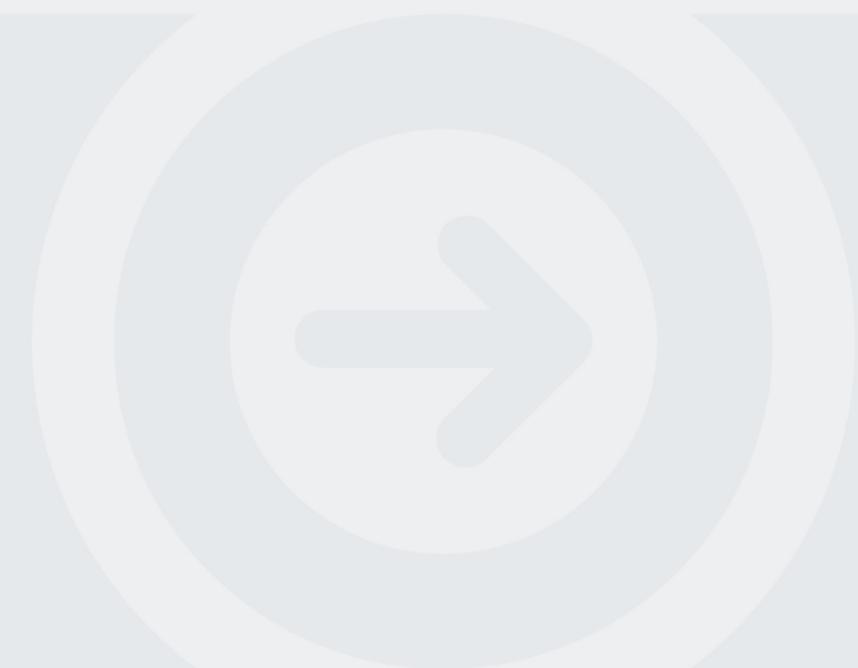


Mapeamento de anomalias na envolvente exterior de edifícios industriais com base em modelos tridimensionais fotogramétricos

PEDRO MIGUEL ALMEIDA MARQUES DE OLIVEIRA

Outubro de 2024



**Mapeamento de anomalias na
envolvente exterior de
edifícios industriais com base
em modelos tridimensionais
fotogramétricos**

**Pedro Miguel Almeida Marques
Oliveira**

**Dissertação para obtenção do Grau de
Mestre em Engenharia Civil, Área de
Especialização em Construções**

Orientador: Prof. Diogo Ribeiro

Co-orientador: Prof. Ricardo Santos

Júri:

Presidente: Prof. Maria do Rosário Oliveira

Vogais: Prof. João Poças Martins e Prof. Diogo Ribeiro

DECLARAÇÃO DE INTEGRIDADE

Declaro ter conduzido este trabalho académico com integridade. Não plagiei ou apliquei qualquer forma de uso indevido de informações ou falsificação de resultados ao longo do processo que levou à sua elaboração.

Declaro que o trabalho apresentado neste documento é original e de minha autoria, não tendo sido utilizado anteriormente para nenhum outro fim.

Declaro ainda que tenho pleno conhecimento do Código de Conduta Ética do P.PORTO.

ISEP, Porto, 04 de outubro de 2024

Pedro Miguel Almeida Marques Oliveira

Resumo

A monitorização e inspeção de ativos industriais são essenciais para garantir a segurança dos utilizadores, no entanto caracterizam-se por serem processos morosos, dispendiosos e sujeitos a erro humano. Com o avanço tecnológico, é possível efetuar a realização de inspeções mais seguras e eficientes, minimizando o trabalho e o risco para os intervenientes. A aplicação de algoritmos de IA na segmentação de instâncias em imagens destaca-se na deteção de anomalias estruturais, como são os casos da *Mask R-CNN* e do *YOLO*. Quando combinadas com plataformas UAV, estas redes tornam-se ferramentas poderosas para inspecionar ativos em zonas de difícil acesso, assegurando maior precisão. A tecnologia de *Ray Casting* complementa este processo, permitindo mapear danos identificados pela inteligência artificial nos modelos 3D concebidos através de tecnologias como o LiDAR ou a Fotogrametria. Neste trabalho foi desenvolvida uma metodologia com capacidade de mapear corrosão, choques mecânicos e acumulações de água em nuvens de pontos de ativos industriais com revestimentos em painéis *sandwich*, utilizando TLS e UAV. Os modelos treinados através do *Detectron2* e do *YOLOv8* conseguem segmentar os três tipos de danos com uma precisão de 68,8 % e 72,5 %,.

Palavras-chave: Monitorização, LiDAR, Fotogrametria, *Detectron2*, *YOLOv8*, *Ray Casting*

Abstract

The monitoring and inspection of industrial assets are essential to guarantee the safety of users, but they are time-consuming, costly and subject to human error. With technological advances, it is possible to carry out safer and more efficient inspections, minimizing work and risk for those involved. The application of AI algorithms in the segmentation of instances in images stands out in the detection of structural anomalies, as is the case with Mask R-CNN and YOLO. When combined with UAV platforms, these networks become powerful tools for inspecting assets in hard-to-reach areas, ensuring greater precision. Ray Casting technology complements this process, allowing damage identified by artificial intelligence to be mapped onto 3D models designed using technologies such as LiDAR or photogrammetry. This study developed a methodology capable of mapping corrosion, mechanical shocks and water accumulations in point clouds of industrial assets with sandwich panel cladding, using TLS and UAVs. The models trained using Detectron2 and YOLOv8 are able to segment the three types of damage with an accuracy of 68.8% and 72.5%.

Keywords: Monitoring, LiDAR, Photogrammetry, Detectron2, YOLOv8, Ray Casting

Agradecimentos

Aos professores e orientadores Diogo Ribeiro e Ricardo Santos, por todo o envolvimento e motivação que me ofereceram ao longo destes meses. Por sempre terem confiado em mim e por terem contribuído para a realização deste trabalho.

A toda a equipa da Multiprojectus, em especial ao Engenheiro Jorge Bastos, pela confiança e disponibilidade que tornou possível a aplicação da metodologia desenvolvida a um caso de estudo.

A todos os meus colegas de trabalho, Francisco, João, Jorge, Óscar, Rafael e Tomás por todo o apoio e incentivo que me fizeram acreditar em mim e pela fantástica equipa que integrei nos últimos meses, não só marcada pela amizade, mas também pelo grande profissionalismo associado.

À minha mãe, a pessoa que esteve presente em todas as etapas da minha vida. Um obrigado especial, por toda a educação e carinho que me tornaram na pessoa que sou hoje. Sem ela, nada disto era possível.

À Beatriz, a pessoa que me acompanhou ao longo deste percurso, tendo sempre as palavras certas, que me enchiam de amor e motivação. Um enorme obrigado por viveres as minhas conquistas como de tuas se tratassem.

Ao meu irmão, que sempre esteve comigo e me apoiou e por ser uma referência para mim, enquanto pessoa e profissional.

Ao meu pai, que apesar de não estar presente fisicamente, dá-me muita força para nunca desistir de nada.

A toda a minha família e amigos próximos que estiveram presentes nesta caminhada, motivando-me e contribuindo para o meu sucesso.

A todos referidos, um muito obrigado.

Índice

1	Introdução	1
1.1	Enquadramento	1
1.2	Objetivos	3
1.3	Estrutura da dissertação	3
2	Estado da Arte	5
2.1	Reconstrução geométrica 3D	5
2.1.1	LiDAR	5
2.1.2	Terrestrial Laser Scanner (TLS)	6
2.1.3	Fotogrametria baseada em UAVs	9
2.1.4	Fusão de dados	13
2.2	Inteligência artificial	14
2.2.1	Redes Neurais Artificiais	15
2.2.2	Redes Neurais Convolucionais	19
2.2.3	Redes Neurais Convolucionais por Região	25
2.3	<i>Ray Casting</i>	38
3	Metodologia de deteção automática de anomalias	43
3.1	Aquisição de dados	44
3.1.1	<i>BLK360 G1, Cyclone Field 360 e iPad</i>	44
3.1.2	DJI M2EA e DJI M3M	45
3.2	Processamento de dados	47
3.2.1	<i>Cyclone Register 360</i>	47
3.2.2	<i>iTwin Capture Modeler Master & Engine</i>	50
3.3	Fusão das nuvens de pontos	54
3.4	Algoritmo de deteção de corrosão, choque mecânico e acumulação de água	56
3.4.1	Registo do <i>dataset</i>	57
3.4.2	Treino	59
3.4.3	Teste e inferências	68
3.5	<i>Ray Casting</i>	71
4	Aplicação da metodologia a um ativo industrial	83
4.1	Memória descritiva	83
4.2	Aquisição de dados	83
4.2.1	<i>BLK360 G1, Cyclone Field e iPad</i>	84
4.2.2	DJI M2EA e DJI M3M	87
4.3	Processamento de dados e criação da nuvem de pontos e da <i>mesh</i>	88
4.3.1	<i>Cyclone Register 360</i>	88
4.3.2	<i>iTwin Capture Modeler</i>	90
4.3.3	<i>CloudCompare</i>	92

4.4	Aplicação do modelo de identificação de anomalias	93
4.5	Modelo fotogramétrico com patologias	96
5	Conclusões e desenvolvimentos futuros	103
5.1	Conclusões.....	103
5.2	Desenvolvimentos futuros	105
	Referências bibliográficas	107

Lista de Figuras

Figura 1 - Diferença entre classificação, deteção e segmentação	2
Figura 2 - Funcionamento conjunto das tecnologias digitais.....	5
Figura 3 - Métodos de princípio de funcionamento de um TLS (Wang e Mei, 2020)	6
Figura 4 - Exemplo de alvos que podem funcionar como GCP's (Tang <i>et al.</i> , 2020).....	7
Figura 5 - Utilização do MLS (Grupo Acre Portugal, 2019)	8
Figura 6 - Princípio de triangulação com <i>smartphone</i> (adaptado de Błaszczak-Bąk <i>et al.</i> , 2023)	8
Figura 7 - Levantamento MLS com auxílio do SMT (Błaszczak-Bąk <i>et al.</i> , 2023)	9
Figura 8 - Esquematização dos parâmetros extrínsecos da câmara (Alshawabkeh e Baik, 2023)	12
Figura 9 - Comportamento dos GSD com a variação da altura do UAV (Wingtra One, 2019)	12
Figura 10 - Método gráfico para localizar o centro geométrico de duas nuvens de pontos (Cao <i>et al.</i> , 2023).....	14
Figura 11 - Relação entre IA, ML e DL (Ye, Jin e Yun, 2019)	15
Figura 12 - Representação de um neurónio (Haykin, 2009)	16
Figura 13 - Arquitetura <i>Feedforward</i> Multicamadas de uma Rede Neuronal Artificial (Bre, Gimenez e Fachinotti, 2018)	17
Figura 14 - Rede <i>Feedforward</i> de Camada Única (Haykin, 2009)	18
Figura 15 - Rede Recorrente sem (a) e com (b) <i>loops</i> de <i>auto-feedback</i> e neurónios ocultos (Haykin, 2009)	19
Figura 16 - Arquitetura de uma CNN (Phung e Rhee, 2018)	20
Figura 17 - Exemplo de processo de convolução 2D (Vieira <i>et al.</i> , 2021)	20
Figura 18 - Exemplo de <i>max</i> e <i>mean pooling</i> (Aljaafari, 2018)	22
Figura 19 - Exemplo de operação de <i>flattening</i> (SuperDataScience Team, 2018).....	23
Figura 20 - Exemplos <i>underfitting</i> , <i>optimal-fitting</i> e <i>overfitting</i> em Regressão, Classificação e DL (Towards Data Science, 2021).....	25
Figura 21 - Evolução da deteção de objetos no DL (adaptado de Hosseinzadeh <i>et al.</i> , 2024)	26
Figura 22 - Arquitetura da rede R-CNN (Girshick <i>et al.</i> , 2013).....	27
Figura 23 - Arquitetura da rede <i>Fast R-CNN</i> (Girshick, 2015)	28
Figura 24 - Arquitetura <i>Faster R-CNN</i> (a) e funcionamento da RPN (b)	29
Figura 25 - Evolução da família YOLO (GeeksforGeeks, 2024).....	30
Figura 26 - Arquitetura base da YOLO (Bochkovskiy, Wang e Liao, 2020).....	31
Figura 27 - Esquema representativo do funcionamento da deteção de um objeto através de <i>bounding boxes</i> (Silva, 2024)	31
Figura 28 - Exemplo de <i>object detection</i> recorrendo à YOLO (Redmon <i>et al.</i> , 2015)	32
Figura 29 - Arquitetura de uma <i>Mask R-CNN</i> comparada com a de uma <i>Faster R-CNN</i> (Xu <i>et al.</i> , 2022) 33	
Figura 30 - Arquitetura de uma <i>Mask R-CNN</i> , utilizando uma <i>Resnet-50</i> como <i>Backbone</i> (Santos, 2022)	34
Figura 31 - <i>Ray Casting</i> pretendido	38
Figura 32 - Esquema do conceito HELIOS++ (Winiwarter <i>et al.</i> , 2022).....	40
Figura 33 - Metodologia desenvolvida para deteção automática de anomalias	43
Figura 34 - <i>BLK360 G1</i> (Allen Precision Equipment, 2023)	44
Figura 35 - Utilização do <i>Cyclone FIELD 360</i> em campo.....	45
Figura 36 - DJI M2EA (a) e DJI M3M (b) (DJI, 2023)(DJI, 2024)	45
Figura 37 - Comando de controlo do DJI M3 (DJI, 2024)	46
Figura 38 - Esquema do processamento de dados obtidos	47
Figura 39 - Exemplo de melhoria dos links entre estações do <i>Cyclone FIELD 360</i> (a) para o <i>Cyclone Register 360</i> (b).....	48
Figura 40 - Georreferenciação no <i>Cyclone Register 360</i>	48
Figura 41 - Excerto de um relatório final do <i>Cyclone Register 360</i>	49
Figura 42 - Exemplo do resultado da nuvem de pontos do <i>BLK360 G1</i>	50
Figura 43 - Posicionamento das imagens importadas no <i>iTwin</i>	51
Figura 44 - Importação das coordenadas (a) e marcação dos pontos (b) no <i>iTwin</i>	51

Figura 45 - Definições de aerotriangulação no <i>iTwin</i>	52
Figura 46 - Seleção da área de interesse a ser reconstruída.....	53
Figura 47 - Definições da nuvem de pontos (a) e da <i>mesh</i> (b).....	53
Figura 48 - Exemplo de resultado da <i>mesh</i>	54
Figura 49 - Resultados das nuvens de pontos do LiDAR (a) e da fotogrametria (b).....	54
Figura 50 - Parâmetros a definir no ICP.....	55
Figura 51 - Resultado da fusão aplicada na pala da NACEX.....	55
Figura 52 - Processo de desenvolvimento do algoritmo de detecção de anomalias.....	56
Figura 53 - Comparação entre a <i>YOLOv8</i> e as restantes redes YOLO (Ultralytics, 2022).....	57
Figura 54 - Exportação das anotações (a) e ferramenta de anotações (b) no VIA.....	58
Figura 55 - Exportação das anotações (a) e ferramenta de anotações (b) no <i>Roboflow</i>	59
Figura 56 - Impacto do recorte 2000x2000 no número de imagens de treino.....	60
Figura 57 - Resultado da introdução de pontos (b) em anotações desconfiguradas (a).....	62
Figura 58 - Operações de <i>Data Augmentation</i>	63
Figura 59 - Operações de <i>Data Augmentation</i> no <i>Roboflow</i>	64
Figura 60 - Exemplos de inferências dos modelos <i>Detectron2</i> e <i>YOLOv8</i>	70
Figura 61 - Metodologia de <i>Ray Casting</i>	71
Figura 62 - <i>Mesh</i> do quadro com alvos.....	72
Figura 63 - Exemplos de remoção da distorção em imagens.....	72
Figura 64 - Exemplos de equalização em imagens.....	73
Figura 65 - Exemplos de imagens inferidas.....	74
Figura 66 - Remoção do <i>background</i> com filtro RGB.....	74
Figura 67 - Resultado do <i>Ray Casting</i> inicial.....	75
Figura 68 - Explicação da DBSCAN (a) (Pandey, 2020) e resultado na pala do edifício industrial da NACEX (b).....	75
Figura 69 - Exemplos das quinze imagens mais próximas (a) e das cinco imagens mais ortogonais (b) a um determinado dano específico.....	76
Figura 70 - Modelo de câmara <i>pinhole</i> (Doxygen, 2024).....	77
Figura 71 - Exemplo de uma projeção do centróide do <i>cluster</i> 3D no dano 2D.....	79
Figura 72 - Excerto de uma imagem no ficheiro <i>.xml</i>	79
Figura 73 - Exemplo de corrosão em <i>Ray Casting</i> final.....	80
Figura 74 - Imagem geral do ativo industrial em estudo.....	83
Figura 75 - Localização das estações e dos alvos.....	84
Figura 76 - Montagem do <i>BLK360 G1</i> (a), colocação dos alvos (b), localização das estações (c).....	85
Figura 77 - Definição dos parâmetros dos <i>scans</i> (a) e resultado da campanha (b).....	86
Figura 78 - Nuvem de pontos resultante do pré-processamento do levantamento efetuado com o <i>BLK360 G1</i>	86
Figura 79 - Utilização do comando de controlo (a) e voo automático (b).....	87
Figura 80 - Exemplos de imagens do ativo capturadas pelos drones.....	88
Figura 81 - Melhoria dos <i>links</i> do caso de estudo no <i>Cyclone Register 360</i>	89
Figura 82 - Marcação dos alvos (a) e identificação dos GCP's utilizados (b).....	89
Figura 83 - Excerto do relatório final do <i>Cyclone Register 360</i>	90
Figura 84 - Resultado da nuvem de pontos do <i>BLK360 G1</i>	90
Figura 85 - Georreferenciação dos alvos no <i>iTwin Capture Modeler</i>	91
Figura 86 - Perspetivas da <i>mesh</i> do caso de estudo resultante do <i>iTwin Capture Modeler</i>	91
Figura 87 - Nuvens de pontos do LiDAR (a) e da fotogrametria (b) do caso de estudo.....	92
Figura 88 - Nuvem de pontos do caso de estudo final resultante da ICP.....	92
Figura 89 - Exemplos de predições <i>Detectron2</i> e <i>YOLOv8</i> em imagens da cobertura do caso de estudo.....	93
Figura 90 - Exemplos de predições <i>Detectron2</i> e <i>YOLOv8</i> em imagens da fachada Este do caso de estudo.....	94
Figura 91 - Exemplos de predições <i>Detectron2</i> e <i>YOLOv8</i> em imagens da fachada Oeste e Sul do caso de estudo.....	95
Figura 92 - Modelo da fachada Norte.....	96
Figura 93 - Modelo da fachada Este.....	97
Figura 94 - Modelo da fachada Oeste.....	98

Figura 95 - Modelo da fachada Oeste (continuação).....	99
Figura 96 - Modelo da fachada Sul	100
Figura 97 - Modelo da cobertura	101
Figura 98 - Modelo da cobertura (continuação).....	102

Lista de Tabelas

Tabela 1 - Matriz de confusão	35
Tabela 2 - Precisão e duração de levantamentos consoante o modo de densidade.....	44
Tabela 3- Comparação de características entre o DJI M2EA e o DJI M3M -	46
Tabela 4 - Comparação de rendimentos entre algoritmos <i>Mask R-CNN</i> (Detectron2 contributors, 2019)	56
Tabela 5 - Comparação dos valores máximos obtidos entre as imagens originais e recortadas.....	61
Tabela 6 - Resultados máximos dos treinos com anotações corrigidas.....	62
Tabela 7 - Intervalos e valores otimizados das operações de <i>Data Augmentation</i>	64
Tabela 8 - Resultados máximos dos treinos com <i>Data Augmentation</i> aplicado	65
Tabela 9 - Valores iniciais e otimizados para os treinos do <i>Detectron2</i> e do <i>YOLOv8</i>	67
Tabela 10 - Valores máximos obtidos para o <i>Detectron2</i> e para o <i>YOLOv8</i> com os hiperparâmetros otimizados.....	68
Tabela 11 - Valores máximos de teste obtidos para os melhores modelos	69
Tabela 12 - Resultados das medidas da experiência em ambiente controlado.....	81

Acrónimos

Lista de Acrónimos e Siglas

AEC	<i>Arquitetura, Engenharia e Construção</i>
ANN	<i>Artificial Neural Network</i>
AP	<i>Average Precision</i>
C2C	<i>Cloud to Cloud</i>
C2M	<i>Cloud to Mesh Distance</i>
CNN	<i>Convolutional Neural Network</i>
CUDA	<i>Compute Unified Device Architecture</i>
CV	<i>Computer Vision</i>
DL	<i>Deep Learning</i>
EASA	<i>European Union Aviation Safety Agency</i>
FC	<i>Fully-Connected</i>
FN	<i>False Negative</i>
FP	<i>False Positive</i>
GB	<i>Gigabyte</i>
GCP	<i>Ground Control Point</i>
GNSS	<i>Global Navigation Satellite System</i>
GPU	<i>Graphics Processing Unit</i>
GSD	<i>Ground Sample Distance</i>

ACRÓNIMOS

HCI	<i>Human-Computer Interaction</i>
HDR	<i>High Dynamic Range</i>
IA	<i>Inteligência Artificial</i>
IoU	<i>Intersect over Union</i>
IMU	<i>Inertial Measurement Unit</i>
IS	<i>Instance Segmentation</i>
LiDAR	<i>Light Detection and Ranging</i>
M3C2	<i>Multiscale Model to Model Cloud Comparison</i>
mAP	<i>mean Average Precision</i>
ML	<i>Machine Learning</i>
MLS	<i>Mobile Laser Scanner</i>
MVS	<i>Multi-View Stereo</i>
OD	<i>Object Detection</i>
px	<i>Pixel</i>
RAM	<i>Random Access Memory</i>
RANSAC	<i>Random Sample Consensus</i>
ReLU	<i>Rectified Linear Unit</i>
RGB	<i>Red Green Blue</i>
RoI	<i>Region of Interest</i>
RMS	<i>Root Mean Square</i>
RTK	<i>Real-Time Kinematic</i>

SfM	<i>Structure from Motion</i>
SMT	<i>Spatial Measurement Targets</i>
TB	<i>Terabyte</i>
TLS	<i>Terrestrial Laser Scanner</i>
TN	<i>True Negative</i>
TOF	<i>Time of Flight</i>
TP	<i>True Positive</i>
UAV	<i>Unmanned Aerial Vehicle</i>
YOLO	<i>You Only Look Once</i>

1 Introdução

1.1 Enquadramento

Os avanços tecnológicos e a sua incorporação na Engenharia Civil têm impulsionado o desenvolvimento das sociedades em termos de construção. No entanto, diversos edifícios não podem ser substituídos, seja por razões financeiras ou pelo seu valor histórico, o que torna necessário realizar monitorizações e inspeções das suas estruturas, com o intuito de prolongar a sua vida útil (Santos, 2022).

