza em situações com perspetiva mais complexa, maior distância ou possíveis efeitos de brilho/reflexo na superfície. Notou-se ainda que, em cenários com várias estruturas, o modelo

tendeu a produzir múltiplas caixas por imagem, o que é coerente com a natureza do problema, mas podia introduzir redundância na contagem caso não exista um pós-processamento cuidadoso.



Figura 23 - YOLOv8x inferência *bird-drop*

De seguida, a **Figura 23** apresentou exemplos de inferência para as classes *bird-drop*, com o objetivo de avaliar o comportamento do YOLOv8x em padrões de sujidade localizada e de pequena dimensão, bem como a sua robustez perante variações de textura, perspectiva e contraste.

A **Figura 23** mostrou que o YOLOv8x conseguiu identificar ocorrências evidentes de *bird-drop* com elevada confiança (por exemplo, 0,93), o que é consistente com a capacidade do modelo em detetar padrões localizados e de contraste acentuado. Observou-se também que, em imagens onde coexistem *dust-solar-panel* e *bird-drop*, o modelo foi capaz de distinguir ambas as classes em vários casos, atribuindo caixas separadas a cada fenómeno. No entanto, esta distinção não é totalmente estável: em algumas situações, sobretudo quando o *bird-drop* surge sobre fundo já marcado por poeira, o modelo tendeu a confundir padrões e a classificar ocorrências de *bird-drop* como *dust-solar-panel* (ou vice-versa), refletindo a

sobreposição visual entre texturas e manchas. Adicionalmente, em cenários com perspectiva mais complexa e maior distância, a confiança associada a *dust-solar-panel* podia descer (por exemplo, 0,43), sugerindo que a detecção de sujidade difusa é mais sensível ao enquadramento e ao contraste do que a detecção de ocorrências concentradas.

Por fim, a **Figura 24** apresentou exemplos de inferência do YOLOv8x na classe *dust-solar-panel*, com o objetivo de avaliar a consistência da detecção de sujidade difusa e a sua sensibilidade a transições graduais entre áreas limpas e áreas com acumulação de poeira.



Figura 24 - YOLOv8x inferência *dust-solar-panel*

A **Figura 24** evidenciou que o YOLOv8x identificou *dust-solar-panel* de forma consistente em diferentes padrões de acumulação, incluindo manchas localizadas e sujidade mais distribuída, atribuindo confianças elevadas na maioria dos casos (por exemplo, 0,81–0,96). Observou-se ainda um exemplo em que o modelo atribui simultaneamente *clean-solar-panel* e *dust-solar-panel* na mesma imagem, o que foi coerente com a presença de regiões com estados distintos — uma zona efetivamente limpa e outra com acumulação de poeira — e, portanto, representou uma separação desejável do ponto de vista operacional. Ainda assim, quando a transição entre limpo e poeira é gradual e o contraste é reduzido, a delimitação destas regiões podia tornar-se mais incerta, refletindo a dificuldade intrínseca da classe *dust-solar-panel*, cuja textura pode aproximar-se de padrões induzidos por

iluminação e reflexos. Este aspeto ajudou a explicar porque a deteção de sujidade difusa tendeu a ser mais sensível ao enquadramento e às condições de luz do que classes mais concentradas, como *bird-drop*.

Concluída a análise qualitativa do YOLOv8x (Figuras 22 a 24), as Figuras, **Figura 25**, **Figura 26** e **Figura 27** apresentaram o mesmo conjunto de classes para o YOLOv11x, permitindo comparar diretamente padrões de erro e consistência das previsões entre os dois modelos de deteção.



Figura 25 - YOLOv11x inferência *clean-solar-panel*

A **Figura 25** mostrou que o YOLOv11x identifica a classe *clean-solar-panel* em diferentes contextos, desde painéis isolados até cenários com múltiplas estruturas, com níveis de confiança globalmente moderados. Em imagens com perspetiva mais complexa e maior distância, observa-se maior variabilidade na confiança atribuída (por exemplo, valores a descer para 0,32), o que sugere sensibilidade ao tamanho aparente do objeto e ao detalhe disponível. Verifica-se ainda um exemplo em que o modelo atribui *dust-solar-panel* a uma superfície aparentemente limpa (0,79), indiciando que padrões de iluminação, reflexos ou textura podem aproximar-se visualmente da classe *dust* e induzir confusão entre estados “limpo” e “poeira” quando a evidência é subtil.

De seguida, a **Figura 26** focou a classe *bird-drop*, permitindo analisar o comportamento do YOLOv11x em ocorrências pequenas e de maior dificuldade.



Figura 26 - YOLOv11x inferência *bird-drop*

A **Figura 26** evidenciou que o YOLOv11x conseguiu detetar ocorrências evidentes de *bird-drop* com elevada confiança (por exemplo, 0,91), confirmando boa sensibilidade a padrões localizados e de contraste acentuado. Em cenários onde o painel apresenta simultaneamente sujidade difusa (*dust-solar-panel*), observou-se que o modelo tendeu a atribuir elevada confiança à classe *dust* (por exemplo, 0,96) e a reduzir a confiança associada ao *bird-drop* quando este surge sobre fundo já marcado por poeira (por exemplo, 0,57), o que sugeriu que a textura da poeira “mascara” parcialmente o sinal do *bird-drop* e torna a distinção menos segura. Esta dificuldade torna-se ainda mais visível em casos com maior distância ou menor detalhe, onde ambas as classes aparecem com confianças mais baixas (por exemplo, *dust-solar-panel* 0,48 e *bird-drop* 0,40), o que refletiu maior incerteza global do modelo. Por fim, verificou-se também um exemplo em que *dust-solar-panel* foi identificado com elevada confiança (por exemplo, 0,92), mas o *bird-drop* presente não foi detetado, evidenciando a possibilidade de falsos negativos para

bird-drop quando a ocorrência é pequena ou visualmente pouco destacada face ao padrão dominante de poeira.

Por fim, a **Figura 27** apresenta exemplos de inferência do *YOLOv11x* na classe *dust-solar-panel*, permitindo avaliar a consistência do modelo na detecção de poeira difusa e a sua robustez quando a acumulação surge com transições graduais entre regiões limpas e regiões com sujidade.