Os atuais procedimentos de inspeção de edifícios e infraestruturas caracterizam-se por serem processos morosos, subjetivos, difíceis de documentar e com riscos associados à segurança dos operadores. Por isso, é fundamental adotar novos procedimentos que, com recurso a tecnologias avançadas, permitam automatizar as inspeções, eliminando muitos dos problemas associados a estes processos (Pantoja-Rosero, Achanta e Beyer, 2023).

A Indústria 4.0 provocou uma revolução nas metodologias aplicadas ao setor de Arquitetura, Engenharia e Construção (AEC), introduzindo inovação no modo como as obras são planeadas, executadas e monitorizadas. A aplicação de tecnologias digitais, como LiDAR (*Light Detection and Ranging*) e fotogrametria, com recurso a *Terrestrial Laser Scanners* (TLS's) e *Unmanned Aerial Vehicles* (UAV's), consolidou-se como uma ferramenta essencial, possibilitando maior controlo sobre todas as fases de uma obra, desde a construção até a operação e manutenção (Nwaogu *et al.*, 2023). De acordo com Zhao e Lv (2023), a análise de modelos tridimensionais completos, provenientes de tecnologias digitais aplicadas *in situ*, apresenta vantagens significativas na conservação, restauro e construção de diversas estruturas, sendo considerada um método de inspeção não destrutivo, que se destaca pela elevada eficiência na obtenção de informações e com custos reduzidos.

Com o avanço da capacidade de processamento dos sistemas informáticos e sua acessibilidade no mercado de consumo e o crescimento da Inteligência Artificial (IA) na análise de imagens, o *deep learning* (DL) e a visão computacional (*computer Vision* - CV) têm assumido papéis fundamentais na automatização dos processos de análise de informação gráfica. A sua aplicação mais bem-sucedida, as Redes Neurais Convolucionais (*Convolutional Neural Networks* - CNN), tem obtido resultados promissores nos campos da classificação de imagens (*image classification* - IC), deteção de objetos (*object detection* - OD) e segmentação de instâncias (*instance*

segmentation - IS). A diferença entre estas três áreas está representada na Figura 1 (Burger e Burge, 2012).



Figura 1 - Diferença entre classificação, deteção e segmentação

No âmbito da engenharia civil, as Redes Neurais Convolucionais (CNN) já são amplamente utilizadas para a identificação automática de patologias, como fissuras, exposição de armaduras e corrosões, a partir de imagens de estruturas de betão armado, conforme diversas pesquisas demonstram (Santos, 2022). Ao longo dos anos, as CNN foram sendo aprimoradas, com o intuito de expandir as suas capacidades, resultando no surgimento das *Region-based Convolutional Neural Networks* (R-CNN). Estas redes interpretam a localização de objetos como um problema de regressão, através de regiões propostas, o que lhes permitiu obter melhores desempenhos (Girshick *et al.*, 2013). Posteriormente, as redes YOLO destacaram-se, pois para além de localizarem objetos conseguem também segmentá-los, tanto em imagens como em vídeos, resolvendo também problemas de eficiência (Redmon *et al.*, 2015). Distinguem-se ainda pela sua facilidade de aplicação em comparação a outras redes. Devido à eficiência e ao potencial de crescimento destas aplicações, em 2017, He *et al.* (2017) propuseram a *Mask R-CNN*, que combina Redes Neurais Artificiais (RNA) com CNN's, sendo conhecidas pela sua excelência quando o objetivo se trata da IS (Aggarwal, 2018).

Com o intuito de automatizar de forma completa a inspeção de um edifício industrial, as instâncias geradas pelos modelos de IA, são mapeadas no modelo tridimensional do mesmo. Esse mapeamento é realizado com recurso a uma técnica de projeção, designada por *Ray Casting*. Esta técnica consiste numa ferramenta essencial em gráficos computacionais, utilizada para detetar interseções entre raios virtuais de luz e objetos em ambientes 3D. Contrariamente

a métodos mais complexos, como o *ray tracing*, o *Ray Casting* limita-se em identificar o primeiro ponto de cruzamento de um raio com um objeto tridimensional. Este método é amplamente implementado em simulações de LiDAR e renderização em tempo real devido à sua eficiência (Rougeron, Garrec e Andriot, 2022).

1.2 Objetivos

A metodologia proposta visa tornar mais eficiente a monitorização de edifícios industriais, utilizando modelos 3D e visão computacional para, de forma automática, identificar e mapear danos na sua envolvente. Neste sentido, definiram-se os seguintes objetivos específicos:

- Realizar um enquadramento teórico sobre inovações em visão computacional e inteligência artificial (IA), com ênfase na aplicação dos algoritmos *Mask R-CNN* e *YOLO* para a segmentação de anomalias;
- Obtenção de dados (imagens e nuvens de pontos) da envolvente de edifícios industriais, com recurso a UAV's e a TLS, com o intuito de criar modelos tridimensionais;
- Desenvolvimento de uma aplicação, baseada nos algoritmos *Mask R-CNN* e *YOLO*, com a capacidade de segmentar automaticamente corrosões, choques mecânicos e acumulações de água em painéis sandwich;
- Desenvolvimento de um procedimento de visualização dos danos identificados no modelo 3D, com recurso a uma técnica de projeção do tipo *Ray Casting*.

1.3 Estrutura da dissertação

A dissertação encontra-se estruturada em cinco capítulos. O presente capítulo, a Introdução, enquadra o tema do trabalho, apresenta os objetivos gerais e específicos e a estrutura do documento. O segundo capítulo, de Estado da Arte, funciona como um enquadramento teórico, explorando temas como a reconstrução geométrica 3D, focando em particular as tecnologias digitais utilizadas. Também são abordadas as diferentes Redes Neurais Artificiais (*Artificial Neural Network* - ANN) e CNN's, no âmbito da identificação de anomalias em edifícios, seguida de uma secção relacionada com *Ray Casting*. O terceiro capítulo, apresenta a metodologia desenvolvida para o mapeamento de danos, descreve os procedimentos utilizados para aquisição e processamento de dados, o trabalho desenvolvido na conceção de modelos de

segmentação de anomalias e o *Ray Casting*. De forma a validar a metodologia concebida, surge o quarto capítulo, que consiste na aplicação da metodologia a um ativo industrial. O quinto capítulo, apresenta as conclusões finais do trabalho e os desenvolvimentos futuros. O documento finaliza com as Referências, que listam as fontes citadas ao longo da dissertação.

2 Estado da Arte

O presente capítulo aborda as tecnologias e métodos fundamentais para o desenvolvimento da metodologia proposta, dividindo-se em três partes principais. A primeira secção explora aspetos técnicos associados à reconstrução geométrica 3D, com recurso a tecnologias como LiDAR e fotogrametria. A fusão de dados é obtida através de UAV's, TLS's e MLS's, culminando na conceção de modelos 3D precisos. Por sua vez, a segunda secção analisa a evolução das Redes Neurais Artificiais, destacando a *Mask R-CNN* e a *YOLO* como tipos de redes ideais para a segmentação de instâncias. Por fim, a técnica de Ray Casting é apresentada como uma ferramenta de projeção de raios, com o potencial de permitir o mapeamento dos danos identificados nos modelos 3D reconstruídos, através de IA.

2.1 Reconstrução geométrica 3D

Existem duas abordagens para a reconstrução 3D de um objeto a partir de tecnologias digitais. Os sistemas ativos, como o sistema LiDAR, utilizam radiação (como sinais de *laser*) para realizar medições remotas. Por outro lado, os sistemas passivos utilizam sensores que captam radiação electromagnética, essencialmente refletida, normalmente no intervalo do visível (Figura 2).

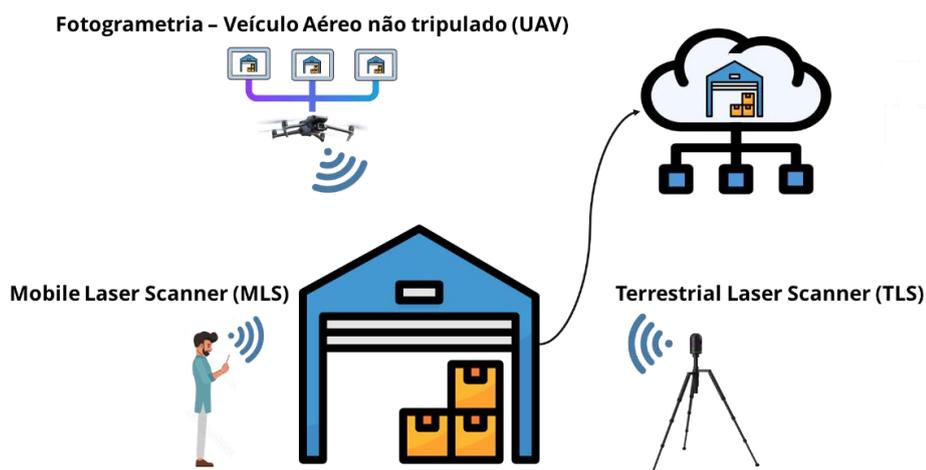


Figura 2 - Funcionamento conjunto das tecnologias digitais

2.1.1 LiDAR

Através do varrimento com recurso a esta tecnologia, é possível efetuar medições e também

proceder à digitalização tridimensional da informação obtida. Esta técnica avançada de levantamento oferece precisão elevada, permitindo a criação de modelos detalhados, com recurso à conceção de nuvem de pontos e ao registo de dados geoespaciais 3D. A tecnologia LiDAR pode ser aplicada, tanto através de TLS, ou mais recentemente com recurso a *Mobile Laser Scanner* (MLS), duas estratégias que serão exploradas nas secções seguintes.

2.1.2 Terrestrial Laser Scanner (TLS)

Para a obtenção de nuvens de pontos tridimensionais de alta resolução, recorre-se a *Terrestrial Laser Scanners*, que são reconhecidos pela capacidade de capturar a geometria dos objetos em toda a sua envolvente através de feixes de laser emitidos em todas as direções (Oliveira et al., 2023).

Este tipo de levantamento pode ser realizado através da medição laser baseada em triangulação (Kim et al. (2020)) ou da rotação do espelho de reflexão do aparelho, cobrindo assim vários ângulos, enquanto o aparelho vai girando. A aquisição dos dados pode ser realizada através de dois métodos, tal como explicado por Alshawabkeh e Baik (2023), e Wang e Mei (2020):

- **Time of flight (TOF) method** (Figura 3 (a)): emissão de feixes que atingem a superfície de um objeto. Esses mesmos feixes são refletidos de volta ao sensor do equipamento e com base na duração e direção do sinal, é possível determinar a distância do ponto em relação ao TLS;
- **Phase based method** (Figura 3 (b)): emissão de impulsos, onde o *laser scanner* mede a diferença de fase entre o comprimento de onda emitido e recebido.

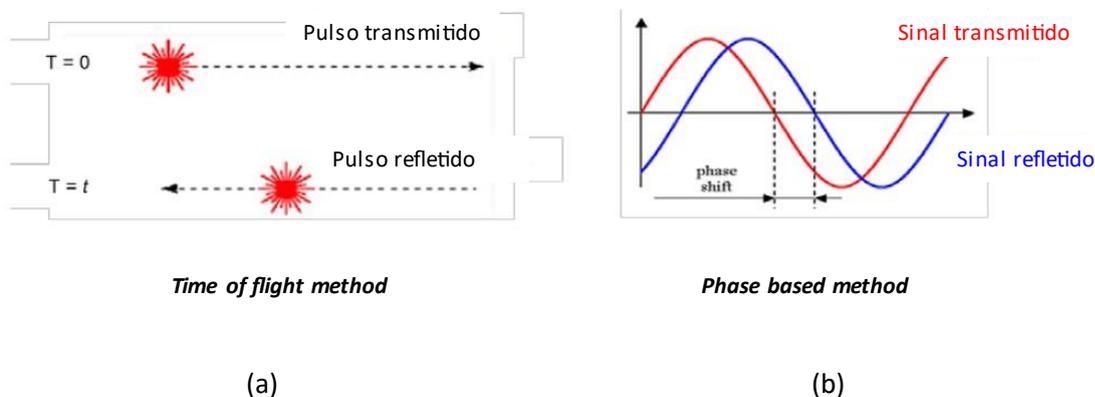


Figura 3 - Métodos de princípio de funcionamento de um TLS (Wang e Mei, 2020)

Antes da utilização de um TLS, é importante realizar uma visita ao local e perceber as dimensões do ativo a estudar e as condições de acesso, de modo a facilitar o planeamento da execução dos trabalhos. Esta fase é essencial, de modo a determinar a quantidade de levantamentos e de estações necessários, a localização das estações ou pontos de controlo terrestre (*Ground Control Points* - GCPs) e o modo de densidade do varrimento. Durante o levantamento, é crucial garantir a sobreposição adequada das nuvens de pontos para estabelecer associações robustas entre estações, tendo em vista a criação de *bundles*. Essa sobreposição é essencial para garantir a qualidade dos dados e facilitar o alinhamento e integração dos modelos obtidos com o sistema de coordenadas geodésicas. Para isso, é necessário utilizar os GCP (Figura 4), por intermédio de métodos topográficos, para posteriormente proceder ao alinhamento e integração dos modelos obtidos. A qualidade da nuvem de pontos está correlacionada com a localização e com o número de estações LiDAR, de forma a obter elevada sobreposição de *scans* e cobertura total do caso de estudo. Dessa forma, a maximização do número de estações torna-se atrativo para o utilizador, porém, esse aumento vai condicionar o tempo de processamento da informação obtida (Oliveira *et al.*, 2023).



Figura 4 - Exemplo de alvos que podem funcionar como GCP's (Tang *et al.*, 2020)

2.1.2.1 Mobile Laser Scanner (MLS)

Nos últimos anos, a tecnologia LiDAR tem sido integrada em dispositivos móveis, com destaque para a Apple, que em 2020 lançou o primeiro *smartphone* equipado com sensores LiDAR. Este tipo de equipamentos pode ser uma alternativa ao TLS em levantamentos de áreas de difícil acesso, destacando-se pela sua portabilidade e mobilidade, tal como demonstrado na Figura 5.



Figura 5 - Utilização do MLS (Grupo Acre Portugal, 2019)

O método de digitalização tem como base o princípio de triangulação (Figura 6), no qual os feixes incidentes são projetados no objeto, proporcionando a determinação da distância do dispositivo ao objeto de estudo através do posicionamento da câmara e do ângulo de incidência. Na Figura 6, Z é um eixo ótico, O é a origem, P é um ponto único no objeto. A linha de base é a distância entre o centro ótico e o laser, e relativamente aos ângulos, θ é o ângulo entre a linha de base e o plano do laser e α é o ângulo entre a linha de base e o segmento OP. Os parâmetros θ e α são fundamentais para a precisão das medições, já que quanto maior for θ , maior será a cobertura espacial do padrão do laser. Por sua vez, o parâmetro α é influenciado pela complexidade da geometria e se a superfície é plana ou não (Błaszczak-Bąk *et al.*, 2023).

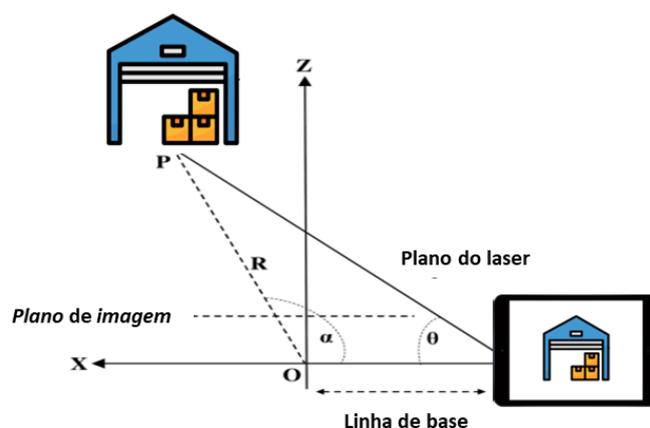


Figura 6 - Princípio de triangulação com *smartphone* (adaptado de Błaszczak-Bąk *et al.*, 2023)

Após analisar o desempenho das medições do laser LiDAR incorporado no *smartphone*, Błaszczak-Bąk et al. (2023) mostraram que as medições com LiDAR em smartphones funcionam em superfícies complexas, mas falham em superfícies lisas e homogêneas. Com o intuito de resolver essas dificuldades, foi desenvolvida a metodologia *Spatial Measurement Targets* (SMT), que adiciona relevo a essas superfícies, permitindo a fusão entre os *scans* de superfícies homogêneas e os de superfícies heterogêneas, proporcionando assim uma diminuição do ruído, tal como demonstrado na Figura 7 (Błaszczak-Bąk et al., 2023).

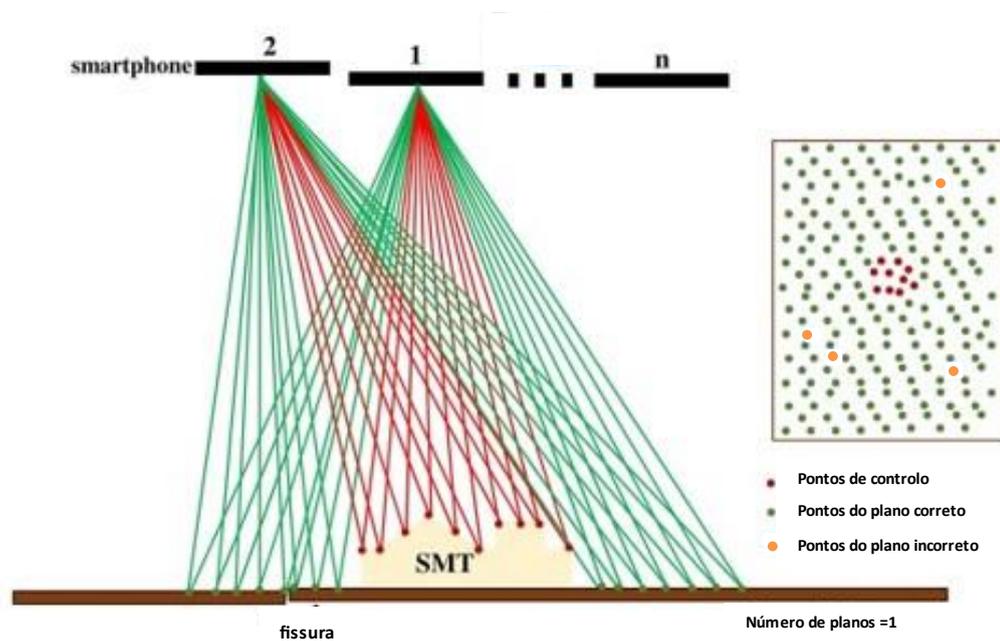


Figura 7 - Levantamento MLS com auxílio do SMT (Błaszczak-Bąk et al., 2023)

2.1.3 Fotogrametria baseada em UAVs

Enquanto o LiDAR utiliza lasers para medir distâncias e criar mapas de pontos 3D, a fotogrametria baseia-se na captura de imagens bidimensionais (2D) de diferentes ângulos, que são posteriormente processadas para criar modelos tridimensionais detalhados. Esta técnica permite a reconstrução precisa de um cenário ou objeto a partir de imagens obtidas sob diversas perspectivas, utilizando projeções matemáticas e sobreposição (Fiorillo, Limongiello e Bolognesi, 2021).

Existem diferentes abordagens na fotogrametria, dependendo da necessidade e das características do objeto a ser capturado (Kovanič et al., 2023):

- *close-range photogrammetry* - captura de imagens a curta distância, tipicamente a

menos de 300 metros, e que pode ser realizada tanto por via aérea quanto terrestre;

- *camera array* - câmaras fixas de modo a obter imagens de várias perspetivas;
- *wide-angle photogrammetry* - utilização de lentes panorâmicas de grande abertura angular, permitindo uma cobertura mais ampla do cenário;
- *aerial photogrammetry* - usada em espaços abertos, com auxílio de satélites, aviões, helicópteros ou UAVs, destacando-se pela capacidade de capturar grandes áreas com rapidez e eficiência. Normalmente está associada a uma resolução relativamente baixa, que limita o seu campo de aplicação.

A reconstrução de modelos 3D é realizada por meio da metodologia *Structure from Motion* (SfM), um algoritmo amplamente utilizado que permite criar um modelo tridimensional a partir de múltiplas imagens 2D capturadas em movimento. Este processo envolve a atribuição de uma posição específica em latitude, longitude e altitude a cada pixel de uma imagem (Chatzistamatis et al., 2018; Deliry e Avdan, 2021). A SfM surgiu como uma evolução da visão computacional e da fotogrametria tradicional, possibilitando a reconstrução precisa de estruturas complexas. Além disso, o método *Multi-View Stereo* (MVS) é frequentemente aplicado para densificar a nuvem de pontos gerada pelo SfM, resultando em modelos 3D mais detalhados que capturam com exatidão as características essenciais das estruturas.

2.1.3.1 Plataformas UAV

A utilização dos UAVs em Portugal é acompanhada pelo Regulamento n.º 1093/2016 da Autoridade Nacional da Aviação Civil, publicado pelo Diário da República de 14 de dezembro 2016, onde estão definidas as “Condições de operação aplicáveis à utilização do espaço aéreo pelos sistemas de aeronaves civis pilotadas remotamente (“Drones”)”. Para além de regras gerais de operação, este documento apresenta alguns condicionamentos e interdições para a realização de voos. Além desta regulamentação, a utilização deve seguir outras legislações, nomeadamente o Decreto-Lei nº58/2018, a Portaria nº2/2021 e ainda a adaptação das recomendações europeias emanadas pela EASA (*European Union Aviation Safety Agency*).

Atualmente, os drones têm incorporados (Alshawabkeh e Baik, 2023):

- Sistemas inerciais IMU (*Inertial Measurement Unit*) – Condicionam a precisão dos resultados de georreferenciação e incluem geralmente um conjunto de acelerómetros,

giroscópios, barômetros e magnetômetros;

- *Gimbal* - sistema estabilizador da câmara incorporado no UAV que permite isolar do sensor alguns dos movimentos inerciais do UAV e minimiza os efeitos de desfocagem (Chatzistamatis *et al.*, 2018);
- Sistemas de navegação por satélite (*Global Navigation Satellite System* - GNSS)
- Sistemas de correção de posicionamento *Real-Time Kinematic* (RTK) - permitem calcular em tempo real a posição do UAV no espaço com bastante precisão, comparativamente com os sistemas inerciais ou os sistemas baseados em GNSS (Fabris, Fontana Granotto e Monego, 2023).

O uso destes sensores reduz a necessidade de muitos pontos de controle no solo (GCPs), tornando o processo de levantamento mais ágil e eficiente. Para além destes sensores, o drone contempla ainda uma câmara, que pode ser:

- Câmara 4K e HD;
- Câmara térmica ou multiespectral;
- Sensor LiDAR;
- Câmara 360°.

Existem dois tipos de parâmetros da câmara que devem ser considerados no processamento: os parâmetros intrínsecos e extrínsecos. Os parâmetros intrínsecos são necessários para relacionar o ponto do *pixel* com o sistema de coordenadas da câmara, e ao mesmo tempo, são responsáveis por registar as informações correspondentes às configurações da câmara, como por exemplo, a distância focal, os fatores de escala, as posições, as dimensões dos *pixels* na matriz de sensores e ainda as distorções associadas às lentes. Por outro lado, os parâmetros extrínsecos registam os 6 graus de liberdade do objeto no espaço, compostos por 3 coordenadas (x,y,z) e 3 ângulos diedros (w , ϕ e k) (Figura 8). As coordenadas (x,y,z) correspondem à latitude, longitude e altitude, respetivamente (Nigri, 2009).

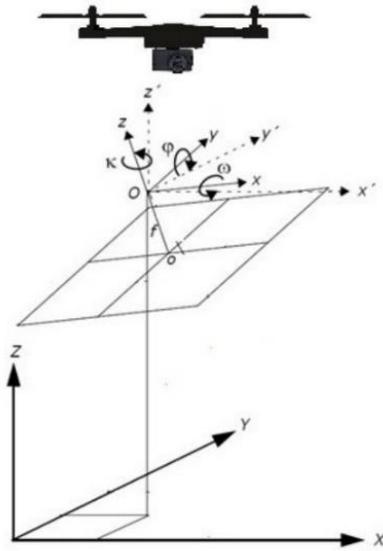


Figura 8 - Esquematização dos parâmetros extrínsecos da câmera (Alshawabkeh e Baik, 2023)

O planeamento de um levantamento fotogramétrico é essencial para assegurar a precisão dos resultados. A definição da *Ground Sample Distance* (GSD), que corresponde à distância entre pixels numa imagem real, é crucial para obter a resolução desejada. Um GSD mais pequeno resulta num modelo de alta resolução, sendo influenciado por fatores como a câmara utilizada e a altitude do drone em relação ao solo, conforme ilustrado na Figura 9 (Tariq *et al.*, 2017).

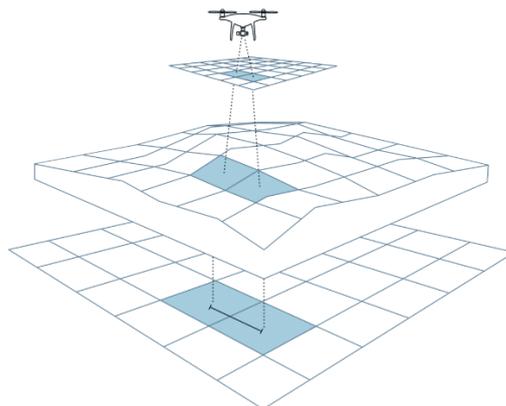


Figura 9 - Comportamento dos GSD com a variação da altura do UAV (Wingtra One, 2019)

Atualmente, UAVs são operados com *softwares* como *DJI Fly*, *DJI GS Pro* ou *Litchi Flight*, que permitem voos automatizados via *smartphone* ou *tablet*. Em estruturas complexas, recomenda-se dividir os levantamentos ou prolongar os voos para garantir uma boa captura dos pontos de referência e uma alta sobreposição de imagens. Para além disso, fatores ambientais, como horário do voo, vento, temperatura e altitude, também devem ser considerados, pois influenciam os resultados (Chatzistamatis et al., 2018; Tariq et al., 2017).

2.1.4 Fusão de dados

Após o processamento das nuvens de pontos provenientes de diferentes tecnologias, é necessário proceder à sua fusão, que se baseia na interseção entre os pontos de referência (coordenadas dos *pixels*) e os pontos resultantes da filtragem, incluindo a eliminação de *outliers* (Liu e Chou, 2023). Para tal, Bouziani, Chaaba e Ettarid (2021) propõem a aplicação do algoritmo iterativo ICP (*Iterative Closest Point*), criado em 1992 por Besl e McKay (1992). Este estabelece uma relação entre a distância de dois pontos correspondentes (Figura 10), presentes nas nuvens, podendo ser sintetizado nos seguintes passos:

- Pré-seleção dos pontos mais próximos entre as nuvens de pontos, recorrendo a uma subamostra, que deve ser aplicada de forma a reduzir o tempo e simplificar processos;
- Utilização do algoritmo *kd-tree* para encontrar os pontos comuns nas duas nuvens que, posteriormente, serão utilizados para o alinhamento;
- Exclusão de pares de pontos que vão introduzir erros e imprecisão na fusão, definindo assim critérios para exclusão desses pontos, como por exemplo, pontos distantes dos correspondentes na outra nuvem;
- Definição de funções métricas, de forma a minimizar erros na fusão e melhorar a precisão da mesma.

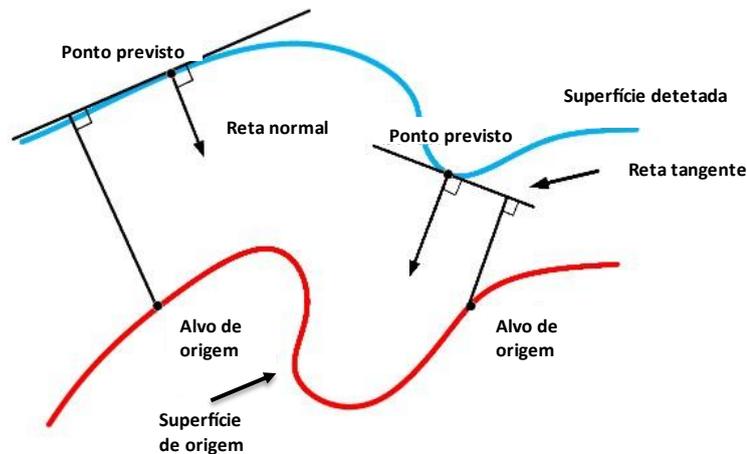


Figura 10 - Método gráfico para localizar o centro geométrico de duas nuvens de pontos (Cao *et al.*, 2023)

Em alternativa ao algoritmo ICP, Luhmann, Chizhova e Gorkovchuk (2020) apresentam a comparação *Cloud-to-mesh*, incorporada no software *RealityCapture*, tendo como referência a nuvem obtida pelo *Laser Scanner*, que por sua vez, pode ser utilizada como complemento, quando não é possível registar GCPs. Jiang *et al.* (2022), Bouziani, Chaaba e Ettarid (2021), Fabris, Fontana Granotto e Monego (2023), Jiang *et al.* (2022), e Son *et al.* (2020) complementam-se e propõem quatro métodos de validação dos resultados:

- *Cloud to Cloud (C2C)*;
- *Cloud to Mesh Distance (C2M)*;
- Aplicação do *plugin M3C2 (Multiscale Model to Model Cloud Comparison)*;
- Teste de segmentação (*Random Sample Consensus - RANSAC*).

2.2 Inteligência artificial

O DL é uma ramificação do *machine learning* (ML), amplamente utilizada, que permite a modelos computacionais com várias camadas de processamento realizar representações de dados em diferentes níveis de abstração, baseando-se na correlação entre *inputs* e *outputs* (Ferreira *et al.*, 2017; Zhong *et al.*, 2019). Por sua vez, o ML (subcampo da IA) é um método que trata de modelos com a capacidade de realizarem tarefas complexas, sem recorrer a instruções específicas, ou seja, o objetivo prende-se com a interpretação e classificação de novos dados

após a realização de um treino com um conjunto de dados com classes predefinidas (Bishop, 2006; Libralão et al., 2003). A IA pode ser compreendida como um sistema onde a inteligência é expressa através de máquinas de forma semelhante à inteligência humana (Silver *et al.*, 2017). A relação existente entre estes três conceitos está expressa na Figura 11.

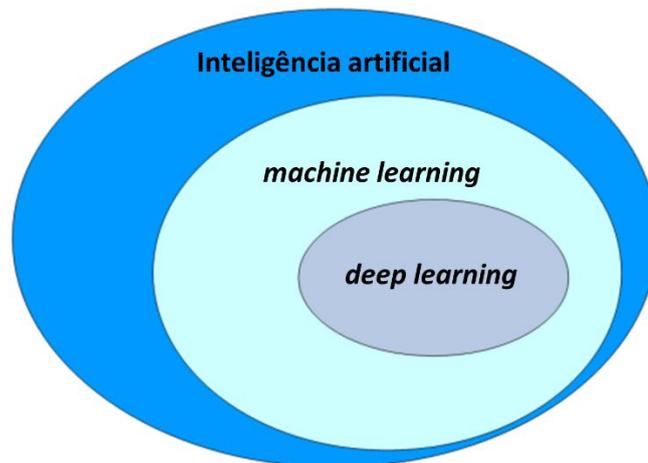


Figura 11 - Relação entre IA, ML e DL (Ye, Jin e Yun, 2019)

2.2.1 Redes Neurais Artificiais

Os primeiros estudos relacionados com as Redes Neurais remontam à década de 40, com o desenvolvimento do Modelo M-P por parte do psicólogo americano Mcculloch e do matemático Pitts. No entanto, a primeira rede neuronal artificial surge apenas em 1957 e ficou conhecida desde então como modelo *Perceptron*. Esta foi desenvolvida por Rosenblatt, com influência direta do trabalho que já havia sido realizado por Mcculloch e por Pitts (Wu e Feng, 2018).

As ANN podem ser compreendidas como sistemas computacionais que se assemelham às Redes Neurais biológicas e são consideradas um dos maiores avanços da inteligência artificial com influência numa grande variedade de áreas (Wu e Feng, 2018).

As ANNs funcionam através da comunicação entre um elevado número de nós (neurónios artificiais) que permitem o processamento e transmissão de informação para os neurónios da camada seguinte, tal como definido na seguinte Equação (1) e representado na Figura 12:

$$a_j^l = \sigma(\sum_k w_{jk}^l a_k^{l-1} + b_j^l). \quad (1)$$

onde l é a ordem da camada onde ocorre a computação; k e j referem-se ao número de unidades sinápticas da camada anterior e da atual, respectivamente; a trata-se da atividade ou

saída do respetivo neurónio; w é o peso sináptico, que corresponde à conexão entre dois neurónios e que determinam a força e a direção do sinal transmitido; b representa o parâmetro linear (bias); σ é a função de ativação, que é a função de saída específica desempenhada por cada neurónio (Santos, 2022; Bulsari, 1993).

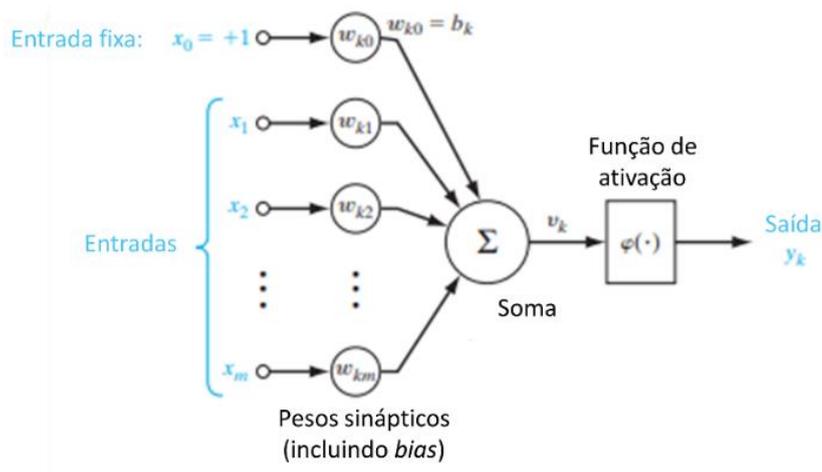


Figura 12 - Representação de um neurónio (Haykin, 2009)

Segundo Wu e Feng (2018), as redes neuronais artificiais caracterizam-se por terem quatro aspetos básicos, que se prendem com:

- **Não-linearidade:** Este tipo de redes apresenta um comportamento que é não-linear, relevante para a melhoria do desempenho e da tolerância à falha das mesmas;
- **Não-limitativa:** O comportamento de uma ANN resulta da interação entre vários neurónios conectados;
- **Não-qualitativa:** As ANN's conseguem auto adaptar-se dinamicamente durante o processamento de informações;
- **Não-convexidade:** A evolução de um sistema neuronal possibilita uma diversidade de resultados.

Habitualmente as ANNs encontram-se divididas em 3 camadas principais (Redes *Feedforward* Multicamadas - *Multilayer Feedforward Networks*), a camada de entrada, a camada oculta e a camada de saída, representadas na Figura 13 (Shukla e Abdelrahman, 2004). No que diz respeito à camada de entrada, a mesma é composta por neurónios, organizados de forma linear, que

recebem os dados (*inputs*) que serão processados pela rede. Por sua vez, a camada oculta é constituída por um grupo de neurónios que se encontram dispostos de forma paralela entre si. Por fim, a camada de saída que tem uma dimensão correspondente ao número de classes e que se baseia na função *softmax*, apresentada na Equação (2) (Santos, 2022):

$$f(y)_i = \frac{e^{z_i}}{\sum_{i=1}^t e^{y_i}} \quad (2)$$

onde $f(y)_i$ representa a saída da função softmax para a classe i ; e^{z_i} trata-se do exponencial do valor não normalizado (z_i), que por sua vez indica a saída do modelo para a classe i ; e $\sum_{i=1}^t e^{y_i}$ consiste na soma das exponenciais de todos os valores não normalizados, garantindo que a saída seja uma distribuição de probabilidade.

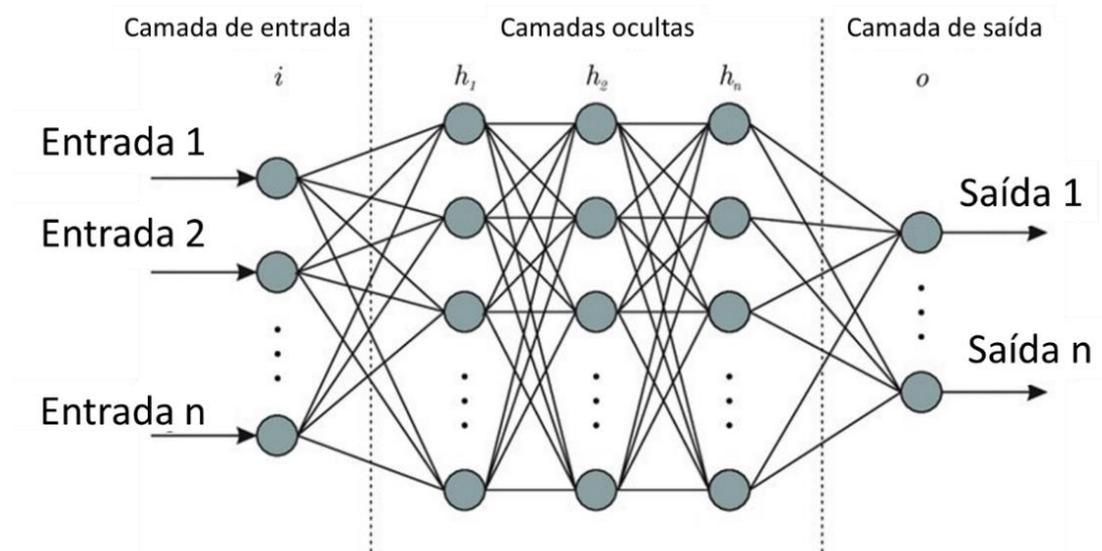


Figura 13 - Arquitetura *Feedforward* Multicamadas de uma Rede Neuronal Artificial (Bre, Gimenez e Fachinotti, 2018)

Apesar de a arquitetura base acima definida ser a mais comum, este tipo de redes apresenta ainda mais dois tipos de arquiteturas de ANNs, tal como apresentado por (Haykin, 2009):

- **Redes *Feedforward* de Camada Única (*Single-Layer Feedforward Network*):** trata-se da arquitetura mais simples. Como o nome indica, esta é constituída por uma camada de entrada de nós que projeta de forma direta um sinal para a camada de saída, sendo que neste tipo de rede a informação nunca propaga em sentido contrário (Figura 14). A camada de entrada não é contabilizada, visto que não existe qualquer tipo de computação na mesma, daí estas Redes Neurais Artificiais serem designadas por “camada única”;

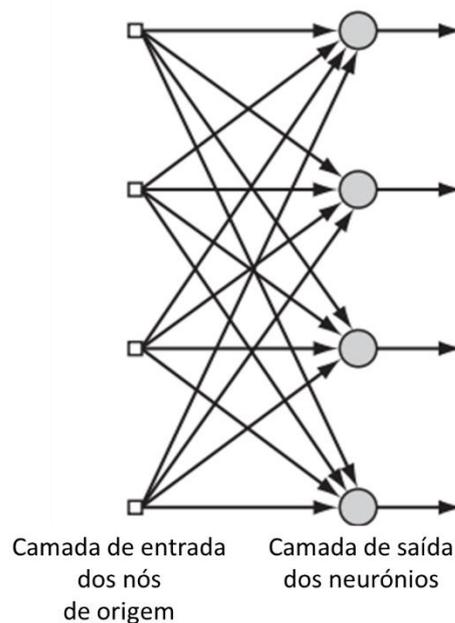


Figura 14 - Rede *Feedforward* de Camada Única (Haykin, 2009)

- **Redes Recorrentes (*Recurrent Networks*):** este tipo de redes caracteriza-se essencialmente pelo facto de terem pelo menos um ciclo de realimentação, ou seja, trata-se de uma ANN composta apenas por uma camada de neurónios onde o sinal de saída origina um sinal de entrada, tal como apresentado na Figura 15. A existência destes ciclos de realimentação tem um efeito impactante tanto na aprendizagem como no desempenho da rede, incluindo *unit-time delay operators* levando a um comportamento não-linear por parte da mesma.

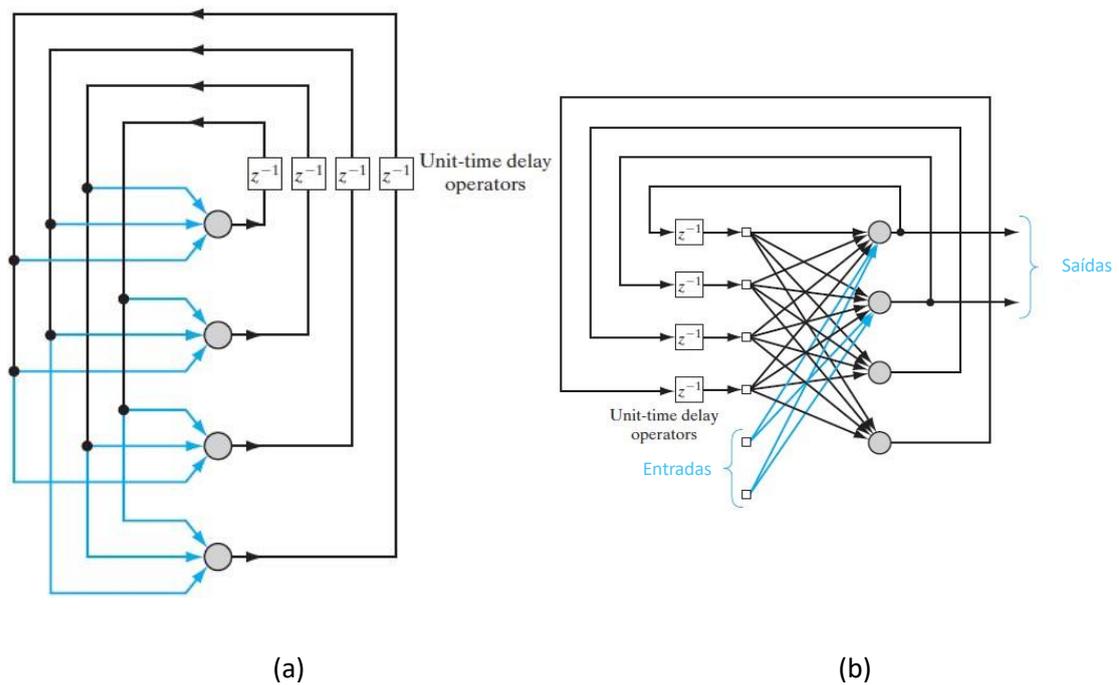


Figura 15 - Rede Recorrente sem (a) e com (b) *loops* de *auto-feedback* e neurónios ocultos (Haykin, 2009)

2.2.2 Redes Neurais Convolucionais

Em 1990, Lecun et al. (1990) apresentaram as Redes Neurais Convolucionais que são redes multicamadas inspiradas nas ANNs. Este tipo de rede é projetado para receber matrizes em vez dos habituais vetores, podendo lidar com dados temporais e espaciais. Contudo, na maioria dos casos, as CNNs são utilizadas principalmente para detecção e classificação de imagens (Oliveira, 2022; Silva, 2024). Relativamente à arquitetura deste tipo de rede, a mesma é habitualmente fracionada em região de extração e classificação, sendo a primeira constituída por uma ou várias camadas convolucionais, a que se seguem camadas de *pooling* que se ligam a camadas totalmente conectadas (*fully-connected*) na etapa da classificação (Figura 16). Esta organização permite aproveitar a estrutura 2D de uma imagem de entrada e, dado que todas as saídas na camada de convolução partilham o mesmo peso, resulta numa diminuição tanto do número de parâmetros quanto do custo computacional, em comparação com as ANNs (Ferreira *et al.*, 2017).