Figura 27 - YOLOv11x inferência *dust-solar-panel*

A **Figura 27** mostrou que o *YOLOv11x* conseguiu identificar *dust-solar-panel* com elevada confiança quando a acumulação é visualmente evidente (por exemplo, 0,95 e 0,93), o que indicou boa capacidade para reconhecer padrões de poeira com contraste suficiente. No entanto, em cenários com transição gradual entre áreas limpas e áreas com poeira, observa-se maior incerteza e coexistência de classes, com confianças mais baixas (por exemplo, *clean-solar-panel* 0,35 e *dust-solar-panel* 0,32), refletindo a dificuldade em estabelecer uma fronteira nítida quando o sinal visual é subtil. Adicionalmente, verifica-se um exemplo em que o modelo produz múltiplas caixas para *dust-solar-panel* com confianças distintas (por exemplo, 0,26 e 0,78), sugerindo fragmentação da deteção em regiões parcialmente ambíguas. No conjunto das Figuras 25 a 27, o *YOLOv11x* revela comportamento semelhante ao observado no *YOLOv8x*: desempenho consistente

em casos claros, mas maior sensibilidade a condições de baixo contraste e a transições progressivas entre *clean* e *dust*.

De seguida, a **Figura 28** apresenta exemplos de inferência do *YOLOv11x-seg* para a classe *clean-solar-panel*. Ao contrário dos modelos de deteção, a análise centra-se aqui na qualidade espacial da segmentação, isto é, na forma como a máscara cobre o painel e respeita os seus limites, sobretudo em cenários com perspectiva complexa, múltiplos painéis e variações de iluminação.



Figura 28 - YOLOv11-seg inferência *clean-solar-panel*

A **Figura 28** mostra que o *YOLOv11x-seg* consegue segmentar a classe *clean-solar-panel* de forma globalmente coerente, produzindo máscaras que acompanham a geometria do painel e permitem uma leitura mais rica do que a simples *BBox*. Em cenários com painéis isolados e boa visibilidade, a máscara cobre a área relevante com consistência, o que é particularmente útil para tarefas de quantificação espacial. No entanto, quando existem múltiplos painéis à distância ou quando a cena apresenta reflexos e brilho acentuado, observa-se maior variabilidade na confiança e na estabilidade da segmentação, surgindo previsões com valores mais baixos (por exemplo, ~0,35) e maior sensibilidade a limites menos definidos. Importa ainda notar que, num dos exemplos, o modelo atribui *dust-solar-panel* com confiança elevada (0,753) numa imagem essencialmente limpa, o que evidencia uma confusão entre *clean* e *dust* induzida por condições de iluminação e padrões de brilho, particularmente relevantes em superfícies fotovoltaicas. Este comportamento é compatível com a exigência da segmentação: para além de detetar, o modelo tem de delimitar contornos com rigor, ficando mais exposto a

variações visuais que, por vezes, aproximam o aspeto do painel limpo do aspeto de poeira difusa.

A Figura 29 apresenta exemplos de inferência do *YOLOv11x-seg* em situações associadas a *bird-drop*, incluindo cenários em que este ocorre isoladamente e casos em que coexiste com sujidade difusa. O objetivo é avaliar não apenas a deteção da ocorrência, mas sobretudo a capacidade do modelo em segmentar a região de *bird-drop* com contorno plausível, evitando a sua absorção pela classe *dust-solar-panel*.

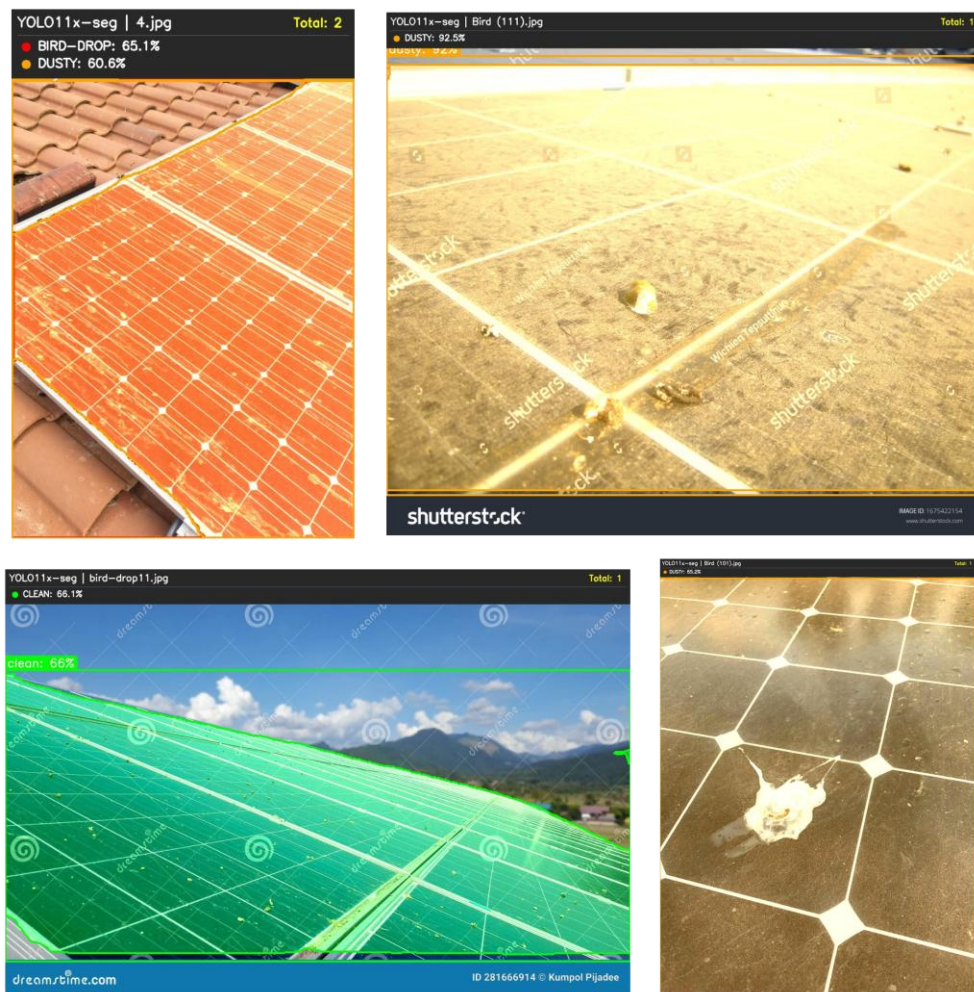


Figura 29 - YOLOv11-seg inferência *bird-drop*

A Figura 29 evidencia que o *YOLOv11x-seg* nem sempre consegue isolar o *bird-drop* como uma região distinta quando o fundo apresenta poeira ou variações de luminosidade. Em alguns exemplos, o modelo atribui *dust-solar-panel* com confiança elevada e expande a máscara para uma área muito extensa do painel, o que pode “encobrir” a ocorrência localizada e reduzir a utilidade da segmentação para quantificação fina. Observa-se também que, em determinados casos, o modelo alterna entre classes (por exemplo, *clean* versus *dust*) para explicar