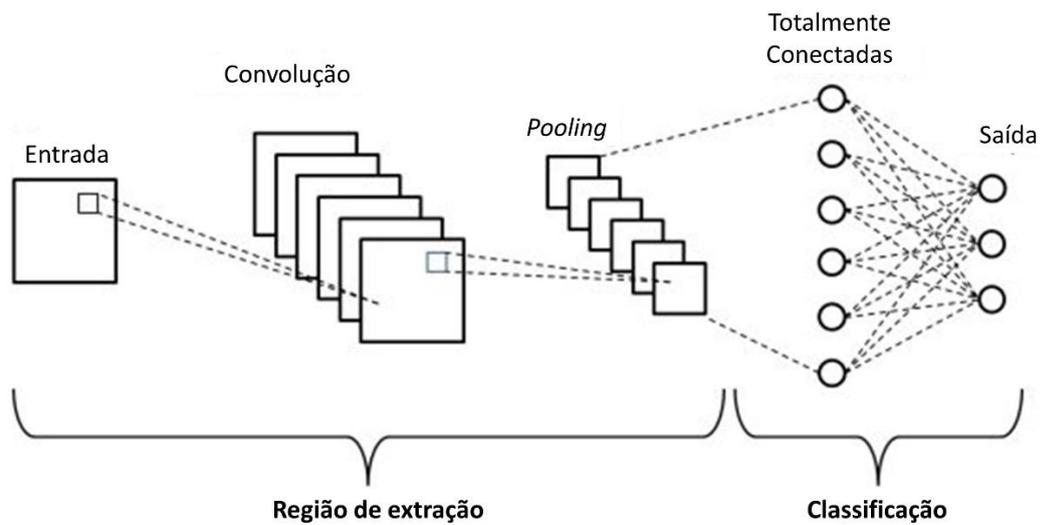


Figura 16 - Arquitetura de uma CNN (Phung e Rhee, 2018)

2.2.2.1 Camada convolucional

A primeira camada das redes CNNs consiste no bloco de construção central das mesmas e caracteriza-se por assegurar a obtenção das características intrínsecas de uma imagem (Santos, 2022; Ferreira et al., 2017). A operação de convolução (Figura 17) baseia-se na Equação (3):

$$Y = w \times X \tag{3}$$

onde uma matriz (X), que representa uma imagem, é multiplicada por uma matriz quadrada de pesos sinápticos (w) ou um filtro ao qual se dá o nome *kernel*.

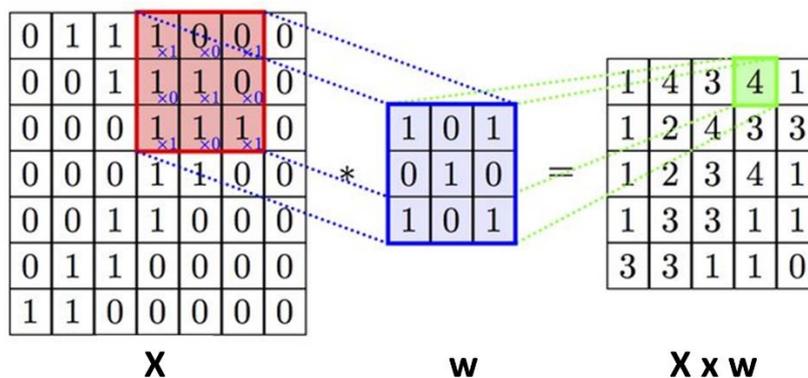


Figura 17 - Exemplo de processo de convolução 2D (Vieira et al., 2021)

Esta camada é constituída por vários *kernels*, com valores iniciais gerados de forma aleatória que se vão ajustando com recurso ao método de *backpropagation*, sendo que, segundo Araújo, Carneiro e Silva (2017), “quanto maior o número de filtros maior o número de características extraídas, porém a complexidade computacional, relativa ao tempo e ao uso de memória, também será maior”. Durante estas operações são definidos hiperparâmetros, através da Equação (4) (Silva, 2024):

$$D_{saída} = \frac{D_{entrada} - D_{kernel} + 2 \times D_{padding}}{D_{stride}} + 1 \quad (4)$$

onde $D_{entrada}$ corresponde à dimensão do *input*, D_{kernel} é a quantidade e o tamanho dos filtros na camada, D_{stride} representa o número de pixels em deslocamento e $D_{padding}$ ainda uma técnica de adição de zeros (*padding*).

De uma forma mais prática, tal como explica Wu (2017), ao processar uma imagem, um computador recebe uma matriz de pixels com valores que vão de 0 a 255. A entrada é tridimensional composta por altura, largura e profundidade, que é determinada pela quantidade de canais de cor (RGB) ou por duas imagens em tons de cinza.

2.2.2.2 Camada de *pooling*

Após a camada de convolução, surge a camada de *pooling*, cujo objetivo é reduzir a quantidade de parâmetros e a dimensão da rede, sem que as informações mais relevantes sejam perdidas, evitando assim o efeito de *overfitting* (Santos, 2022; Ferreira et al., 2017; Silva, 2024). Existem diversos tipos de operações *pooling* (máximo - *max pooling*, média- *mean pooling*, soma, entre outros), representadas na Figura 18, no entanto o *max pooling* é o mais utilizado devido à eficácia do mesmo (Silva, 2024).

Esta técnica, representada na Equação (5), permite a identificação do valor mais alto dos pixels numa determinada área, delimitada pelo *kernel*.

$$Y = \max_{i,j}(x_{i,j}) \quad (5)$$

onde $x_{i,j}$ representa a matriz de entrada na janela de mapeamento. De uma forma mais explícita, Santos (2022), explica: uma entrada $L_{entrada} \times H_{entrada} \times P_{entrada}$ como dimensões, ao realizar a operação de *pooling*, o resultado final será $L_{saída} \times H_{saída} \times P_{saída}$, cujas dimensões são calculadas através das Equações (6)(7)(8):

$$L_{saída} = \frac{(L_{entrada} - M)}{D_{stride}} + 1 \quad (6)$$

$$H_{saída} = \frac{(H_{entrada} - M)}{D_{stride}} + 1 \quad (7)$$

$$P_{saída} = P_{entrada} \quad (8)$$

onde M representa a dimensão da janela de mapeamento e D_{stride} consiste, tal como o nome indica, na operação de *stride*. A Equação (8) demonstra que não existe qualquer tipo de alteração no que à profundidade da camada diz respeito, isso apenas acontece nas dimensões do plano (Santos, 2022).

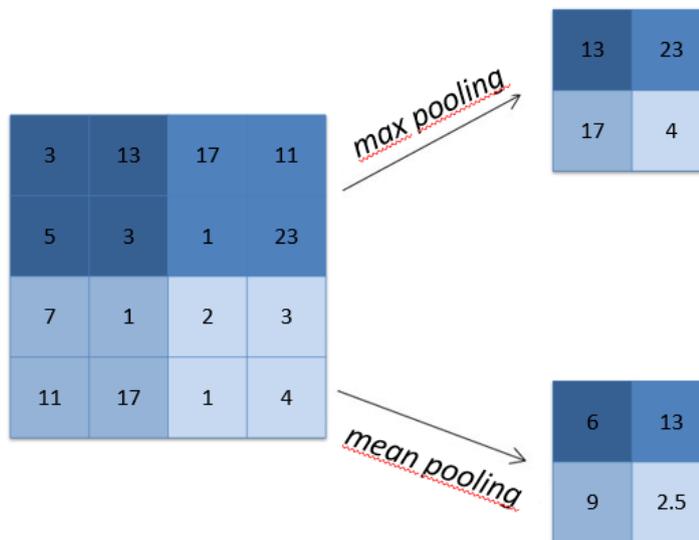


Figura 18 - Exemplo de *max* e *mean pooling* (Aljaafari, 2018)

2.2.2.3 Camada *fully-connected* (FC)

Esta camada que sucede à de *pooling* e antecede a camada de saída, consiste numa Rede Neuronal Artificial Multicamadas, em que os neurónios estão totalmente conectados, como o nome indica. Isto significa que todos os neurónios da camada anterior, estão ligados a todos os neurónios da camada seguinte. O objetivo desta camada é utilizar as características da imagem de entrada, obtidas nas duas camadas anteriores, para classificar a imagem de acordo com as

classes presentes na base de dados de treino (Ferreira *et al.*, 2017). Para que o procedimento seja realizado da forma correta, tem de ocorrer o *flattening*, que consiste na conversão das matrizes de características provenientes da camada de *pooling* num vetor coluna, tal como representado na Figura 19 (Silva, 2024).

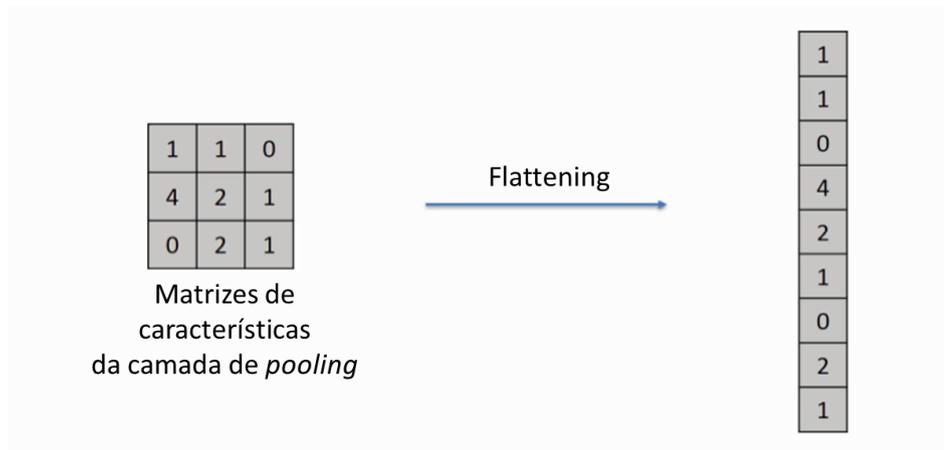


Figura 19 - Exemplo de operação de *flattening* (SuperDataScience Team, 2018)

Posto isto, o objetivo passa depois por aprimorar a capacidade de previsão das classes. Tal como referido anteriormente, as ANN necessitam de uma função de ativação, que no caso das CNN, corresponde habitualmente à função de unidade linear retificada (ReLU). Esta função tem como finalidade o mapeamento da saída para zero quando o valor de entrada é negativo e no caso de ser positivo, remete-o para a saída, sem qualquer tipo de alteração. Para além disso, a ReLU (Equação (9)) dificulta o surgimento de *overfitting* em camadas de convolução extensas (Silva, 2024).

$$f_{ReLU}(x) = (0, x) \quad (9)$$

2.2.2.4 Arquiteturas

Tal como apresentado nas secções anteriores, a combinação entre as camadas de convolução, as camadas de *pooling*, as funções de ativação e classificação constitui a arquitetura de uma CNN. Todavia, existe uma multiplicidade de configurações que alteram o custo computacional das mesmas (Santos, 2022). As áreas de aplicação deste tipo de redes têm aumentado continuamente, com o intuito de resolver vários entraves, sendo que as principais abordagens foram (Ferreira *et al.*, 2017):

- **LeNet-5:** uma das primordiais e mais relevantes redes, é constituída por 7 camadas que servem para perceber dígitos escritos à mão (Lecun *et al.*, 1990);
- **AlexNet:** rede apresentada por Alex Krizhevsky, em 2012, tratando-se de uma versão melhorada e mais profunda da anterior (Krizhevsky, Sutskever e Hinton, 2017);
- **ZFNet:** consiste num aperfeiçoamento da AlexNet, proposta por Matthew Zeiler e Rob Fergus, com recurso a um ajuste de hiperparâmetros, mais concretamente os tamanhos dos filtros nas camadas convolucionais (Zeiler e Fergus, 2013);
- **VGGNet:** desenvolvida pela equipa da Visual Geometry Group (VGG) da Universidade de Oxford. O principal intuito da VGGNet foi mostrar que a profundidade de uma rede é uma característica essencial para bom desempenho (Simonyan e Zisserman, 2014);
- **GoogLeNet:** introduziu os “*Inception modules*” que diminuem, substancialmente, o número de parâmetros na rede (Szegedy *et al.*, 2015);
- **ResNet:** a Residual Network, ou ResNet, foi proposta em 2015 e estabeleceu a ideia de conexões residuais (ou saltos) de modo a resolver o problema de desaparecimento do gradiente em redes muito profundas (He *et al.*, 2016).

No campo da classificação de imagens, um dos principais benefícios das CNNs, comparativamente com as ANNs, resulta da aptidão de generalizar modelos com mais facilidade, diminuindo a hipótese de acontecer *overfitting*. Este problema pode surgir através do excesso de parâmetros na rede, falta de versatilidade dos dados de entrada, demasiadas iterações ou falta de regularização. O *overfitting* consiste na absorção de ruídos por parte do modelo, levando a uma redução na capacidade de generalização dos dados de validação e de teste (Santos, 2022). Os gráficos da Figura 20, apresentam casos de *underfitting* em que o a arquitetura da rede é pouco complexa ou o modelo está mal treinado, *optimal-fitting*, tal como o nome indica, o modelo tem a complexidade ideal, com a estrutura e os hiperparâmetros otimizados. Para além disso, no *overfitting* o modelo é extremamente treinado e complexo, no entanto memoriza apenas os dados de treino, perdendo a capacidade de generalização.

De forma a solucionar o *overfitting*, divide-se o *dataset* em três grupos: treino, validação e teste. O treino tem como intuito a otimização do modelo a treinar, ao contrário da validação que corresponde a uma avaliação no decorrer do treino, periodicamente, com o objetivo de

explorar se existe uma grande diferença entre as métricas (Santos, 2022).

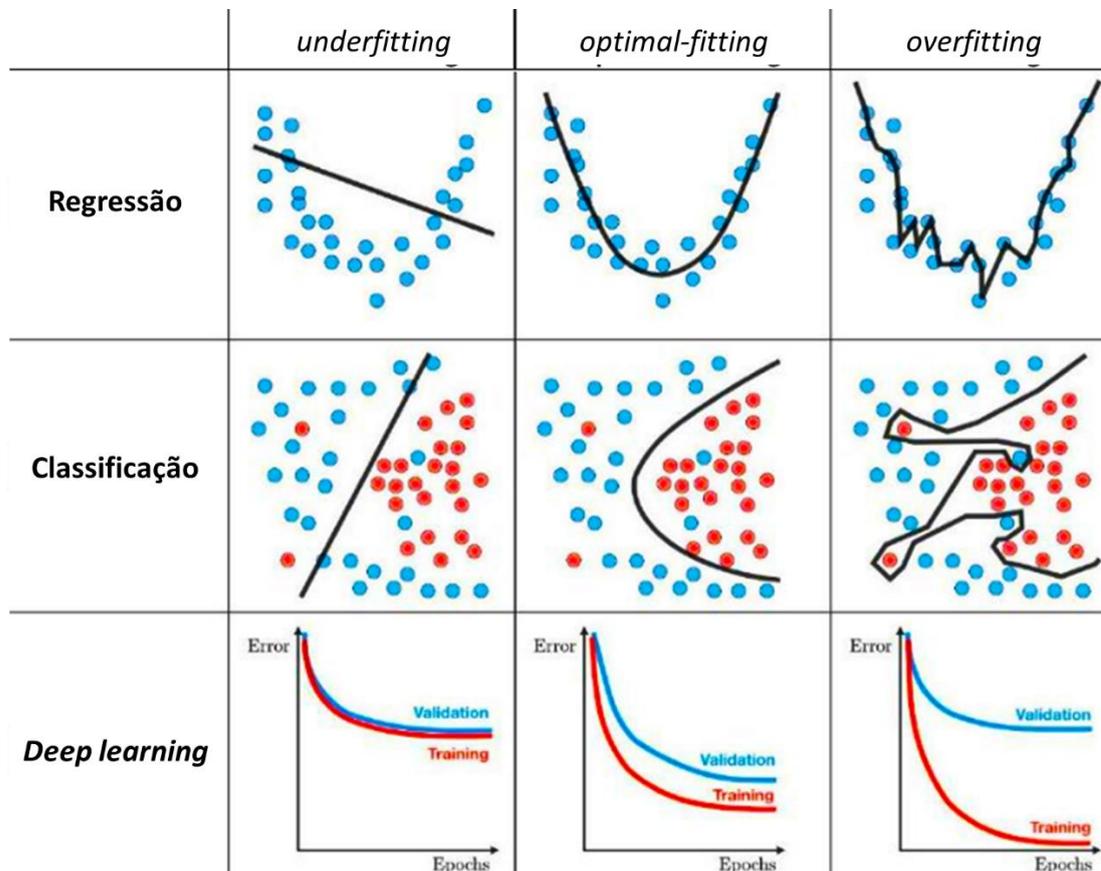


Figura 20 - Exemplos *underfitting*, *optimal-fitting* e *overfitting* em Regressão, Classificação e DL
(Towards Data Science, 2021)

2.2.3 Redes Neurais Convolucionais por Região

Atualmente, existem vários algoritmos que compõem as famílias das *Region-based Convolutional Neural Network*, entre os quais, R-CNN, *Fast R-CNN*, *Faster R-CNN*, YOLO e *Mask R-CNN*, cuja evolução das mesmas está representada na Figura 21. Estas redes possuem não apenas a capacidade de classificar, mas também de detetar objetos numa imagem através de regiões de proposição, sendo que as duas últimas implementam a segmentação por instâncias (Santos, 2022). Por essa mesma razão, estas últimas serão adotadas e exploradas nas aplicações práticas deste trabalho (Capítulo 4).

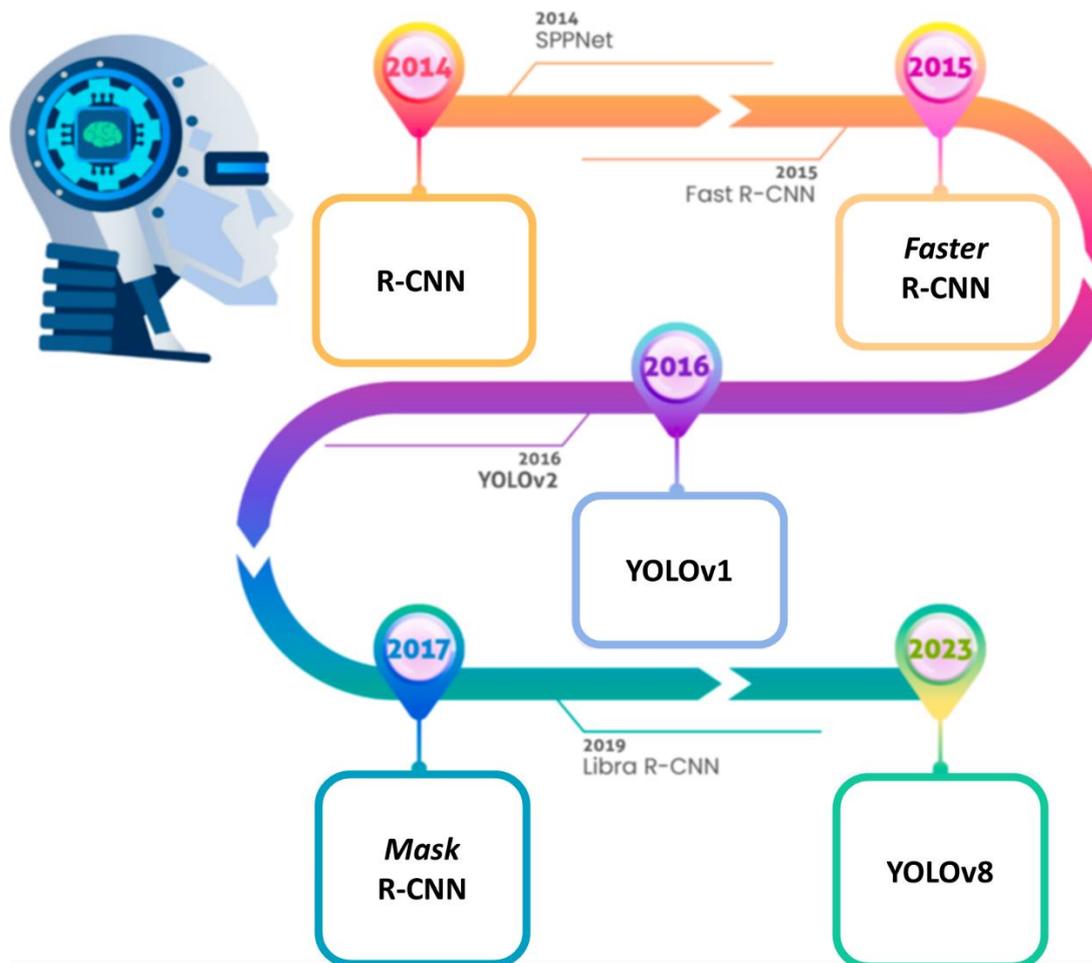


Figura 21 - Evolução da detecção de objetos no DL (adaptado de Hosseinzadeh et al., 2024)

Para compreender a arquitetura da YOLO e da *Mask R-CNN*, é necessário entender primeiro as bases da família de arquiteturas R-CNN (Oliveira, 2022).