padrões visuais complexos, sugerindo sensibilidade à textura e ao brilho. Em termos operacionais, este comportamento é crítico: mesmo quando a presença de sujidade é sinalizada, a incapacidade de separar *bird-drop* de dust pode comprometer a análise por tipo de sujidade e a priorização de intervenções. Esta leitura ajuda a contextualizar o decréscimo das métricas de máscara face às métricas por caixa observado na **Tabela 6**, reforçando que o custo da segmentação se manifesta sobretudo na delimitação de ocorrências pequenas e irregulares.

Por fim, a **Figura 30** apresenta exemplos de inferência do *YOLOv11x-seg* para a classe *dust-solar-panel*, com o objetivo de avaliar a estabilidade da segmentação em sujidade difusa e a forma como o modelo lida com transições graduais entre áreas limpas e áreas com acumulação de poeira.

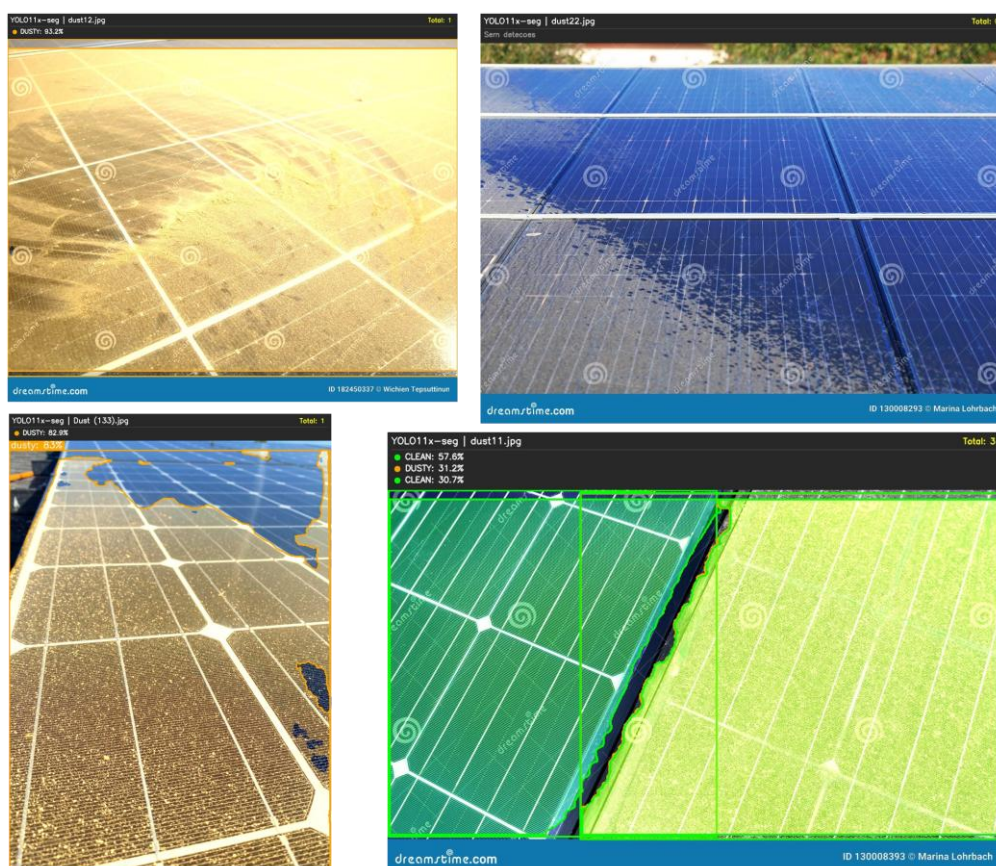


Figura 30 - YOLOv11-seg inferência *dust-solar-panel*

A **Figura 30** mostra um comportamento misto do *YOLOv11x-seg* na segmentação de *dust-solar-panel*. Em alguns casos, a máscara acompanha de forma plausível a região com poeira, sobretudo quando a acumulação é extensa e apresenta contraste suficiente, produzindo delimitações úteis para caracterização espacial. No entanto, observam-se também falhas relevantes: existe um exemplo sem deteções, apesar de a poeira ser visível, sugerindo um falso negativo em condições de baixo contraste ou iluminação adversa. Noutros casos, a segmentação surge

fragmentada ou parcialmente incoerente, com sobreposição de classes (*clean* e *dust*) e confianças moderadas, o que indica dificuldade em estabilizar a fronteira quando a transição entre estados é gradual. Em termos operacionais, este padrão é crítico porque afeta diretamente a quantificação de área suja e a consistência de decisões baseadas em percentagens; além disso, ajuda a explicar porque as métricas de máscara são inferiores às métricas por caixa na Tabela 5, já que pequenas imprecisões de contorno e inconsistências na cobertura penalizam fortemente a avaliação em segmentação.

Concluída a análise qualitativa do YOLOv11x-seg (Figuras 28 a 30), as Figuras, **Figura 31**, **Figura 32** e **Figura 33** apresentaram exemplos de inferência do DINOv2, permitindo comparar a qualidade das máscaras e os padrões de erro sob as mesmas classes.

As Figuras 31 a 33 apresentaram exemplos de inferência do DINOv2 no conjunto de validação, organizados pelas três classes. A **Figura 31** focou a classe *clean-solar-panel*, permitindo avaliar a estabilidade das máscaras em painéis com diferentes perspectivas, escalas e contextos, incluindo cenários com múltiplos módulos no campo de visão.

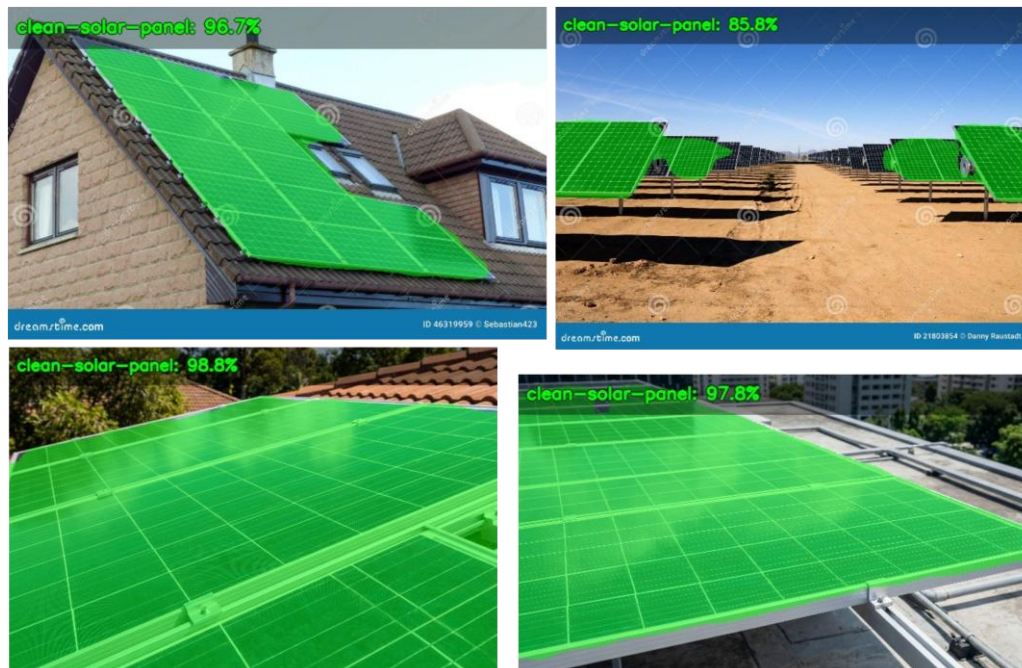


Figura 31 - DINOv2 inferência *clean-solar-panel*

A **Figura 31** mostrou que o DINOv2 segmentou *clean-solar-panel* de forma consistente, produzindo máscaras contínuas e bem alinhadas com a geometria do painel em diferentes condições. Observaram-se níveis de confiança elevados na maioria dos exemplos (por exemplo, 96,7%, 97,8% e 98,8%), o que sugeriu robustez quando o objetivo é delimitar áreas extensas e relativamente

homogêneas. Mesmo em cenários com múltiplos painéis e maior complexidade geométrica, o modelo manteve segmentações plausíveis, embora com ligeira redução de confiança num caso (85,8%), coerente com a presença de maior distância e menor detalhe. Em termos operacionais, este comportamento é particularmente relevante, pois indica potencial para estimar de forma estável a área limpa disponível, desde que a classe *clean* seja validada também em condições de forte brilho e sombras, onde a fronteira entre *clean* e *dust* tende a ser mais difícil.

De seguida, a **Figura 32** apresenta exemplos para *bird-drop*, permitindo avaliar a capacidade do *DINOv2* em segmentar ocorrências pequenas e de contorno irregular, que representam um dos cenários mais exigentes do problema.

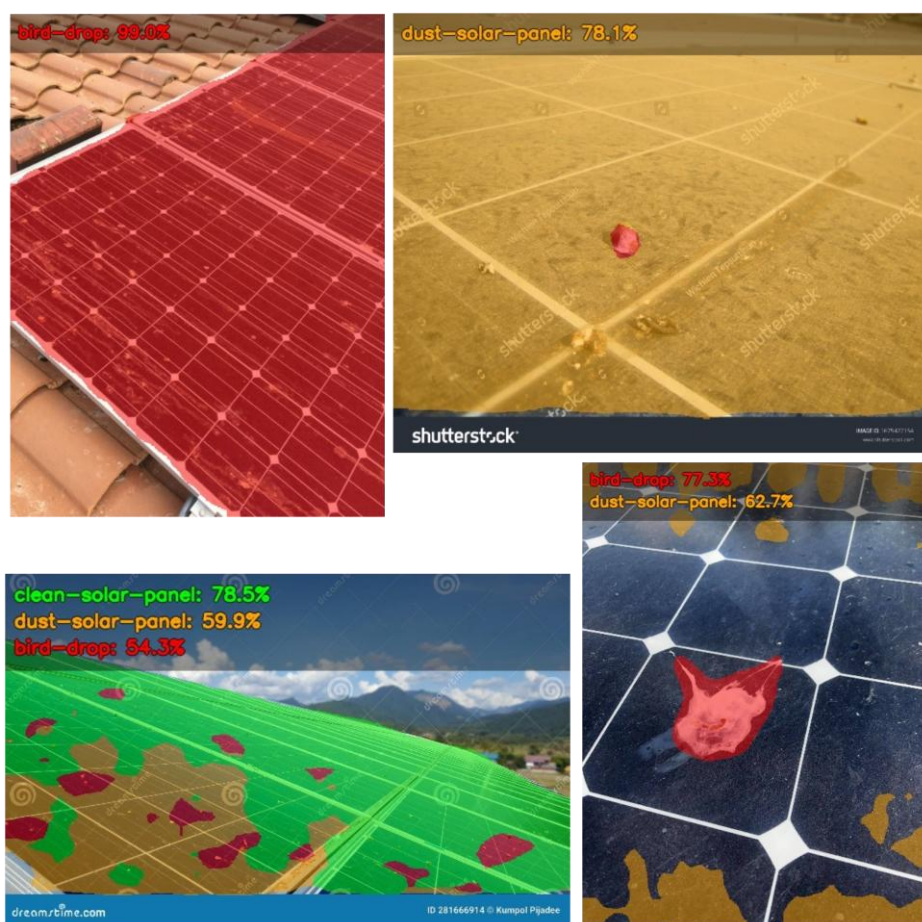


Figura 32 - *DINOv2* inferência *bird-drop*

A **Figura 32** evidenciou que o *DINOv2* conseguiu segmentar *bird-drop* com elevada confiança quando a ocorrência é visualmente dominante e apresenta contraste claro face ao fundo, como se observou no exemplo com *bird-drop* a 99,0%. Importa notar que, neste conjunto, existiam imagens anotadas com mais do que uma classe (por exemplo, *dust* + *bird-drop* ou *clean* + *dust*), pelo que a