2.2.3.1 R-CNN

O algoritmo R-CNN proposto por Girshick et al. (2013), é constituído por três módulos, tal como representado na Figura 22:

- **Proposição de Rols (*Regions of Interest*)** - trata-se de um algoritmo de busca seletiva, isto é, gera vários objetos por imagem, que se agrupam consoante as parecências em relação a cor, textura, tamanho e compatibilidade, ou seja, de forma não supervisionada (Uijlings et al., 2013);
- **Extração de características** - constituído por uma Rede Neuronal Convolucional, sendo

que a arquitetura da mesma pode ser AlexNet ou ZFnet, mas sem a *softmax* (última camada de classificação). Esta CNN recolhe o mapa de características (*feature map*) de cada região e envia-os, separadamente, para o terceiro módulo (Santos, 2022; Oliveira, 2022).

- **Classificação** - surge através de um conjunto de máquinas de vetores suporte (*Support Vector Machine* - SVM) que tenta maximizar a distância entre as diferentes classes, descobrindo diferenças entre as mesmas (Santos, 2022).

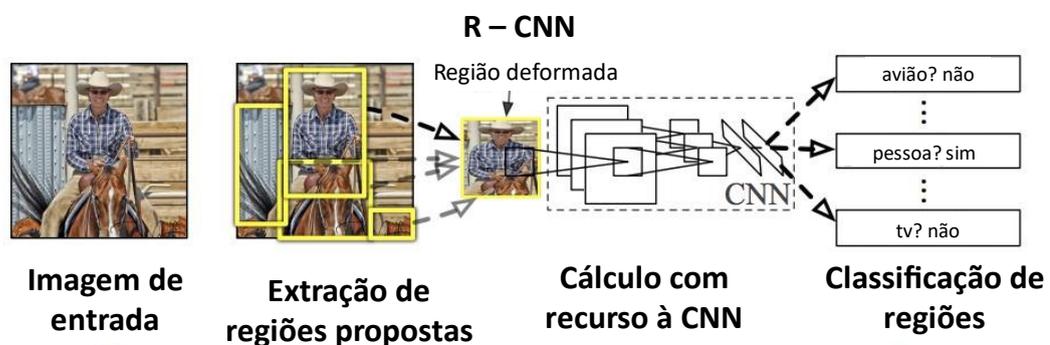


Figura 22 - Arquitetura da rede R-CNN (Girshick *et al.*, 2013)

2.2.3.2 Fast R-CNN

Tal como referido por Oliveira (2022), o modelo R-CNN exigia muito tempo de processamento, uma vez que cada ROI precisava passar pela CNN e em seguida, pela SVM. Para resolver esse problema, em 2015, Girshick (2015) implementou uma alteração no modelo, ao integrar o *pooling* numa região de interesse, possibilitando assim que a CNN permita a introdução de imagens de tamanhos arbitrários como entrada, não sendo assim necessário realizar operações de corte e distorção. Neste caso, a arquitetura de CNN selecionada foi a VGG-16 que obtém e transfere uma projeção das regiões de interesse do mapa de características para a camada de ROI *pooling* (Oliveira, 2022). Esta camada, por sua vez, faz um *maxpooling* e gera um vetor de tamanho fixo, que é classificado com recurso a uma função *softmax* e localizado na imagem correspondente com recurso a uma regressão linear, posteriormente a um processamento de uma rede *Fully Connected*. As ROI, tal como na R-CNN, são obtidas através de busca seletiva, mas no caso da *Fast R-CNN*, a rede pode ser totalmente treinada em conjunto, fazendo com que o treino seja 9 vezes mais rápido e a inferência cerca de 213 vezes mais rápida (Santos, 2022), levando assim à designação de *Fast R-CNN* (Figura 23).

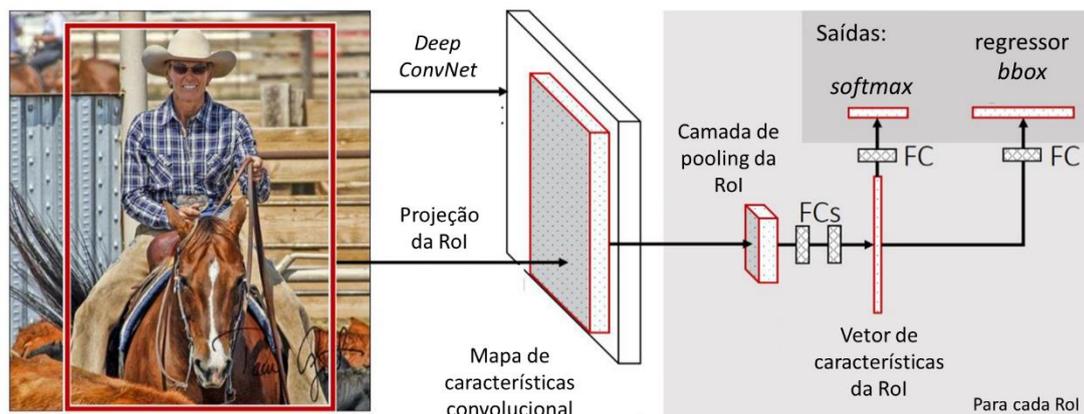


Figura 23 - Arquitetura da rede *Fast R-CNN* (Girshick, 2015)

2.2.3.3 *Faster R-CNN*

A *Faster Region-based Convolutional Neural Network (Faster R-CNN)* foi um aprimoramento às redes abordadas anteriormente, proposto por Ren et al. (2015), sendo uma rede capaz de realizar previsões de máscaras de segmentação ou Regiões de Interesse (Rois) de uma imagem pixel a pixel (Liu e Chou, 2023). Este modelo é dividido em duas etapas, com a primeira sendo a Region Proposal Network (RPN), que substitui o algoritmo anterior de busca seletiva para o desenvolvimento de caixas delimitadoras de objetos candidatos (Xu *et al.*, 2021). A RPN trata-se de um classificador, composto por uma CNN (podendo ser uma ZFnet ou uma VGG-16), como é possível observar na Figura 24 (a). Utilizando uma janela deslizante e âncoras centradas (Rois hipotéticos predefinidos) em cada pixel, a RPN examina o mapa de características. De uma maneira mais detalhada, o funcionamento da RPN, composta por uma camada convolucional e duas camadas FCs, envolve o mapeamento da parcela para um vetor de menor espaço vetorial, que é direcionado para as duas camadas totalmente conectadas, que por sua vez, são responsáveis pela quantidade de âncoras a colocar.

No caso de uma série de K âncoras, com diferentes escalas, existem $2K$ *scores* que consistem na probabilidade da existência ou não de um objeto, obtido através de uma função *softmax*. Por fim, os $4K$ *scores*, conseguidos com recurso a uma regressão linear, caracterizam-se como as coordenadas x e y das caixas delimitadoras (*bounding-boxes*) dos objetos, para além das coordenadas parametrizadas (altura e comprimento), que são obtidas de acordo com as Equações (10), (11), (12) e (13) e tal como demonstrado na Figura 24 (b) (Santos, 2022; Oliveira, 2022).

$$t_x = \frac{(x - x_a)}{w_a} \quad (10)$$

$$t_y = \frac{(y - y_a)}{h_a} \quad (11)$$

$$t_h = \log\left(\frac{h}{h_a}\right) \quad (12)$$

$$t_w = \log\left(\frac{w}{w_a}\right) \quad (13)$$

onde a t_x e t_y se baseiam na parametrização do ajuste das coordenadas x e y , respetivamente, x , y , x_a e y_a representam as coordenadas inferidas das âncoras; já h , w , h_a e w_a correspondem à altura e largura inferidas e ainda à altura e largura da âncora.

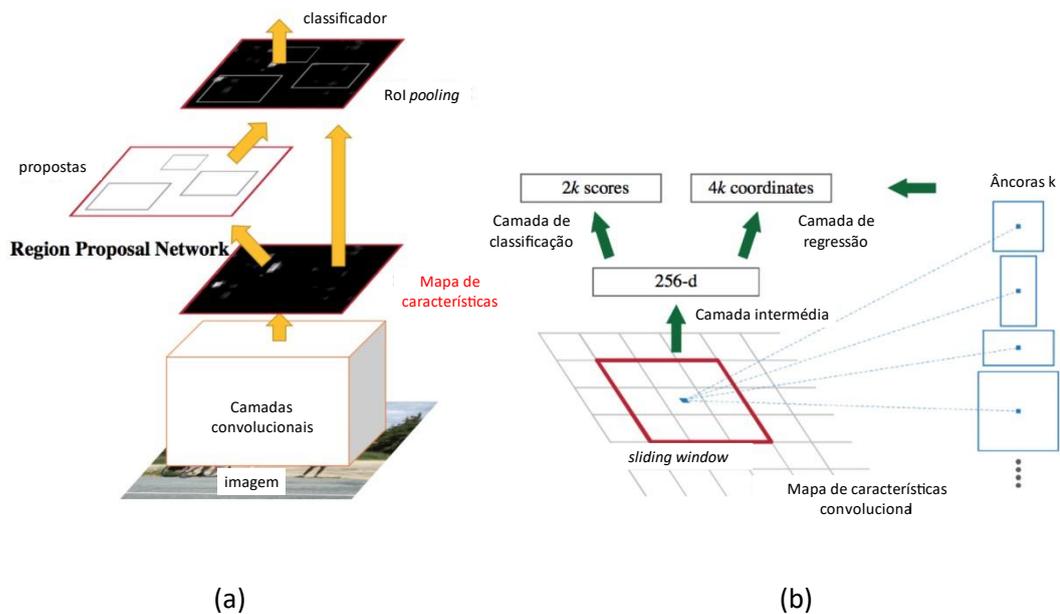


Figura 24 - Arquitetura *Faster R-CNN* (a) e funcionamento da RPN (b)

Estas âncoras, tal como referido por Santos (2022), são “dimensionadas de acordo com um número “n” de razões de proporcionalidades (altura x largura) e um número “m” que representa o seu tamanho no formato quadrangular”. Por exemplo, caso um mapa de características tenha $W \times H$ como dimensões, serão geradas cerca de $W \times H \times k$ âncoras.

Uma particularidade da *Region Proposal Network* é o facto da mesma ser treinada em

simultâneo com o detetor de classificação da rede *Fast R-CNN*, segunda etapa da arquitetura *Faster R-CNN* abordada anteriormente, proporcionando assim um procedimento mais rápido e integrado (He *et al.*, 2017).

2.2.3.4 YOLO

A criação da família YOLO (*You Only Look Once*) em 2016, com a *YOLOv1* proposta por Redmon *et al.* (2015), representou um marco no domínio das CNNs. A partir dessa versão inicial, surgiram diversas iterações e melhorias, conforme ilustrado na Figura 25, que não destaca a versão 7 e , devido ao pouco impacto e novidades associadas às mesmas. Uma das principais características que distingue as redes YOLO das demais é o facto de que a rede realiza a predição a partir de uma única passagem da imagem de entrada, o que a torna especialmente adequada para aplicações em tempo real, permitindo a deteção de objetos a uma taxa de até 30 FPS (*frames per second*). O modelo redefine a deteção de objetos como um problema de regressão, utilizando apenas os pixels da imagem para avaliar a probabilidade das classes, bem como as coordenadas e dimensões das predições (Redmon *et al.*, 2015).

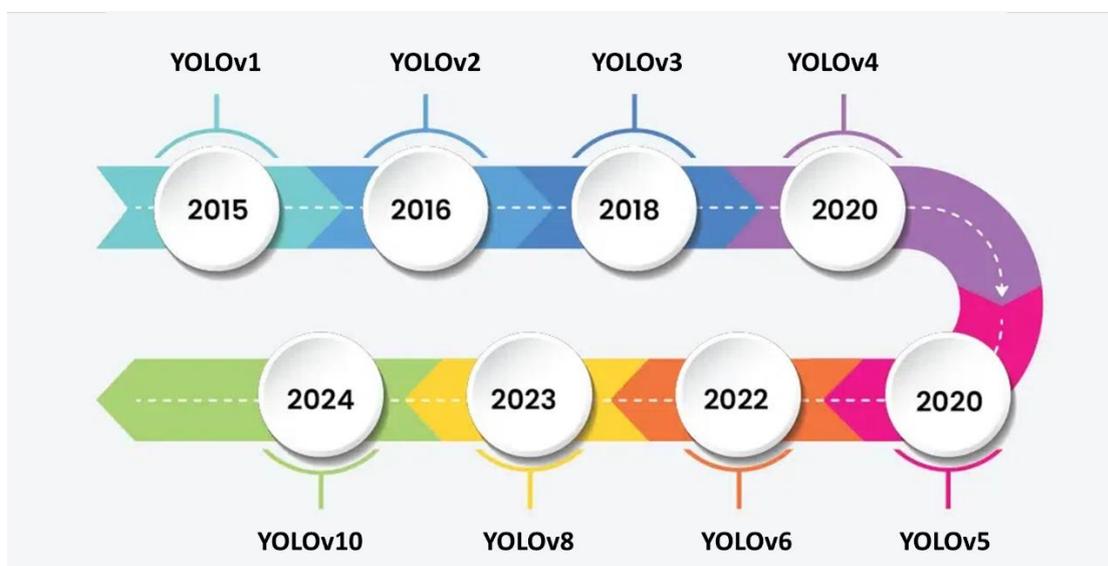


Figura 25 - Evolução da família YOLO (GeeksforGeeks, 2024)

A arquitetura base da família YOLO, classificada como um detetor de um estágio, encontra-se dividida em três fases, tal como apresentado na Figura 26. A primeira fase, denominada *Backbone* consiste numa rede neuronal profunda constituída por várias camadas de convolução, sendo que tem como principal intuito a extração de atributos essenciais. A segunda

fase, chamada *Neck*, tem como objetivo a obtenção de diversos mapas de características, enquanto a última fase, conhecida como *Head* ou *Dense Prediction*, tal como o nome indica, gera um vetor com as informações de cada *bounding box*, realizando assim as previsões necessárias (Araújo, 2022).

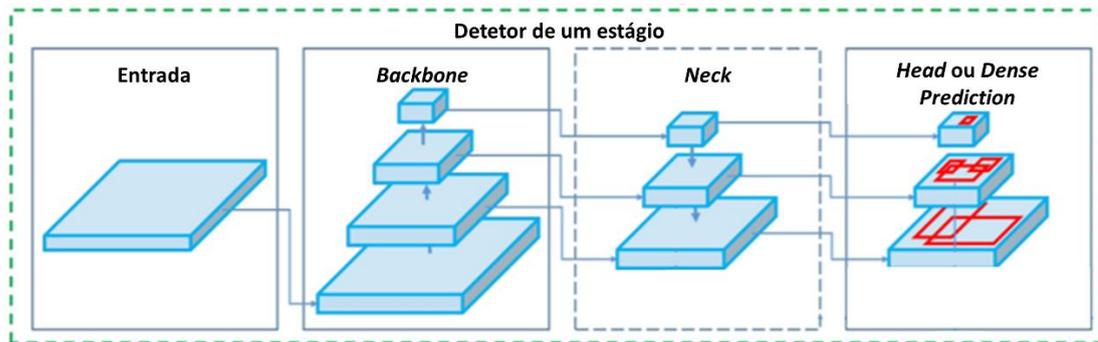


Figura 26 - Arquitetura base da YOLO (Bochkovskiy, Wang e Liao, 2020)

O funcionamento da YOLO consiste em dividir as imagens de entrada, numa espécie de grelha de células $S \times S$, sendo que o tamanho dessas células pode variar consoante a versão da YOLO usada, existindo a possibilidade de tamanhos como 3×3 , 5×5 e 19×19 . No interior de cada uma delas, são atribuídas *bounding boxes* de acordo com as coordenadas centrais (b_x, b_y), a altura (b_h) e a largura (b_w) das mesmas, a probabilidade (P_c) de existir um objeto e ainda a classe (C) atribuída a esse mesmo objeto, tal como representado na Figura 27. De modo a realizar uma filtragem de todos os objetos encontrados, o algoritmo estabelece um limite de confiança, desconsiderando os que se encontram abaixo do mesmo (Pedoeem, Huang e Chen, 2018; Araújo, 2022).

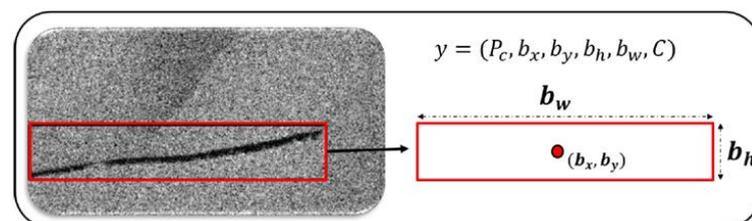


Figura 27 - Esquema representativo do funcionamento da detecção de um objeto através de *bounding boxes* (Silva, 2024)

Uma célula apenas pode conter uma classe detetada, mesmo que na mesma existam várias *bounding boxes*, isto porque se o centro de um objeto se encontrar no interior de uma célula,

a mesma é encarregue de detetar esse mesmo objeto. Supondo que uma célula contém mais do que uma deteção, surge o conceito de mapa de probabilidade de classes com o intuito de resolver esse entrave. Como só pode ser detetado um objeto, as demais caixas delimitadoras sofrem de um processo de supressão, que serve para eliminar objetos duplicados, tal como demonstrado na Figura 28 (Araújo, 2022).

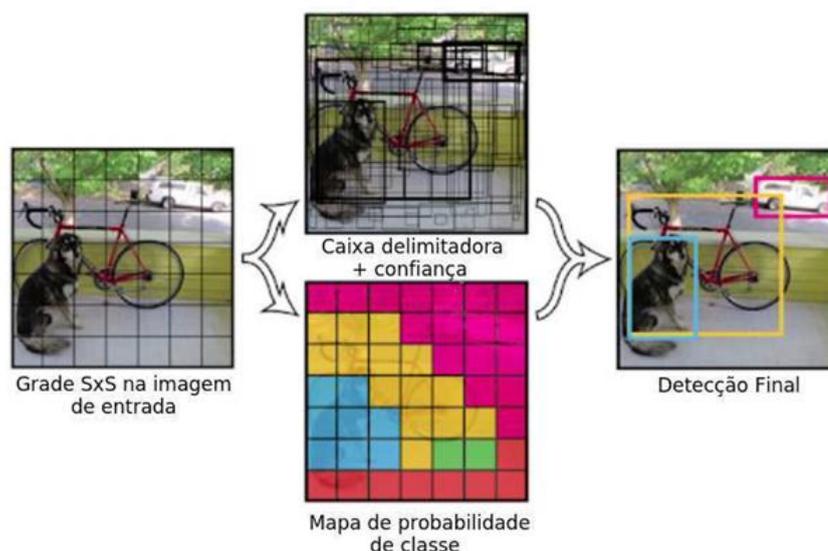


Figura 28 - Exemplo de *object detection* recorrendo à YOLO (Redmon *et al.*, 2015)

2.2.3.5 Mask R-CNN

A *Mask R-CNN* é uma evolução da *Faster R-CNN*, que incorpora um ramo adicional com a capacidade de realizar a IS classificadas em cada RoI, ao mesmo tempo que efetua a classificação e regressão por meio de outro ramo. Este aprimoramento foi possível através da adição do módulo *RoIAlign*, como ilustrado na Figura 29. O *RoIAlign* introduz máscaras binárias envolvendo os objetos de interesse, através de uma interpolação bilinear, não considerando as bordas das *bounding boxes*. Além disso, introduz-se o conceito de *Fully Convolutional Network* (FCN), que é uma CNN composta exclusivamente por camadas convolucionais, permitindo, assim como a YOLO, a classificação "*pixel a pixel*". Ao contrário da YOLO, este algoritmo é identificado como um detetor de dois estágios (He *et al.*, 2017; Chowdhury e Moon, 2023).