presença simultânea de máscaras não é, por si só, indício de erro. Nos casos em que *bird-drop* coexiste com *dust-solar-panel*, o modelo tendeu a reconhecer ambas as classes, mas com confianças mais moderadas (por exemplo, *bird-drop* a 77,3% juntamente com *dust-solar-panel* a 62,7%), sugerindo que a separação entre “mancha localizada” e “sujeidade difusa” é mais exigente quando as fronteiras são pouco definidas. Observou-se ainda um exemplo em que o modelo ativa também *clean-solar-panel* em simultâneo, o que podia ser coerente com a anotação quando existem regiões limpas e sujas na mesma imagem, mas também podia refletir maior incerteza em cenários de grande área de painel e perspectiva complexa. Em termos operacionais, estes exemplos indicam que o *DINOv2* é particularmente eficaz a assinalar *bird-drop* quando este é evidente, mas que, em imagens com coexistência de classes, a utilidade prática depende da estabilidade espacial das máscaras: quanto mais o modelo expandir *dust* para regiões ambíguas, maior o risco de enviesar estimativas de área e dificultar a interpretação por tipo de sujeidade.

Por fim, a **Figura 33** apresentou exemplos de inferência do *DINOv2* para a classe *dust-solar-panel*. Estes casos permitiram avaliar a consistência da segmentação em sujeidade difusa, bem como a capacidade do modelo em lidar com transições graduais entre zonas limpas e zonas com poeira, situação frequente em painéis PV.

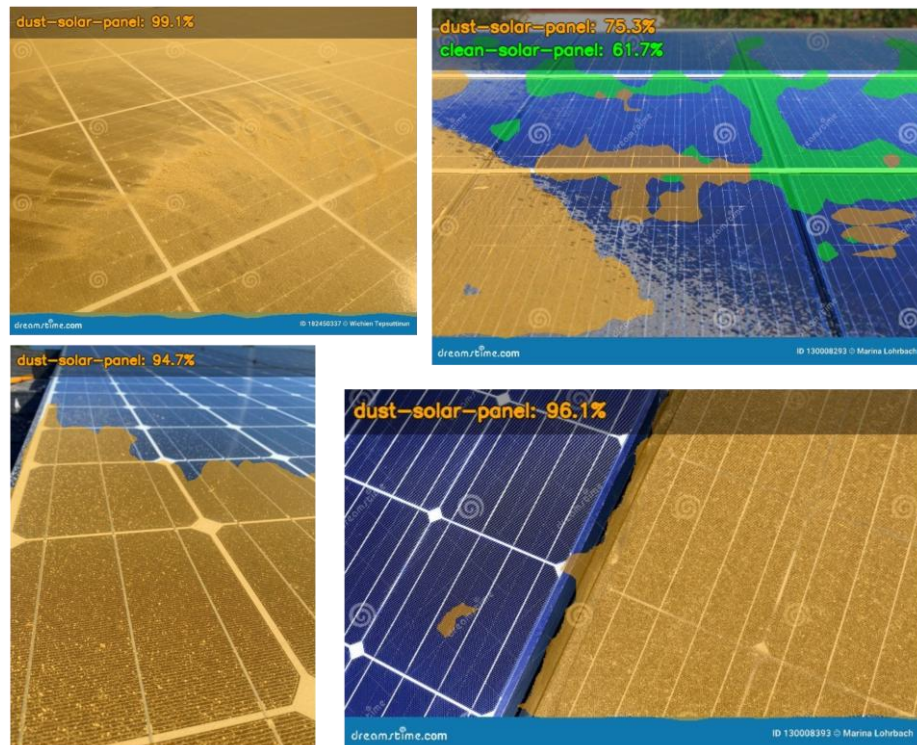


Figura 33 - *DINOv2* inferência *dust-solar-panel*

A **Figura 33** mostrou que o *DINOv2* identificou *dust-solar-panel* de forma robusta, atribuindo confianças elevadas na maioria dos exemplos (por exemplo, 94,7%,

96,1% e 99,1%), o que sugeriu boa sensibilidade a padrões de poeira extensa e a variações de textura associadas à acumulação. Observou-se ainda que, em cenários onde a imagem contém simultaneamente regiões limpas e regiões com poeira, o modelo ativou também a classe *clean-solar-panel* (por exemplo, *dust* a 75,3% e *clean* a 61,7%), o que foi coerente quando existe coexistência real de estados no mesmo painel. Ainda assim, nota-se que a fronteira entre classes tende a ser irregular e, em certos pontos, a máscara de *dust* expande-se para áreas ambíguas, refletindo a dificuldade em delimitar transições graduais sob iluminação variável e reflexos. Em termos operacionais, este comportamento é globalmente favorável para “mapear” sujidade e localizar regiões problemáticas, mas exige cautela quando o objetivo é quantificar com precisão a percentagem de área suja: pequenos erros de contorno podem alterar estimativas de área, sobretudo em imagens com gradientes suaves e brilho. No conjunto, as evidências qualitativas são consistentes com o desempenho quantitativo do *DINOv2*, que se revelou particularmente forte na segmentação de *dust* e competitivo em *bird-drop*, mantendo a classe *clean* como a mais sensível a variações de contexto visual.

Com base na análise quantitativa e qualitativa apresentada, foi possível retirar três conclusões centrais. Em primeiro lugar, na tarefa de deteção, os modelos *YOLOv8x* e *YOLOv11x* revelaram desempenhos próximos, com convergência estável e melhoria progressiva até ao final do treino, sendo que o *YOLOv8x* manteve uma vantagem consistente nas métricas mais exigentes, em particular no *mAP@0.50:0.95*. Em termos práticos, esta diferença, embora moderada, é relevante porque sugere maior robustez global na localização e classificação sob diferentes limiares de IoU, reduzindo a probabilidade de degradação quando a qualidade da caixa é avaliada com maior rigor.