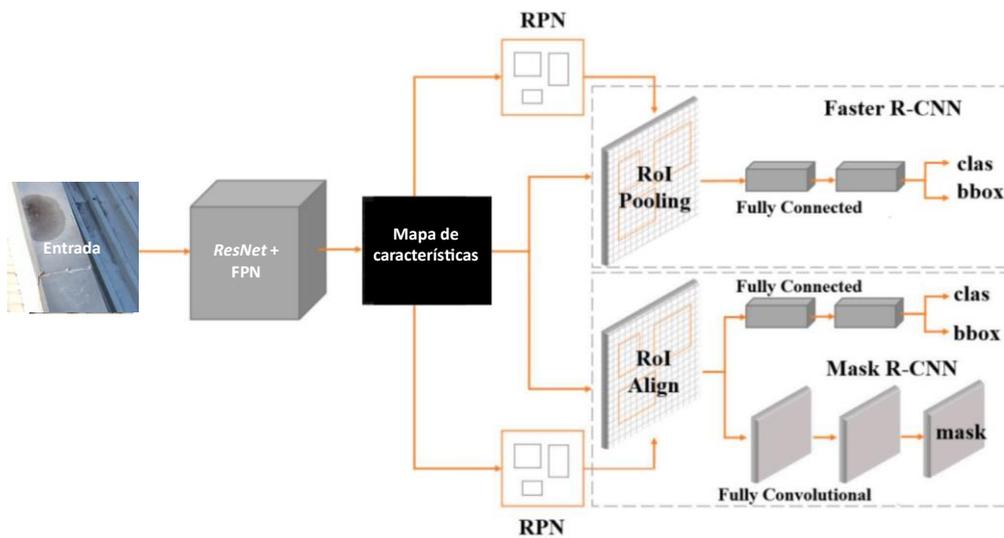


Figura 29 - Arquitetura de uma *Mask R-CNN* comparada com a de uma *Faster R-CNN* (Xu *et al.*, 2022)

A *Mask R-CNN* destaca-se das demais redes por utilizar a *ResNet-50* ou *ResNet-101* como Backbone, que são redes dedicadas à extração de características. Neste modelo, a *Region Proposal Network* e a camada de *RoI Align* recebem informações de várias camadas das operações de convolução, em contraste com a *RoI Pooling*, sendo esta abordagem denominada *Feature Pyramid Network* (FPN), conforme ilustrado na Figura 30 (Lin *et al.*, 2016). De uma forma mais aprofundada, a arquitetura de uma *Mask R-CNN* é constituída por (Santos, 2022):

- **Rede Backbone-** Tal como apresentado na Figura 30, a região de extração de características divide-se em 4 camadas de *feature maps* (res2, res3, res4 e res5) que representam a informação semântica do objeto;
- **Feature Pyramid Network (FPN)-** Tem como principal intuito a obtenção de amostras de diferentes níveis de convolução enquanto a imagem é processada. Para isso, existem conexões laterais, que contribuem para a IS de instâncias pequenas;
- **Region Proposal Network (RPN)-** Opera sobre o *feature map* criado pelas duas camadas anteriores, propondo *bounding boxes*;
- **Region of Interest (RoI) Align-** Com recurso a interpolação bilinear, os objetos são localizados de forma precisa, e extraídos do mapa de regiões propostas criado pela RPN.
- **Region of Interest (RoI) Heads-** O principal objetivo desta camada é gerar as máscaras,

através de uma FCN, simultaneamente à classificação e detecção realizadas pela *Fully Connected Network*.

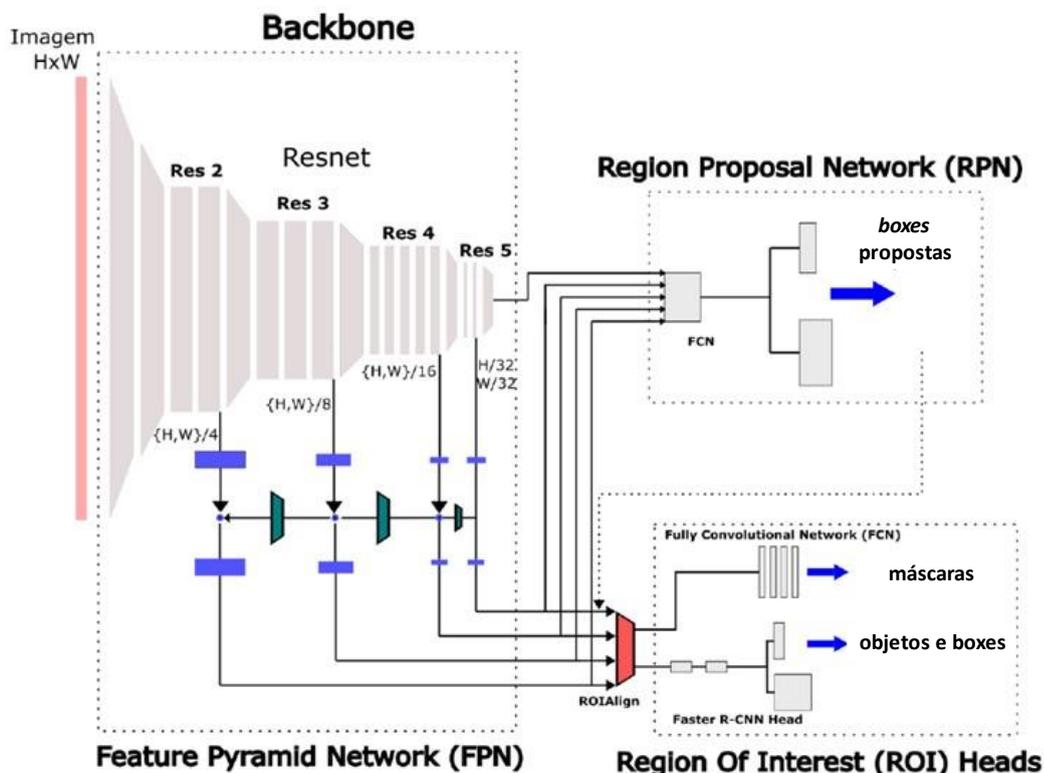


Figura 30 - Arquitetura de uma *Mask R-CNN*, utilizando uma *Resnet-50* como *Backbone* (Santos, 2022)

À semelhança da *Faster R-CNN*, a *Mask R-CNN* consiste num modelo simples de implementar e treinar, contribuindo para uma variedade de *designs* de arquiteturas flexíveis. Para além disso, o *RoIAlign* constitui apenas um pequeno *overhead* computacional, não tendo muito peso em termos de velocidade e custo computacional (He et al, 2018). A *Mask R-CNN* é treinada utilizando funções de perda que são geradas nas *RoI Heads*, após a realização das predições das classes, detecções com caixas delimitadoras e máscaras (Santos, 2022).

2.2.3.6 Métricas

As métricas de avaliação da classificação visam quantificar a percentagem de amostras classificadas que correspondem corretamente aos seus rótulos. Relativamente à qualidade da detecção e segmentação de objetos em imagens, afere-se a sobreposição esperada através dos rótulos, calculando as mesmas através de precisões. Essas precisões fundamentam-se na sobreposição entre a instância rotulada previamente e a inferida, com recurso à Equação (14)

do *Intersection over Union* (IoU) (Santos, 2022):

$$IoU = \frac{\text{Área}(B_p \cap B_{gt})}{\text{Área}(B_p \cup B_{gt})} \quad (14)$$

onde as áreas B_p e B_{gt} representam a instância predita e o *Ground Truth*, respectivamente.

O IoU é um hiperparâmetro que define a percentagem mínima de interseção necessária para que uma inferência seja considerada verdadeira. Com base neste resultado, surgem quatro grandezas que constituem uma matriz de confusão, conforme apresentado na Tabela 1:

- **Verdadeiros Positivos (*True Positives* - TP):** Resultados corretamente previstos como positivos;
- **Falsos Positivos (*False Positives* - FP):** Resultados incorretamente previstos como positivos;
- **Verdadeiros Negativos (*True Negatives* - TN):** Resultados corretamente previstos como negativos;
- **Falsos Negativos (*False Negatives* - FN):** Resultados incorretamente previstos como negativos.

Tabela 1 - Matriz de confusão

	Valor real	
Resultado da classificação	Positivo	Negativo
Positivo	Verdadeiros Positivos (TP)	Falsos Positivos (FP)
Negativo	Verdadeiros Negativos (TN)	Falsos Negativos (FN)

As grandezas apresentadas e explicadas anteriormente, constituem a *Precision*, a *Accuracy* e o *Recall*, que são métricas fundamentais para a avaliação da performance de um modelo treinado, representadas nas Equações (15), (16) e (17), respectivamente. Resumidamente, a

Precision, pode ser vista como uma taxa de verdadeiros positivos, uma vez que avalia a proporção de classificações corretas entre todas as previsões positivas, envolvendo TP e FP. A *Accuracy* mede a percentagem de classificações corretas no geral, considerando tanto TP quanto TN em relação a todas as previsões (TP, TN, FP e FN). Por outro lado, o *Recall* tem como foco o número de casos reais que foram corretamente identificados (TP em relação a TP + FN).

$$Precision = \frac{TP}{TP + FP} \quad (15)$$

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (16)$$

$$Recall = \frac{TP}{TP + FN} \quad (17)$$

A precisão média (*Average Precision - AP*), que combina a *Precision* e o *Recall*, é calculada através da Equação (18):

$$AP_{IoU} = \sum_{i=0}^N [w_i P(r_i)] = AUC \quad (18)$$

onde *AUC* (*Area Under the Curve*) representa a área da curva P×R, calculada de forma numérica, de acordo com o peso w_i e a *Precision* que decresce em função do *Recall* (r_i).

Finalmente, a média das precisões (*mean Average Precision - mAP*) é uma métrica utilizada para avaliação tanto do treino quanto do teste das redes *Mask R-CNN* e *YOLO*. Esta métrica é definida em função do IoU, que varia entre 50 a 95%, com uma distância de 5% entre os valores. O cálculo da mAP é realizado conforme descrito na Equação (19):

$$mAP_{IoU=50:5:95} = \frac{\sum_{i=1}^{10} mAP_i}{10} \quad (19)$$

O limite mínimo do IoU é definido como 0,5 (mAP50). quando o valor está acima desse limiar, é considerado como uma detecção correta, enquanto que valores abaixo são classificados como erros. Adicionalmente, a métrica mAP50-95 representa a média das precisões num intervalo que vai de 0,5 a 0,95, avaliando assim a performance do modelo em diferentes níveis de sobreposição (Silva, 2024).

A função de perda (*loss*), que é a função alvo a ser otimizada, analisa o grau de inconsistência entre os valores previstos pela rede e os valores obtidos. O treino de CNNs corresponde ao processo de minimização da função de perda, isto é, um valor abaixo da função de perda, aponta para uma aproximação entre os valores previstos e os valores reais, indicando que a rede é robusta. A função *loss* compreende três partes (qualidade da classificação, pontos das *bounding boxes* e ainda a segmentação da máscara), tal como apresentado na Equação (20) (Santos, 2022):

$$L = L_{class} + L_{bbox} + L_{mask} \quad (20)$$

onde L_{class} e L_{mask} são calculadas através das Equações (21) e (23), que necessitam das (22) e (24), respetivamente:

$$L_{class} = l(x, y) = \{l_1, \dots, l_N\}_{class}^T \quad (21)$$

$$l_{n,class} = -w_{y_n} \log\left(\frac{e^{x_n y_n}}{\sum_{c=1}^c e^{x_n c}}\right) y_n \quad (22)$$

$$L_{mask} = l(x, y) = \{l_1, \dots, l_N\}_{mask}^T \quad (23)$$

$$l_{n,mask} = -w_{y_n} \{y_n \cdot \log[x_n] + (1 - y_n) \cdot \log(1 - x_n)\} \quad (24)$$

onde N representa o tamanho do *batch*, $l(x, y)$ a média dos vários *batches*, n a amostra individual, x e y são a rotulagem feita e o resultado da previsão, respetivamente, enquanto w_{y_n} corresponde ao peso opcional dado para cada amostra de dados, de forma manual.

Relativamente à L_{bbox} , esta define-se de forma diferente das anteriores (Equações (25) e (26)):

$$L_{bbox} = L_{loc}(t^u, v) = \sum_{i \in \{x, y, w, h\}} smooth_{L_1}(t^u - v_i) \quad (25)$$

$$smooth_{L_1}(x) = \begin{cases} 0,5x^2 & \text{se } |x| < 1 \\ |x| - 0,5 & \text{caso contrário} \end{cases} \quad (26)$$

onde t^u é uma tupla (estrutura de dados em linguagens como *Python* que armazena uma sequência ordenada de elementos imutáveis) com 4 coordenadas de previsões realizadas pelo modelo para as *bounding boxes* e v representa uma tupla que representa os *Ground Truths*.

2.3 Ray Casting

O *Ray Casting* é uma técnica de projeção amplamente reconhecida em áreas como computação gráfica, simulações físicas e sistemas de detecção, sendo comumente utilizada em aplicações como renderização de imagens, simulações LiDAR e interação humano-computador (HCI). Essa técnica baseia-se no lançamento de raios de luz virtuais a partir de uma fonte específica, como uma câmara ou sensor, em várias direções. Os raios seguem trajetórias até intersetarem objetos, momento em que são recolhidas informações como o ponto de interseção, a cor e a normal da superfície. Esses dados podem ser utilizados para gerar uma imagem ou para calcular interações físicas no ambiente. No entanto, durante a pesquisa bibliográfica para este trabalho, constatou-se que a quantidade de informações disponíveis sobre *Ray Casting* é bastante limitada, especialmente em aplicações mais recentes. Em particular, não foram encontradas publicações ou exemplos de seu uso no âmbito da engenharia civil, o que evidencia uma lacuna significativa na aplicação desta técnica em áreas como simulações estruturais e otimização de processos construtivos. Além disso, ao contrário do ray tracing, o *Ray Casting* foca-se unicamente na primeira interseção dos raios, sem simular efeitos mais complexos, como reflexão ou refração (Rougeron, Garrec e Andriot, 2022). Dessa forma, o objetivo proposto neste trabalho concentra-se na projeção das inferências dos danos em modelos 3D, conforme apresentado na Figura 31, destacando o potencial do *Ray Casting* para mapear anomalias nos modelos tridimensionais de ativos industriais.

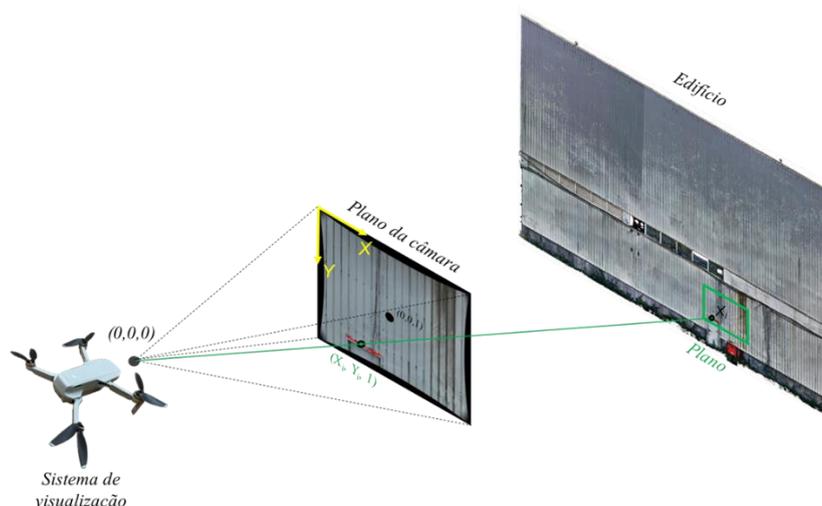


Figura 31 - Ray Casting pretendido

Geralmente, o processo começa com a definição da origem dos raios, que geralmente é a posição da câmara, com a coordenada (0,0,0). A partir dessa origem, os raios são projetados em direção ao plano da câmara. Para cada pixel, um raio é lançado em uma direção específica, que é calculada com base nas coordenadas normalizadas do pixel, levando em conta as propriedades intrínsecas da câmara. Um raio pode ser representado por uma linha que se estende da posição da câmara até o ponto de imagem normalizado. A normalização dos pontos envolve o deslocamento da origem para o centro da imagem e o ajuste do comprimento focal para uma unidade. Os pontos normalizados são então expressos em coordenadas como $(X_i, Y_i, 1)$. Para descrever completamente o raio, podemos introduzir um parâmetro escalar t , que nos permite calcular os pontos ao longo do raio em 3D, apresentado na Equação (27) (Cabral, 2023).

$$Ray(t) = (O + X_i t, O + Y_i t, O + 1 \cdot t) \quad (27)$$

onde X_i e Y_i são constantes, O trata-se do ponto de origem do raio e t a profundidade do ponto.

Para determinar se um raio intersecta um objeto, é comum utilizar-se diferentes equações, dependendo da geometria da superfície. No caso da interseção com o plano, utiliza-se a seguinte Equação (28):

$$Ax + By + Cz + D = 0 \quad (28)$$

onde A , B , C e D representam constantes e x , y e z são coordenadas locais.

Ao resolver esta Equação em relação a t (Equação (29)), é possível obter o ponto de interseção, representado na Equação (30), caso exista.

$$Ax_i t + By_i t + Ct + D = 0 \quad (29)$$

$$t = \frac{-D}{Ax_i + By_i + C} \quad (30)$$

Em termos de principais aplicações desta técnica, o *Ray Casting* evidencia-se em áreas como:

- **Renderização gráfica:** Em ambientes 3D, o *Ray Casting* é usado para renderizar imagens, baseando-se nas interseções entre os raios e os objetos. Avanços nessa área, como as implementações utilizando GPUs da *Nvidia Optix*, mostram como essa técnica

pode ser otimizada. Modelos mais avançados, como o HELIOS++ (*Heidelberg LiDAR Operations Simulator*), apresentado na Figura 32, conseguem capturar múltiplos pontos de interseção, caracterizando-se por ser um recurso essencial para simulações LiDAR (Winiwarter *et al.*, 2022).

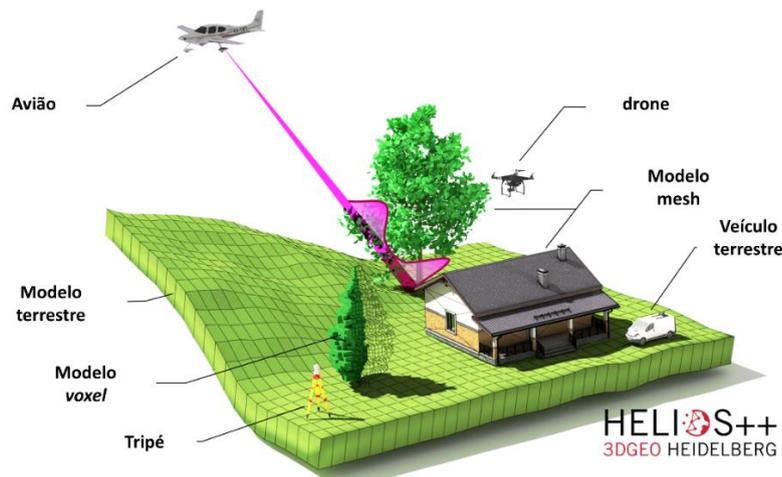


Figura 32 - Esquema do conceito HELIOS++ (Winiwarter *et al.*, 2022)

- **Simulação LiDAR:** O *Ray Casting* é habitualmente utilizado em simulações LiDAR, nas quais cada raio virtual efetua uma simulação de um feixe de laser emitido pelo sensor, com o intuito de medir distâncias e gerar mapas de profundidade. Ferramentas como o DIRSIG (*Digital Imaging and Remote Sensing Image Generation*) possibilitam a realização de uma análise física detalhada da interação entre a luz e os materiais, utilizando essa técnica (Goodenough e Brown, 2017).
- **Interação Humano-Computador:** Em interfaces de seleção de alvos em ambientes 3D, o *Ray Casting* é bastante popular. Em cenários de realidade virtual ou aumentada, o utilizador tem a possibilidade de apontar para objetos distantes com um raio virtual.

Existem três tipos de de estratégia, para a aplicação do *Ray Casting* em algoritmos paralelos, nomeadamente a ordem do objeto, a ordem da imagem ou a ordem híbrida. Todos os tipos de aplicação necessitam, no entanto, dos seguintes aspetos (Ray *et al.*, 1999):

- O sistema de memória gera os valores de *voxel* (*pixel* de uma imagem 3D ou unidade mínima de uma representação virtual tridimensional) necessários a uma taxa que representa o desempenho da arquitetura;

- O cálculo do trajeto do raio define os *voxels* que são penetrados por um dado raio;
- A interpolação estima o valor numa determinada localização, recorrendo aos valores de *voxel* próximos;
- A estimativa de gradiente gera uma normal de superfície, através de uma vizinhança de *voxels* ao redor da localização;
- A classificação mapeia os valores das amostras interpolados e a normal à superfície estimada;
- O sombreamento baseia-se em informações de gradiente e classificação para gerar uma cor que considera a interação da luz nas superfícies estimadas;
- A composição usa valores de cor para calcular uma cor final do *pixel* para visualização.

Caracteriza-se ainda pela sua versatilidade, que permite aplicações nas mais variadas áreas, desde a renderização gráfica até simulações físicas e médicas (Díaz-García *et al.*, 2018). No entanto, o *Ray Casting* apresenta algumas limitações. Uma delas prende-se com a utilização em ambientes densos, com múltiplos objetos ou geometrias complexas, em que pode haver imprecisão na seleção correta de objetos, especialmente em cenários interativos, tal como mencionado por Chen *et al.* (2023). Para além disso, a técnica não considera efeitos de luz complexos, como reflexões e refrações, o que a limita em aplicações que exigem alto realismo (Rougeron, Garrec e Andriot, 2022). De modo a ultrapassar essas restrições, novas técnicas estão em constante desenvolvimento. O uso de GPUs tem melhorado o desempenho em simulações LiDAR, possibilitando a captura de múltiplos picos de sinal com maior precisão (Winiwarter *et al.*, 2022). Cumulativamente, avanços como o *BackTracer*, proposto por Chen *et al.* (2023), têm aperfeiçoado a precisão na seleção em interfaces, em cenários de realidade virtual ou aumentada, mitigando o tremor das mãos e aumentando a eficiência .

3 Metodologia de detecção automática de anomalias

A metodologia desenvolvida de detecção automática de anomalias é composta por cinco etapas, tal como demonstrado na Figura 33. Ao longo deste capítulo essas mesmas etapas serão aprofundadas, destacando-se não só as tecnologias e equipamentos utilizados, mas também mostrando algumas campanhas experimentais e testes laboratoriais que foram essenciais para a validação da metodologia.

A demonstração das etapas da metodologia de detecção de anomalias é realizada com base em dois casos de estudo. O primeiro é a pala de um edifício industrial da empresa NACEX, um exemplo de menor escala, que apresenta corrosões e choques mecânicos, ideal para testar a metodologia. Por sua vez, o segundo é um quadro com alvos, utilizado para validar a precisão das medições de distâncias e áreas na gestão de dados concebida com *Ray Casting*.