Em segundo lugar, nos modelos de segmentação, confirmou-se que o salto de caixas para máscaras aumenta a dificuldade do problema: o *YOLOv11x-seg* mantém valores competitivos quando avaliado por *BBoxes*, mas apresenta quebra quando a métrica incide na qualidade das máscaras, refletindo instabilidade na delimitação fina e maior sensibilidade a transições graduais entre *clean* e *dust* e a ocorrências pequenas como *bird-drop*. Esta leitura é reforçada pelos exemplos de inferência, onde se observam casos de expansão indevida da máscara e confusões em regiões de brilho e baixo contraste.

Por fim, o *DINOv2* evidencia um desempenho forte em segmentação, particularmente nas classes associadas a sujidade (*dust-solar-panel* e *bird-drop*), com máscaras geralmente plausíveis e confiáveis em cenários onde a sujidade é visualmente marcada. Ainda assim, a classe *clean-solar-panel* permanece a mais sensível a variações de iluminação e reflexos, o que explica a assimetria verificada nas métricas por classe e reforça a necessidade de interpretar a estimativa de área limpa com prudência em cenários de brilho intenso. No conjunto, os resultados indicaram que o *DINOv2* é especialmente adequado quando o objetivo é mapear e quantificar sujidade, enquanto os *YOLO* se destacam pela consistência da deteção

e pela simplicidade operacional quando o foco é a sinalização rápida de ocorrências.

Estas evidências permitem enquadrar a seleção do modelo como uma decisão orientada ao objetivo do sistema. Se a prioridade for a detecção robusta e eficiente de ocorrências, com menor complexidade de pós-processamento, o *YOLOv8x* surge como a escolha mais sólida no presente protocolo. Se, pelo contrário, a prioridade for a estimativa espacial da sujidade e a produção de máscaras úteis para quantificação, o *DINOv2* apresenta vantagens claras, sobretudo para *dust-solar-panel* e *bird-drop*. A decisão final e a justificação da escolha do modelo a integrar no protótipo são consolidadas no subcapítulo seguinte, tendo em conta o compromisso entre desempenho, estabilidade qualitativa e requisitos operacionais definidos para o sistema.

5. Desafios e Oportunidades

Os resultados obtidos mostraram que a detecção e a segmentação de sujeira em painéis PV eram viáveis e podiam atingir um nível de desempenho relevante, sobretudo quando as ocorrências eram visualmente marcadas e o cenário apresentava condições favoráveis. No entanto, a passagem de um contexto experimental para um contexto operacional expôs rapidamente a fragilidade de qualquer sistema baseado em visão: a realidade não é “limpa”, nem repetível. Por isso, este capítulo não serve para relativizar o trabalho feito — serve para o tornar mais honesto, ao explicitar onde o método aguentou bem e onde começou a ceder.

Um primeiro desafio prendeu-se com a variabilidade do ambiente e com a forma como o painel “compete” com a própria sujeira em termos de padrões visuais. Alterações de iluminação, reflexos, sombras e textura repetitiva das células fotovoltaicas podiam introduzir ruído que confundiu o modelo, sobretudo em situações onde a diferença entre *clean-solar-panel* e *dust-solar-panel* era gradual e pouco contrastada. Nestes casos, a fronteira entre classes tornou-se menos objetiva e mais dependente do critério de anotação, o que explicou que, mesmo quando o modelo identificou corretamente a tendência geral, surjissem segmentações adicionais em zonas periféricas ou instabilidade na delimitação das áreas afetadas.

Um segundo desafio esteve relacionado com a coexistência de classes e com a ambiguidade inerente a ocorrências reais. Quando *bird-drop* surgiu em simultâneo com poeira difusa, o sistema tendeu a reconhecer ambas as classes, mas com confianças mais moderadas e maior sensibilidade a pequenas variações de textura. Este comportamento não foi apenas um “erro do modelo”; foi também um sinal de que o problema, tal como estava formulado, tinha uma componente contínua e contextual: a poeira podia atuar como ruído estrutural e tornava menos nítida a separação entre as regiões, especialmente nos limites das ocorrências. Acresceu ainda que a qualidade e consistência do *dataset* tiveram um peso determinante: imagens anotadas com mais do que uma classe, critérios pouco uniformes ou exemplos pouco representativos de cenários difíceis afetaram diretamente o treino e a interpretação das métricas.

Apesar destes desafios, existem oportunidades claras e realistas para evolução do trabalho. A mais imediata passa por reforçar e normalizar o conjunto de dados, com regras de anotação mais explícitas e uma cobertura mais equilibrada de condições críticas — diferentes intensidades de sujeira, ocorrências pequenas, reflexos fortes, sombras e combinações de classes. Em paralelo, a abordagem pode ser refinada de forma orientada ao objetivo: quando o foco é rapidez e detecção, modelos do tipo *YOLO* podem ser mais adequados; quando o foco é quantificação espacial e análise detalhada, soluções baseadas em *transformers* tendem a oferecer vantagens. Finalmente, para aproximar a investigação de uma aplicação prática, importa transformar detecções e máscaras em indicadores

acionáveis, como percentagem de área afetada e estimativa de severidade, permitindo que a saída do sistema deixe de ser apenas “um resultado visual” e passe a apoiar decisões concretas de limpeza e manutenção.

6. Conclusão e Trabalho Futuro

6.1 Conclusão

A sujidade em módulos PV — poeira, excrementos de aves e outros detritos — não é apenas um detalhe estético: reduz eficiência, pode contribuir para fenómenos de degradação e justifica a necessidade de inspeção sistemática com vista a uma limpeza mais direcionada. Neste contexto, este trabalho teve como finalidade propor e avaliar uma solução de inspeção automática baseada em visão computacional, com ambição clara de evoluir para um protótipo demonstrador capaz de apoiar a geração de alertas numa plataforma web e numa aplicação móvel. O que se apresenta neste relatório deve ser lido como uma prova de conceito estruturada, alinhada com os objetivos do projeto e com a lógica de desenvolvimento faseado típica de Projeto I.