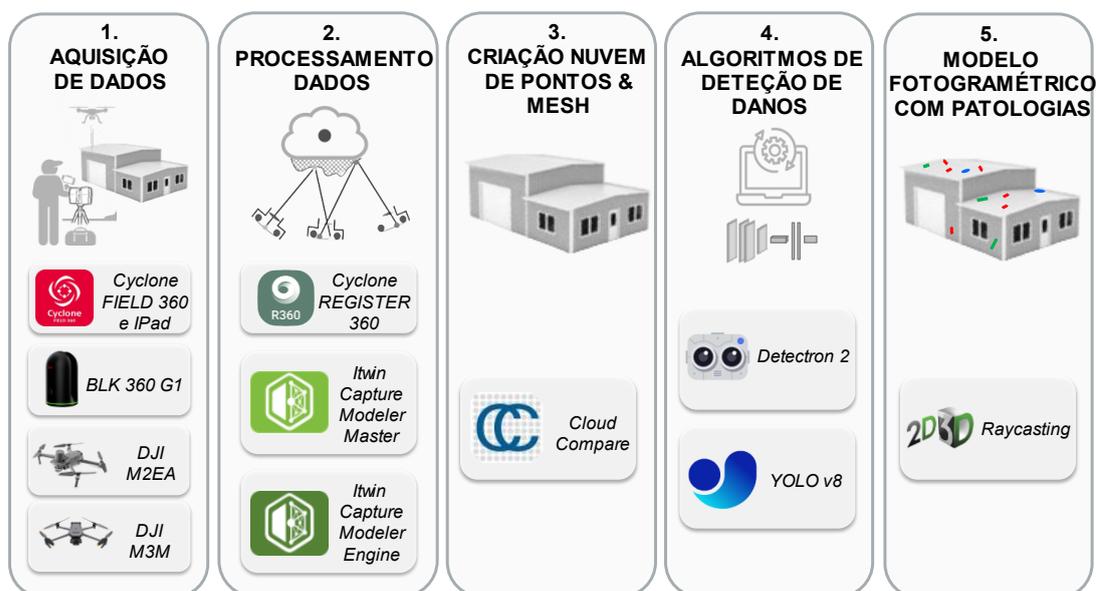


Figura 33 - Metodologia desenvolvida para detecção automática de anomalias

Todo o processamento foi realizado recorrendo a um computador com características específicas, mais concretamente, com uma memória RAM de 64GB, um processador *13th Gen Intel(R) Core (TM) i9-13900KF 3.00 GHz*, um armazenamento de massa de 1 TB e uma placa gráfica *GPU NVIDIA GeForce RTX 4090* com de 24 GB de memória GDDRX.

3.1 Aquisição de dados

A aquisição de dados é uma etapa de especial importância, uma vez que não é só essencial para a criação da nuvem de pontos e da *mesh* do ativo em estudo, como também serve para aumentar o *dataset* de treino do algoritmo de detecção de danos, através das imagens capturadas pelos *drones*.

3.1.1 BLK360 G1, Cyclone Field 360 e iPad

Relativamente ao *Terrestrial Laser Scanner* usado neste trabalho foi o *BLK360 G1* (Figura 34), produto da *Leica* apresentado em 2016, que se caracteriza pela capacidade de obtenção de 360000 pontos por segundo, num campo de visão de 300° sobre um eixo horizontal e de 360° sobre um eixo vertical, com um alcance de 60 metros de raio. Possui ainda três câmaras que permitem obter imagens *HDR* e térmicas.



Figura 34 - *BLK360 G1* (Allen Precision Equipment, 2023)

A digitalização pode ser realizada em 3 modos de densidade (baixo, médio e alto), tal como apresentado na Tabela 2 (Leica Geosystems AG, 2024).

Tabela 2 - Precisão e duração de levantamentos consoante o modo de densidade

Modo de densidade	Precisão (a 10m)	Duração do scan (min)
Baixo	20mm	01:40
Médio	10mm	02:50
Alto	6mm	04:40

É possível integrar o *BLK360 G1* com a aplicação *Cyclone FIELD 360* acessível através de um tablet ou smartphone (Figura 35). Esta aplicação filtra e recebe dados do equipamento em tempo real e, como destacado por Oliveira et al. (2023), permite que, ao mover o equipamento

para uma nova estação e realizar um novo levantamento digital, a conexão com o último scan realizado seja mantida por meio de um *link*. Esta capacidade facilita o processamento posterior no escritório. Caso essa ligação não seja estabelecida em campo, o *Cyclone Register 360* sobrepõe todos os levantamentos executados, o que dificulta o posterior registo das nuvens de pontos.



Figura 35 - Utilização do *Cyclone FIELD 360* em campo

3.1.2 DJI M2EA e DJI M3M

Dado que os ativos industriais são edifícios de dimensões consideráveis em que nem todas as superfícies se apresentam com fácil acesso, foram utilizados dois drones para aumentar a eficiência das campanhas de coleta de dados. Os drones utilizados foram o *DJI Mavic 2 Enterprise Advanced* (DJI M2EA) e o *DJI Mavic 3 Multispectral* (DJI M3M), conforme ilustrado na Figura 36 (a) e na Figura 36 (b), respetivamente.



(a)



(b)

Figura 36 - DJI M2EA (a) e DJI M3M (b) (DJI, 2023)(DJI, 2024)

A comparação das especificações desses dois UAVs está detalhada na Tabela 3.

Tabela 3- Comparação de características entre o DJI M2EA e o DJI M3M -

	M2EA	M3M
Peso / Peso máximo	909 g / 1100 g	951 g / 1050 g
Tempo de voo máximo	31 min	43 min
Velocidade máxima	20 m/s	21 m/s
Câmaras	Térmica: resolução 640x512 px RGB: com sensor de 48 MP	Multiespectral: com 4 sensores RGB: com sensor de 20 MP
Resolução de imagens	8000x6000 px	5280x3956 px
GNSS	GPS + GLONASS	GPS + Galileo + BeiDou + GLONASS
Acessórios	RTK, alto-falante e luzes que, em alternativa, podem ser colocados no <i>drone</i>	RTK e sensores multiespectrais incorporados no <i>drone</i>
Classe	CLASS 2	CLASS 2

O DJI M3M utiliza mais constelações de satélites na obtenção de sinal GNSS e RTK e tem uma duração da bateria um terço superior comparativamente ao DJI M2EA. Isto permite que o equipamento DJI M3M tenha uma navegação e posicionamento mais robustos principalmente em ambientes complexos em termos de interferência do sinal. Para além disso, o facto da resolução das imagens ser diferente entre as câmaras dos dois drones, fez com que fosse necessário haver um ajuste no recorte e na reconstrução das imagens no procedimento de *Ray Casting*.

As operações com os drones DJI M2EA e DJI M3M, são acompanhadas por um comando de controlo (Figura 37) que permite controlar o voo do veículo de forma manual, através da aplicação *DJI Fly*. Para além disso, possibilita a definição de voos automáticos, estabelecendo um percurso e altitudes de voo através de uma área e uma altura predefinidos pelo utilizador.



Figura 37 - Comando de controlo do DJI M3 (DJI, 2024)

3.2 Processamento de dados

Após a recolha de dados por meio do TLS e dos UAVs, é fundamental proceder ao processamento dessas informações. Tal como demonstrado na Figura 38, embora várias etapas, como a georreferenciação, a limpeza da nuvem de pontos e a exportação, sejam comuns a ambos os métodos, há diferenças nos processos e *softwares* utilizados (*Cyclone Register 360* e *iTwin Capture Modeler*), dependendo do tipo de dados recolhidos. Essas diferenças serão discutidas nesta secção.

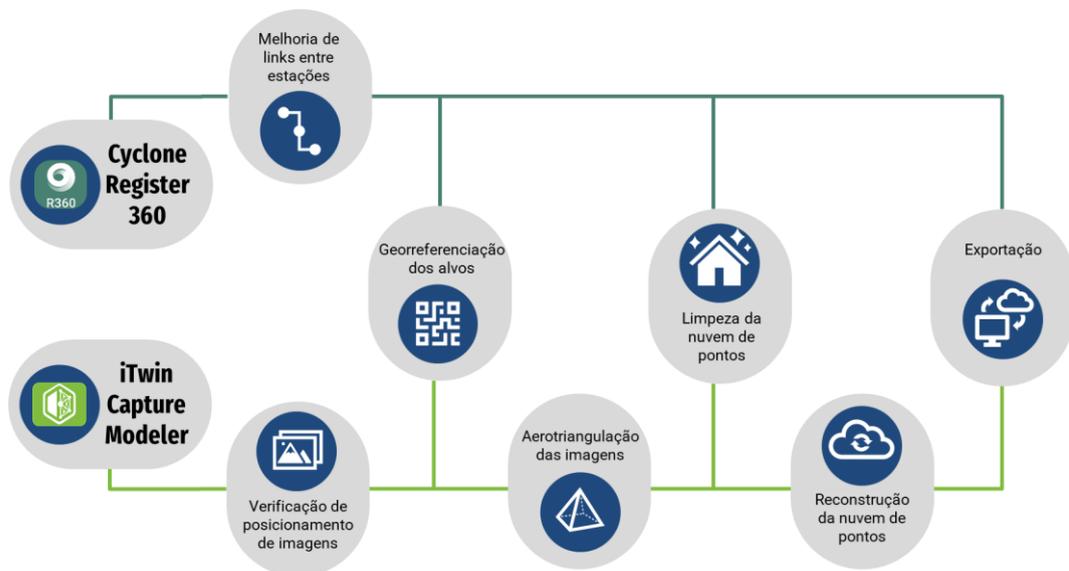


Figura 38 - Esquema do processamento de dados obtidos

3.2.1 Cyclone Register 360

O *Cyclone Register 360*, desenvolvido pela *Leica Geosystems*, foi a escolha mais adequada para processar as informações obtidas pelo *BLK 360 G1*, devido à sua compatibilidade. Após a importação da nuvem de pontos do *Cyclone FIELD 360* para o *Cyclone Register 360*, realizada através de um cabo de rede conectado entre o iPad e o computador, o processamento é organizado em quatro fases:

1. **Melhoria dos links entre estações:** com o objetivo de maximizar a resolução e precisão da nuvem de pontos, é necessário realizar uma melhoria dos *links* oriundos do levantamento (Figura 39 (a)). Esta melhoria baseia-se não só na união de estações próximas umas das outras, mas também da diminuição do erro nos *links* previamente criados. A obtenção de *Cloud-to-Cloud errors* baixos e a formação de triangulações

entre os levantamentos, proporciona um aumento no *Overlap* e consequente descida do *Bundle Error* (Figura 39 (b)), parâmetros de qualidade definidos pela aplicação;

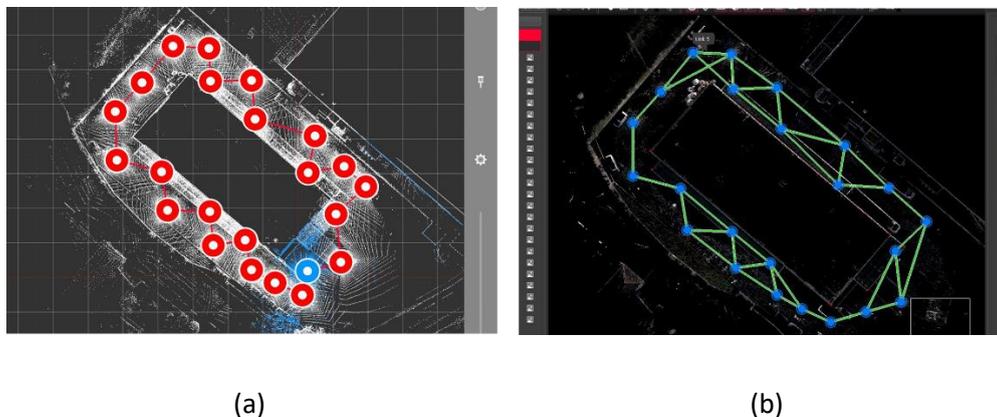


Figura 39 - Exemplo de melhoria dos links entre estações do *Cyclone FIELD 360* (a) para o *Cyclone Register 360* (b)

2. **Georreferenciação (Figura 40):** envolve o estabelecimento de alvos, marcas ou pontos que servem de base para a georreferenciação do modelo da nuvem de pontos antes do trabalho de campo. Com o auxílio de um topógrafo, que fornece um arquivo *.txt* contendo as coordenadas x, y e z levantadas, procede-se à marcação desses pontos nas imagens 360° capturadas pelo equipamento. A importação das coordenadas no *Cyclone Register 360* é, assim, feita de forma automática;

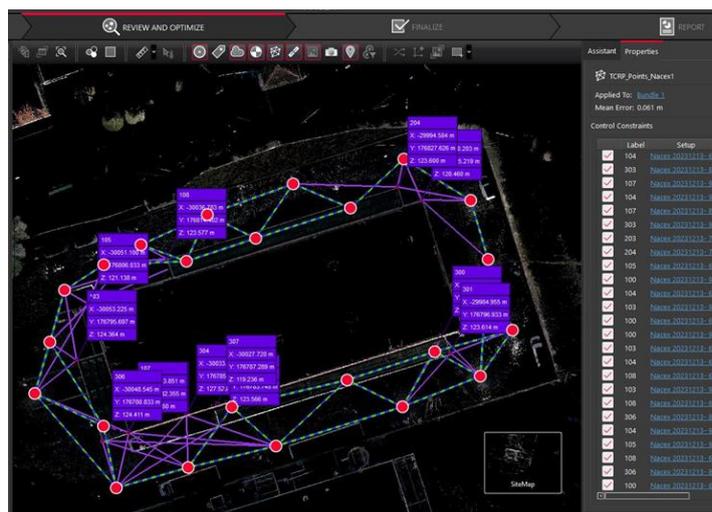


Figura 40 - Georreferenciação no *Cyclone Register 360*

3. **Limpeza e controlo de erros:** o *software* permite limpar todos os pontos da nuvem que não interessam ao utilizador, de forma a eliminar o ruído e ainda diminuir o erro *Cloud-to-Cloud* associado às zonas apagadas. Para além disso, o *Cyclone Register 360* gera um relatório final (Figura 41), onde é apresentada toda a informação relativa aos *links* que compõem a nuvem de pontos;

Link Name	Setup 1	Setup 2	Overlap	Abs. Mean Error
Link 2	Nacex 20231213- 64	Nacex 20231213- 65	57 %	0.004 m
Link 3	Nacex 20231213- 75	Nacex 20231213- 76	59 %	0.005 m
Link 4	Nacex 20231213- 75	Nacex 20231213- 74	66 %	0.008 m
Link 5	Nacex 20231213- 76	Nacex 20231213- 77	63 %	0.012 m
Link 6	Nacex 20231213- 77	Nacex 20231213- 78	66 %	0.004 m
Link 7	Nacex 20231213- 78	Nacex 20231213- 79	66 %	0.005 m
Link 8	Nacex 20231213- 79	Nacex 20231213- 80	65 %	0.004 m
Link 9	Nacex 20231213- 80	Nacex 20231213- 81	47 %	0.005 m
Link 10	Nacex 20231213- 81	Nacex 20231213- 82	58 %	0.004 m
Link 11	Nacex 20231213- 82	Nacex 20231213- 86	37 %	0.005 m
Link 12	Nacex 20231213- 87	Nacex 20231213- 89	68 %	0.005 m
Link 13	Nacex 20231213- 65	Nacex 20231213- 68	59 %	0.009 m
Link 14	Nacex 20231213- 89	Nacex 20231213- 90	73 %	0.007 m
Link 15	Nacex 20231213- 90	Nacex 20231213- 91	74 %	0.005 m
Link 16	Nacex 20231213- 68	Nacex 20231213- 69	64 %	0.009 m
Link 17	Nacex 20231213- 69	Nacex 20231213- 70	65 %	0.012 m
Link 18	Nacex 20231213- 70	Nacex 20231213- 71	59 %	0.011 m
Link 19	Nacex 20231213- 71	Nacex 20231213- 72	62 %	0.009 m
Link 20	Nacex 20231213- 72	Nacex 20231213- 73	58 %	0.008 m
Link 21	Nacex 20231213- 73	Nacex 20231213- 74	25 %	0.004 m
Link 22	Nacex 20231213- 87	Nacex 20231213- 86	40 %	0.006 m
Link 23	Nacex 20231213- 63	Nacex 20231213- 91	49 %	0.007 m
Link 27	Nacex 20231213- 78	Nacex 20231213- 80	57 %	0.005 m
Link 29	Nacex 20231213- 87	Nacex 20231213- 90	57 %	0.005 m
Link 31	Nacex 20231213- 65	Nacex 20231213- 69	57 %	0.010 m
Link 34	Nacex 20231213- 89	Nacex 20231213- 91	50 %	0.008 m
Link 35	Nacex 20231213- 68	Nacex 20231213- 70	48 %	0.008 m
Link 36	Nacex 20231213- 69	Nacex 20231213- 71	57 %	0.011 m
Link 37	Nacex 20231213- 72	Nacex 20231213- 70	37 %	0.010 m
Link 39	Nacex 20231213- 64	Nacex 20231213- 63	51 %	0.005 m
Link 41	Nacex 20231213- 77	Nacex 20231213- 79	62 %	0.006 m
Link 42	Nacex 20231213- 81	Nacex 20231213- 86	68 %	0.006 m
Link 43	Nacex 20231213- 79	Nacex 20231213- 81	42 %	0.009 m
Link 44	Nacex 20231213- 76	Nacex 20231213- 78	51 %	0.006 m
Link 45	Nacex 20231213- 75	Nacex 20231213- 77	25 %	0.006 m
Link 47	Nacex 20231213- 68	Nacex 20231213- 64	44 %	0.007 m
Link 48	Nacex 20231213- 80	Nacex 20231213- 82	32 %	0.011 m

Figura 41 - Excerto de um relatório final do *Cyclone Register 360*

4. **Exportação:** por fim, surge a fase da exportação, em que é escolhido qual o formato (*.rcp*, *.e57*, *.las*, etc.) pretendido para a nuvem de pontos do *BLK360 G1* (Figura 42).

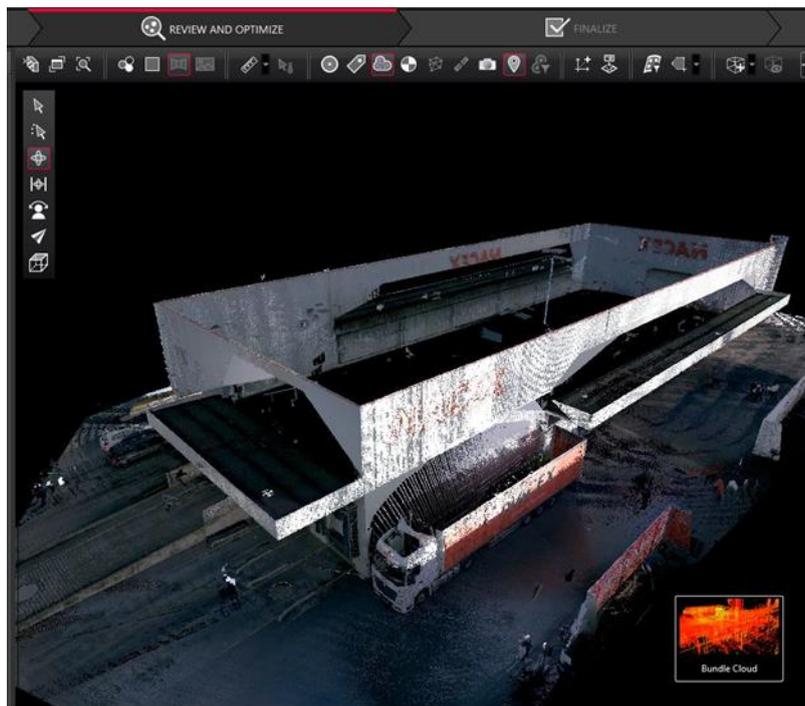


Figura 42 - Exemplo do resultado da nuvem de pontos do BLK360 G1

3.2.2 iTwin Capture Modeler Master & Engine

Embora após vários testes o *Agisoft Metashape* tenha demonstrado melhores resultados em termos de precisão da nuvem de pontos, devido às necessidades do trabalho e à eficiência, o *software* escolhido para o processamento das imagens capturadas pelos drones foi o *iTwin Capture Modeler Master* (disponibilizado pela *Bentley Systems*). Este *software* não só fornece nuvens de pontos precisas, mas também gera *malhas (meshes)*, que são essenciais para o funcionamento do *Ray Casting*. O processamento no *iTwin Capture Modeler Master* é complementado pelo *iTwin Capture Modeler Engine* e consiste em 6 etapas:

1. **Verificação de posicionamento de imagens:** ao importar as imagens para serem processadas no *software*, é necessário selecionar o sistema de coordenadas correto. Esta seleção é fundamental para garantir que a posição das imagens coincida com os dados do levantamento topográfico (Figura 43). Em todos os projetos desenvolvidos durante o trabalho, o sistema de coordenadas utilizado foi o *ETRS89 / Portugal TM06 (EPSG:3763) + EGM96 geoid height (EPSG:5773)* como sistema referencial de coordenadas;