Do ponto de vista técnico, a investigação realizada permitiu consolidar três pilares essenciais: um enquadramento do estado da arte (incluindo revisão sistemática), uma definição de requisitos e um ciclo experimental que compara abordagens com perfis distintos. Os resultados mostraram, de forma consistente, que a escolha do “melhor” modelo não é uma questão abstrata, mas uma decisão orientada ao objetivo do sistema: modelos da família *YOLO* revelaram-se fortes quando o foco é a sinalização rápida e estável de ocorrências, com uma operação mais simples; já soluções baseadas em *transformers*, como o *DINOv2*, evidenciaram vantagens quando o objetivo é mapear e quantificar sujidade através de máscaras, sobretudo em classes associadas a sujidade visualmente marcada, mantendo-se, contudo, maior sensibilidade na classe associada a “área limpa” sob variações de iluminação e reflexos. No conjunto, o trabalho confirmou a viabilidade técnica da abordagem e sustentou a integração futura do modelo selecionado num protótipo com valor operacional.

6.2 Trabalho Futuro

O passo seguinte mais relevante não é “trocar de modelo por trocar”, mas reforçar aquilo que mais condiciona a robustez: dados e critérios. Em trabalho futuro, a prioridade deve ser melhorar e expandir o *dataset*, com regras de anotação mais explícitas e consistentes, cobrindo cenários adversos (reflexos fortes, sombras, baixa iluminação, ângulos variados) e garantindo representatividade de ocorrências pequenas e casos mistos. Esta evolução é particularmente importante para reduzir ambiguidades entre classes e para aproximar o sistema do cenário real de inspeção contínua em diferentes centrais, onde a variabilidade do ambiente é a regra e não a exceção. Se o âmbito do projeto incluir explicitamente “folhas” como fonte de sujidade, então essa classe deve ser tratada de forma assumida e

integrada no processo de anotação e avaliação, evitando desalinhamentos entre problema definido e classes efetivamente aprendidas.

Em paralelo, o trabalho avançará para a criação de um protótipo demonstrador: integração do modelo num fluxo completo de inspeção, com visualização clara de resultados, geração de alertas e registo histórico do estado dos painéis, idealmente com uma interface web e um componente móvel que tornem o *output* acionável. Esta fase implica testes funcionais com utilizadores e cenários próximos do real, e obriga também a decisões práticas sobre execução (local vs nuvem), desempenho (tempos de inferência, processamento em lote, limites de latência) e critérios operacionais para transformar deteções/máscaras em recomendações de ação. A maturidade do projeto em Projeto II vai depender desta transição: sair do “modelo que funciona” e chegar ao “sistema que ajuda a decidir”.

7. Referências

- [1] “Recorde de produção de renováveis abastece 71% do consumo de eletricidade em 2024.” Accessed: Nov. 29, 2025. [Online]. Available: <https://www.ren.pt/pt-pt/media/noticias/recorde-de-producao-de-renovaveis-abastece-71-do-consumo-de-eletricidade-em-2024>
- [2] “Energia em Números, edição 2025.” Accessed: Nov. 29, 2025. [Online]. Available: <https://www.dgeg.gov.pt/pt/destaques/energia-em-numeros-edicao-2025/>
- [3] D. Cristóvão Da Gama, “Edifício Restelo-Rua”, Accessed: Nov. 29, 2025. [Online]. Available: www.erse.pt
- [4] “Soiling Losses – Impact on the Performance of Photovoltaic Power Plants - IEA-PVPS.” Accessed: Oct. 20, 2025. [Online]. Available: <https://iea-pvps.org/key-topics/soiling-losses-impact-on-the-performance-of-photovoltaic-power-plants/>
- [5] “Energia Solar Fotovoltaica - ISQ Portugal.” Accessed: Oct. 20, 2025. [Online]. Available: https://www.isq.pt/insights/energia-solar-fotovoltaica/?utm_source=chatgpt.com
- [6] “EN IEC 62446-1 Sistemas Fotovoltaicos (PV) - Requisitos de Teste, Documentação e Manutenção - Parte 1: Sistemas Conectados à Rede - Documentação, Comissionamento, Teste e Inspeção.” Accessed: Oct. 20, 2025. [Online]. Available: <https://www.eurolab.net/pt/sektorel/yesil-enerji-testleri/en-iec-62446-1-fotovoltaik-pv-sistemler-test-dokumantasyon-ve-bakim-gereksinimleri-bolum-1-sebekeye-bagli-sistemler-dokumantasyon-devreye-alma-testleri-ve-denetim/>
- [7] E. K. Hajar, E. Rachid, B. Mohamed, A. Youssef, and M. Abdelkader, “Application of Deep Learning Based Detector YOLOv5 for Soiling Recognition in Photovoltaic Modules,” in *2023 3rd International Conference on Innovative Research in Applied Science, Engineering and Technology, IRASET 2023*, Institute of Electrical and Electronics Engineers Inc., 2023. doi: 10.1109/IRASET57153.2023.10152990.
- [8] X. Kong, W. Xu, B. Xu, H. Jin, and P. Zhan, “Defect detection of photovoltaic modules based on improved SSD algorithm,” in *2023 International Conference on Image Processing, Computer Vision and Machine Learning, ICICML 2023*, Institute of Electrical and Electronics Engineers Inc., 2023, pp. 1063–1066. doi: 10.1109/ICICML60161.2023.10424859.
- [9] S. Liu, C. Lin, Z. Xu, and Q. Fu, “Intelligent Detection System for Dirty Photovoltaic Panels Based on YOLOv8,” in *2025 IEEE International Conference on Mechatronics and Automation, ICMA 2025*, Institute of Electrical and Electronics Engineers Inc., 2025, pp. 199–203. doi: 10.1109/ICMA65362.2025.11120764.

- [10] S. Sarkar, C. Mukherjee, J. Adhikari, S. B. Pal, R. Ganguly, and S. Ghosh, "Utilizing the YOLO Framework for Unnecessary Particle Identification on Solar Panel Surfaces," in *7th International Conference on Energy, Power and Environment, ICEPE 2025*, Institute of Electrical and Electronics Engineers Inc., 2025. doi: 10.1109/ICEPE65965.2025.11139559.
- [11] D. F. Ramirez, D. Pujara, C. Tepedelenlioglu, D. Srinivasan, and A. Spanias, "Infrared Computer Vision for Utility-Scale Photovoltaic Array Inspection," in *15th International Conference on Information, Intelligence, Systems and Applications, IISA 2024*, Institute of Electrical and Electronics Engineers Inc., 2024. doi: 10.1109/IISA62523.2024.10786613.
- [12] N. Lionel, K. Bingi, R. Ibrahim, R. Korah, G. Kumar, and B. Rajanarayan Prusty, "Autonomous Inspection of Solar Panels and Wind Turbines Using YOLOv8 with Quadrotor Drones," in *Proceedings of the 9th International Conference on Mechatronics Engineering, ICOM 2024*, Institute of Electrical and Electronics Engineers Inc., 2024, pp. 322–326. doi: 10.1109/ICOM61675.2024.10652395.
- [13] A. Gargouri and F. Haddeji, "Implementation of a Photovoltaic Defect Detection System using Deep Learning," Institute of Electrical and Electronics Engineers (IEEE), Jul. 2024, pp. 1–5. doi: 10.1109/ic_aset61847.2024.10596189.
- [14] A. Hamdi, H. Noura, and J. Azar, "Dust Detection on Solar Panels Using DINOv2 with Multi-Head Self-Attention," in *21st International Wireless Communications and Mobile Computing Conference, IWCMC 2025*, Institute of Electrical and Electronics Engineers Inc., 2025, pp. 1686–1693. doi: 10.1109/IWCMC65282.2025.11059675.
- [15] "Trabalhar com camadas de pré-processamento | TensorFlow Core." Accessed: Oct. 27, 2025. [Online]. Available: https://www.tensorflow.org/guide/keras/preprocessing_layers?hl=pt-br
- [16] "(PDF) Defect detection of solar cell based on data augmentation." Accessed: Nov. 30, 2025. [Online]. Available: https://www.researchgate.net/publication/352819912_Defect_detection_of_solar_cell_based_on_data_augmentation
- [17] S. E. Go, J. H. Kim, T. Chuluunsaikhan, W. S. Choi, S. H. Choi, and A. Nasridinov, "Unified Generative Data Augmentation for Efficient Solar Panel Soiling Localization," *Electronics 2024, Vol. 13, Page 4859*, vol. 13, no. 24, p. 4859, Dec. 2024, doi: 10.3390/ELECTRONICS13244859.
- [18] "Roboflow: Computer vision tools for developers and enterprises." Accessed: Dec. 26, 2025. [Online]. Available: <https://roboflow.com/>
- [19] "Ultralytics." Accessed: Jan. 14, 2026. [Online]. Available: <https://www.ultralytics.com/>

- [20] “Explore Ultralytics YOLOv8 - Documentação Ultralytics YOLO.” Accessed: Dec. 06, 2025. [Online]. Available: <https://docs.ultralytics.com/pt/models/yolov8>
- [21] “Ultralytics YOLO11 - Documentos Ultralytics YOLO.” Accessed: Dec. 06, 2025. [Online]. Available: <https://docs.ultralytics.com/pt/models/yolo11/>
- [22] “YOLOv9: Um salto em frente na tecnologia de detecção de objectos.” Accessed: Dec. 06, 2025. [Online]. Available: <https://docs.ultralytics.com/pt/models/yolov9>
- [23] “Comparações de Modelos: Escolha o Melhor Modelo de Detecção de Objetos para o Seu Projeto.” Accessed: Dec. 06, 2025. [Online]. Available: <https://docs.ultralytics.com/pt/compare>
- [24] R. Sapkota and M. Karkee, “Ultralytics YOLO Evolution: An Overview of YOLO26, YOLO11, YOLOv8 and YOLOv5 Object Detectors for Computer Vision and Pattern Recognition,” Oct. 2025, Accessed: Dec. 06, 2025. [Online]. Available: <http://arxiv.org/abs/2510.09653>
- [25] H. Zhang, Z. Yang, and N. Lei, “Defect recognition of solar panel in EfficientNet-B3 network based on CBAM attention mechanism,” in *ACM International Conference Proceeding Series*, Association for Computing Machinery, May 2024, pp. 349–352. doi: 10.1145/3665348.3665407.
- [26] “Energia solar ligada à rede de transporte duplicou em 2024.” Accessed: Oct. 20, 2025. [Online]. Available: <https://www.ren.pt/media/noticias/energia-solar-ligada-a-rede-de-transporte-duplicou-em-2024>
- [27] T. Sun, H. Gao, S. Fan, and X. X. Hu, “Enhancing Dust Detection on Photovoltaic Panels with PP-YOLO: A Deep Learning Approach,” in *2024 4th International Conference on Machine Learning and Intelligent Systems Engineering, MLISE 2024*, Institute of Electrical and Electronics Engineers Inc., 2024, pp. 24–28. doi: 10.1109/MLISE62164.2024.10674106.
- [28] S. Liu, C. Lin, Z. Xu, and Q. Fu, “Intelligent Detection System for Dirty Photovoltaic Panels Based on YOLOv8,” in *2025 IEEE International Conference on Mechatronics and Automation, ICMA 2025*, Institute of Electrical and Electronics Engineers Inc., 2025, pp. 199–203. doi: 10.1109/ICMA65362.2025.11120764.
- [29] “Solar Panel Dust Detection Model > Overview.” Accessed: Dec. 26, 2025. [Online]. Available: <https://universe.roboflow.com/solarpaneldustdetection/solar-panel-dust-detection-5qnt0>
- [30] “Solar Panel > Browse.” Accessed: Dec. 26, 2025. [Online]. Available: <https://universe.roboflow.com/solar-panel-damage-detectionsegmentation/solar-panel-o6dwf/browse?queryText=-class%3ADefective+-class%3A%22Electrical-Damage%22+->

class%3A%22Non-Defective%22+-class%3A%22Physical-Damage%22+-
class%3ASnow&pageSize=200&startingIndex=0&browseQuery=true

- [31] “Kaggle Notebooks.” Accessed: Jan. 14, 2026. [Online]. Available: <https://www.kaggle.com/code>