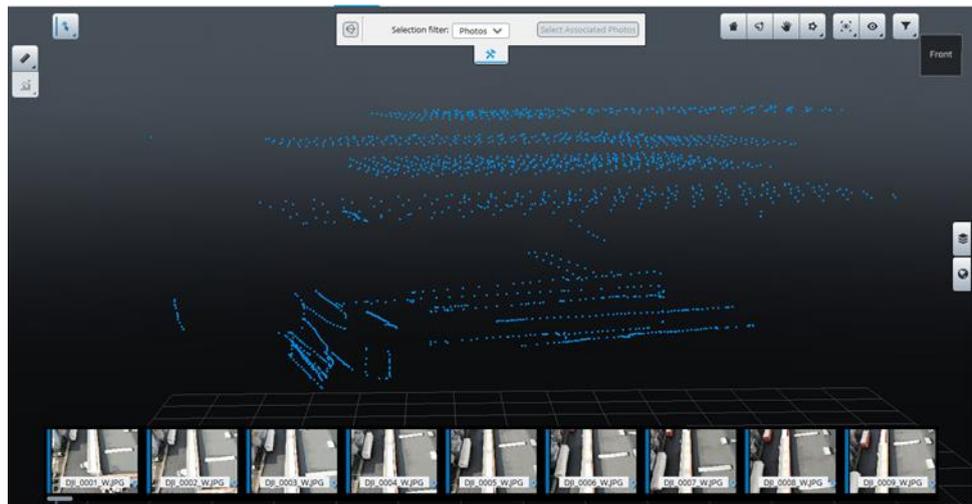


Figura 43 - Posicionamento das imagens importadas no iTwin

2. **Georreferenciação:** é um passo essencial a ser cumprido, já que permite a fusão das nuvens de pontos oriundas de cada uma das tecnologias digitais. A georreferenciação permite que as mesmas se sobreponham sem qualquer tipo de processo manual. Tal como no *Cyclone Register 360*, no *iTwin Capture Modeler* é necessário importar o arquivo fornecido pelo topógrafo (Figura 44 (a)) e marcar os pontos correspondentes nas várias imagens onde eles aparecem, conforme ilustrado na Figura 44 (b). Este processo garante que todas as fontes de dados estejam corretamente alinhadas no mesmo sistema de coordenadas;

Select a line for data preview

	Column 1	Column 2	Column 3	Column 4	Column 5
Row 1	103	-30053.2252	176795.6972	124.3638	
Row 2	104	-30051.1474	176806.1382	121.0579	
Row 3	105	-30051.1001	176806.0332	121.1382	
Row 4	107	-30043.8507	176782.3554	118.6599	
Row 5	108	-30036.7828	176814.4022	123.5766	
Row 6	200	-30035.4972	176811.1802	118.6521	
Row 7	201	-30010.8209	176818.8988	118.7220	

Fields selection

Column	Role	Preview
Column 1	Name	103
Column 2	X	-30053.2252
Column 3	Y	176795.6972
Column 4	Z	124.3638
Column 5	Not Applicable	



(a)

(b)

Figura 44 - Importação das coordenadas (a) e marcação dos pontos (b) no iTwin

3. **Aerotriangulação:** permite corrigir eventuais erros associados ao posicionamento das imagens e melhorar a sobreposição entre as mesmas, tendo em conta os pontos previamente marcados (Figura 45);

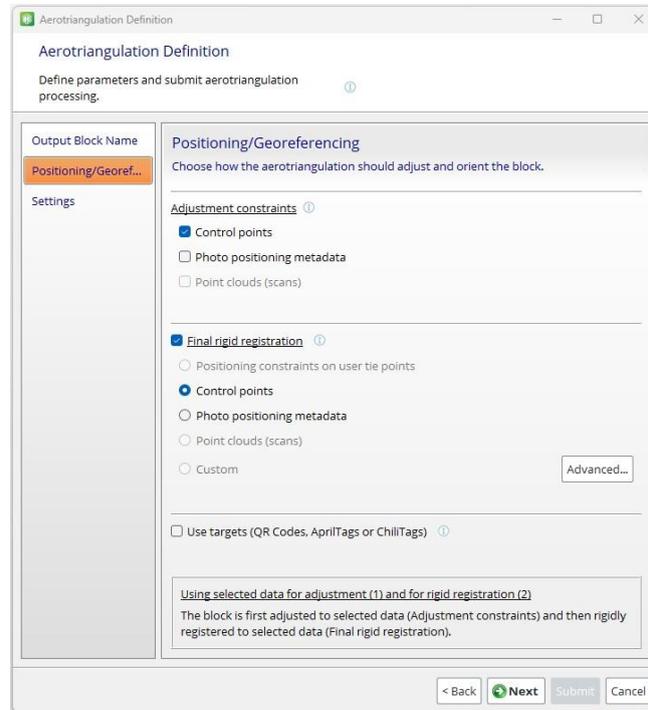


Figura 45 - Definições de aerotriangulação no iTwin

4. **Limpeza da área de interesse:** antes da reconstrução da nuvem de pontos e da *mesh*, deve proceder-se a uma seleção da região de interesse (Figura 46), eliminando áreas que não serão processadas, aliviando assim o custo computacional;
5. **Reconstrução da área de interesse selecionada:** deve-se ainda optar pela opção de *Adaptive Tiling*, tal como demonstrado na Figura 46, que divide a região selecionada, de modo a não sobrecarregar a memória do computador;

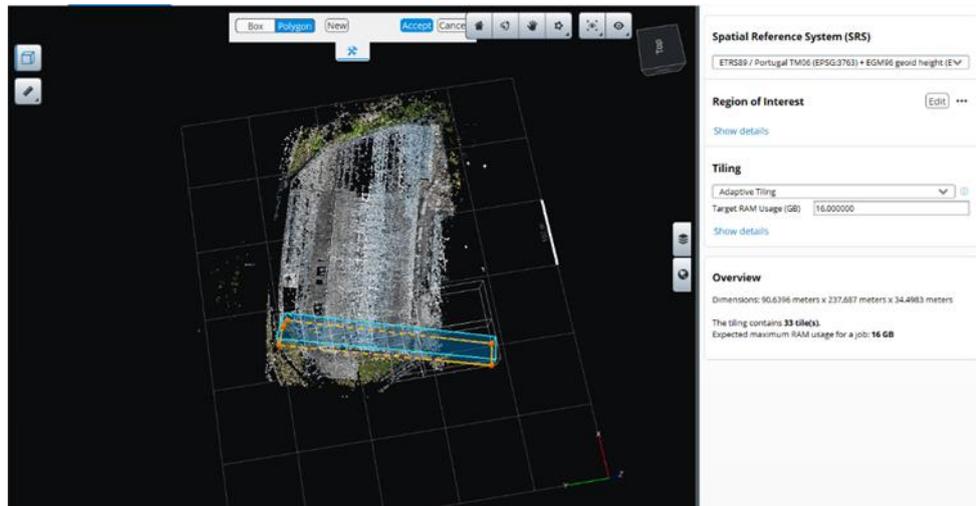
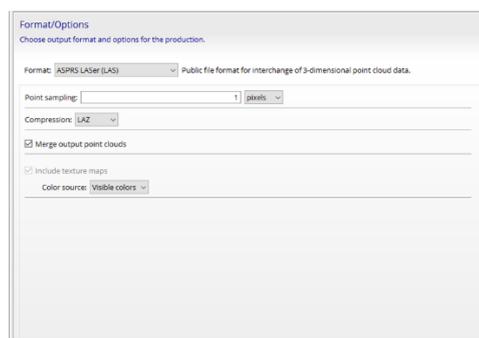
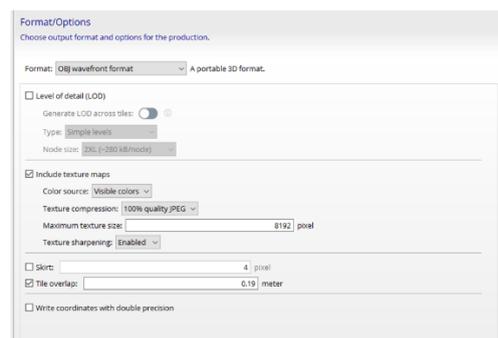


Figura 46 - Seleção da área de interesse a ser reconstruída

6. **Exportação da nuvem de pontos e da *mesh*:** para a produção e posterior exportação da nuvem de pontos e da *mesh*, é necessário definir tanto o formato de arquivo quanto os hiperparâmetros que serão utilizados no processo. Neste trabalho, o formato **.las** foi escolhido para a exportação das nuvens de pontos, com a seleção dos hiperparâmetros correspondentes, conforme mostrado na Figura 47 (a). Já para a *mesh*, foi utilizado o formato **.obj**, também com os seus respetivos hiperparâmetros ajustados, como representado na Figura 47 (b).



(a)



(b)

Figura 47 - Definições da nuvem de pontos (a) e da *mesh* (b)

O resultado da *mesh* exportada no *iTwin Capture Modeler Master*, encontra-se representado na Figura 48.

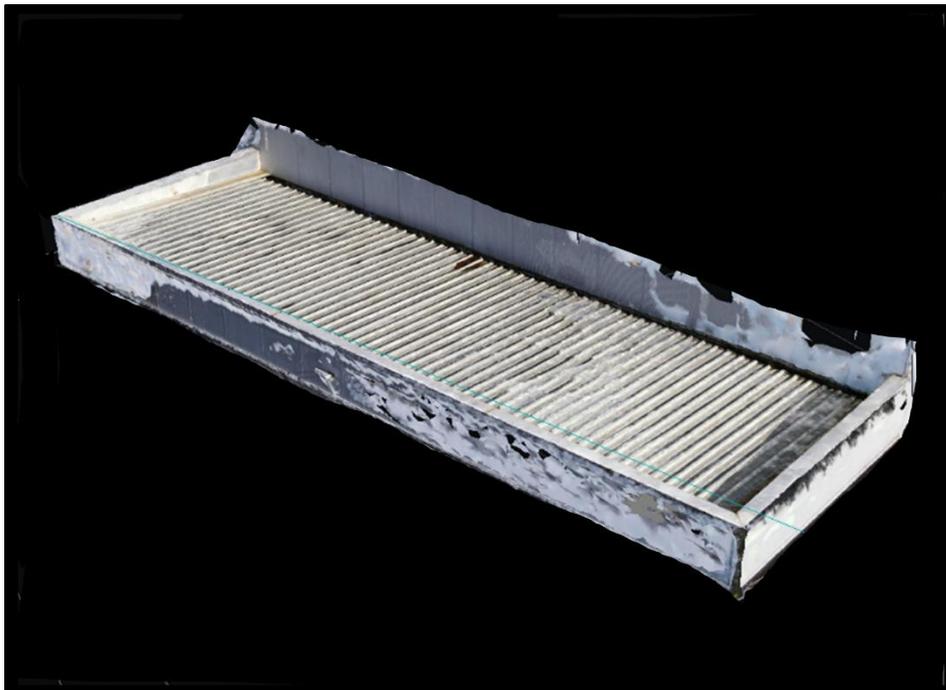


Figura 48 - Exemplo de resultado da *mesh*

3.3 Fusão das nuvens de pontos

Após o processamento e exportação das nuvens de pontos, oriundas do *Cyclone Register 360* (Figura 49 (a)) e do *iTwin Capture Modeler Master* (Figura 49 (b)), realiza-se a fusão destas. A nuvem de pontos gerada pelo TLS foi capturada apenas a partir de posições inferiores, enquanto a nuvem de pontos obtida pelos UAV's abrange a estrutura a partir de ângulos superiores, proporcionando assim uma cobertura completa do ativo.

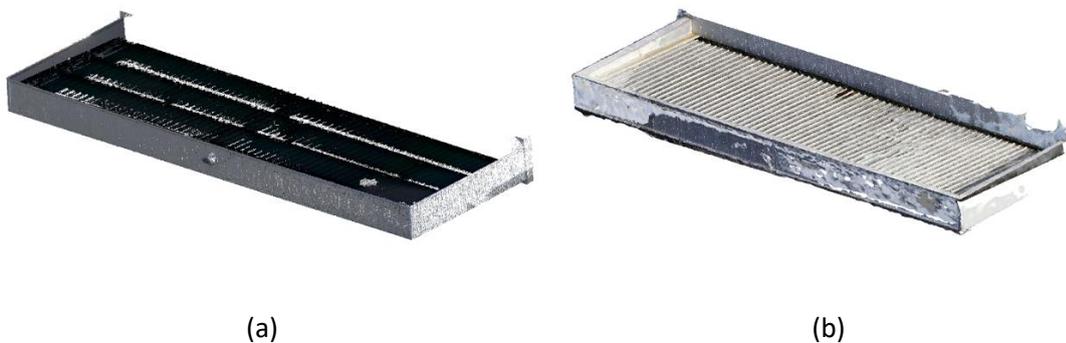


Figura 49 - Resultados das nuvens de pontos do LiDAR (a) e da fotogrametria (b)

A fusão das nuvens de pontos é realizada utilizando uma combinação do algoritmo iterativo ICP

(*Iterative Closest Point*), conforme discutido anteriormente, e a ferramenta *Merge* do software *CloudCompare*. Este processo, conforme ilustrado na Figura 50, envolve a definição de qual nuvem de pontos será a de referência e qual será alinhada. Além disso, é necessário escolher o critério de aplicação do ICP: pode-se basear no número de iterações ou na diferença de RMS (*Root Mean Square*), que especifica a distância máxima aceitável entre as nuvens.

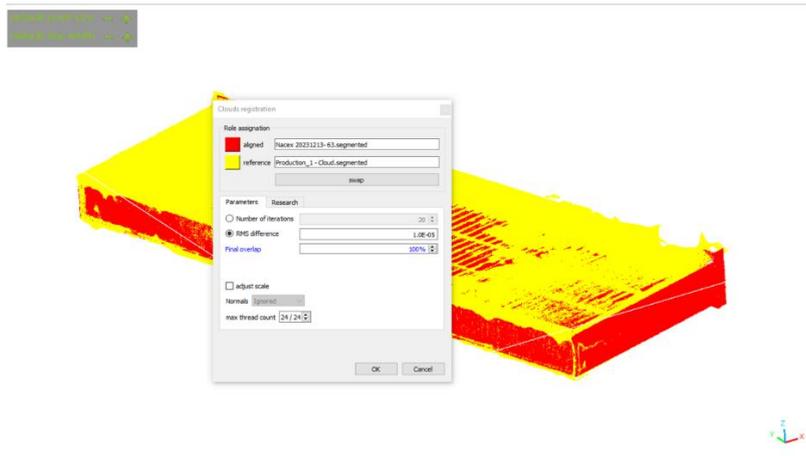


Figura 50 - Parâmetros a definir no ICP

Após a execução, as duas nuvens de pontos são combinadas, resultando numa única nuvem fundida, fazendo com que a pala tenha as duas faces (superior e inferior), como mostrado na Figura 51. As duas foram capturadas com os dois equipamentos a funcionar a partir de diferentes campos de visão.



Figura 51 - Resultado da fusão aplicada na pala da NACEX

3.4 Algoritmo de deteção de corrosão, choque mecânico e acumulação de água

Nesta secção é apresentado o algoritmo de identificação das três anomalias (corrosão, choque mecânico e acumulação de água) em painéis *sandwich*. O processo de desenvolvimento do algoritmo dividiu-se em quatro passos, tal como representado na Figura 52.

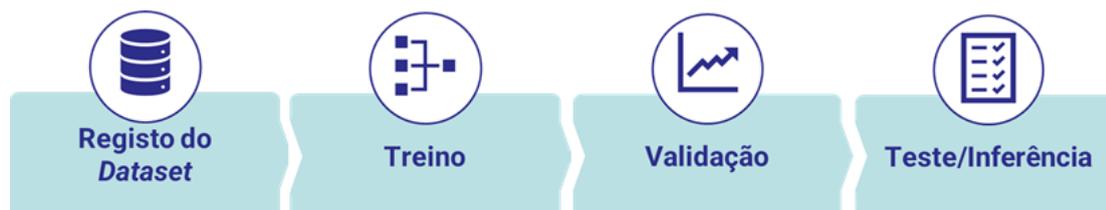


Figura 52 - Processo de desenvolvimento do algoritmo de deteção de anomalias

Este trabalho recorre a dois tipos de algoritmos, sendo eles o *Detectron2* e o *YOLOv8*. O primeiro, *Detectron2*, consiste numa biblioteca que contém algoritmos de deteção e segmentação de objetos, desenvolvida pelo grupo *Facebook AI Research*. O *Detectron2* surgiu com relevância devido ao seu desempenho superior no treino de uma *Mask R-CNN*, especialmente quando comparado a outras soluções, como o *Detectron* original e o *maskrcnn-benchmark*. Esta vantagem pode ser observada na Tabela 4 (Facebook AI Research's, [s.d.], detectron2 contributors, 2019).

Tabela 4 - Comparação de rendimentos entre algoritmos *Mask R-CNN* (Detectron2 contributors, 2019)

Implementation	Throughput (img/s)
Detectron2	62
mmdetection	53
maskrcnn-benchmark	53
tensorpack	50
simpledet	39
Detectron	19
matterport/Mask_RCNN	14

O segundo, *YOLOv8*, proposto pela *Ultralytics* em janeiro de 2023, trata-se de um refinamento da *YOLOv5*, também desenvolvida pela mesma empresa. Este algoritmo tem a capacidade de realizar detecção, classificação e segmentação de objetos, caracterizando-se pela rapidez, precisão e facilidade associadas, tal como é possível perceber nos gráficos da Figura 53 (Roboflow Universe, [s.d.], ULTRALYTICS, [s.d.]).

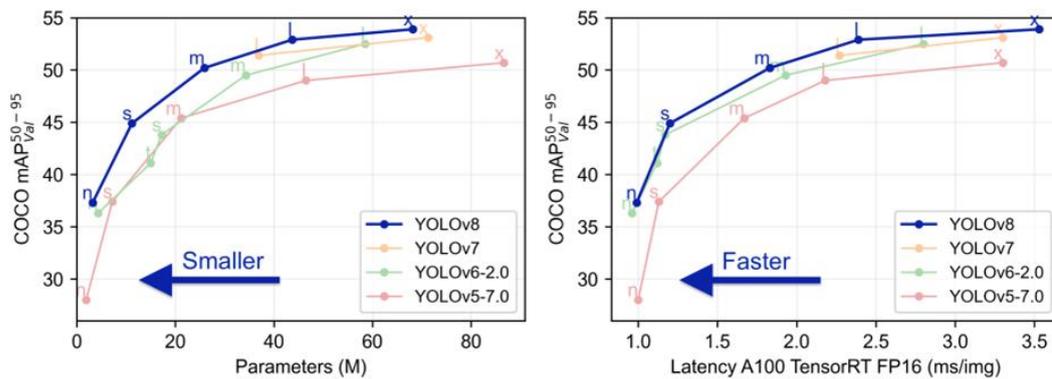


Figura 53 - Comparação entre a *YOLOv8* e as restantes redes YOLO (Ultralytics, 2022)

Os treinos, validações, testes e inferências do *Detectron2* foram realizados em ambiente de programação *Python 3.9*, enquanto para o *YOLOv8* foram desenvolvidos no *Python 3.11*, ambos com suporte de um GPU utilizando o CUDA 12.1. Ao longo das seguintes secções, será realizada uma análise comparativa entre o *Detectron2* e o *YOLOv8*, abordando o processo de preparação dos dados, a escolha dos hiperparâmetros, os resultados obtidos em cada etapa, entre outros. Esta análise permitirá avaliar a eficácia de cada modelo, assim como os desafios e benefícios específicos de cada *framework* no contexto da identificação de anomalias em painéis *sandwich*.

3.4.1 Registo do *dataset*

O primeiro passo do desenvolvimento do algoritmo consiste no registo do *dataset* utilizado para treino, validação e teste. Antes de realizar este registo, é imprescindível realizar a etiquetagem das imagens. Existem vários programas que permitem etiquetar objetos em imagens, sendo alguns dos mais utilizados o *LabelMe*, o *Supersely*, o *Pixel Annotation Tool*, o *VGG Image Annotator (VIA)* e o *Roboflow*.

3.4.1.1 Detectron2

No que diz respeito à base de dados do *Detectron2*, as imagens foram etiquetadas com recurso ao VIA, concebida pelo *Visual Geometry Group* (VGG). Este *software* permite realizar anotações em imagens, vídeos e áudios (Figura 54 (b)). Este programa destaca-se dos demais devido à facilidade de uso associada ao mesmo, acedendo-se diretamente através do navegador, com uma interface gráfica intuitiva e sem qualquer tipo de custos. O VIA permite ainda exportar as etiquetas em três tipos de formatos (*csv*, *json* e *COCO*), tal como apresentado na Figura 54 (a). No âmbito deste trabalho, foi escolhido o formato *.json* que inclui informações diversas, tais como, as coordenadas x e y referentes aos *pixels* da imagem e ainda o tipo de anomalia representado.

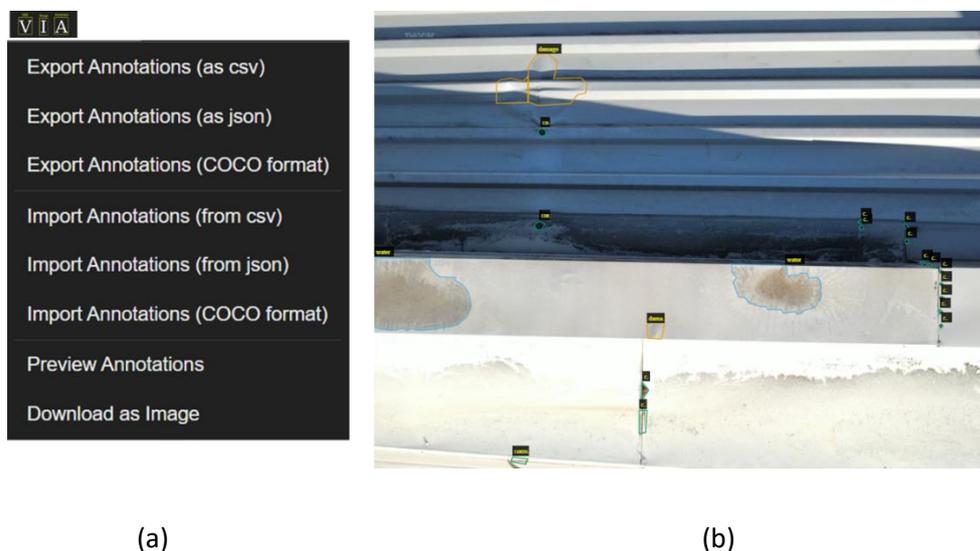


Figura 54 - Exportação das anotações (a) e ferramenta de anotações (b) no VIA

O registo do conjunto de dados no *Detectron2* envolve a utilização de uma função que converte os dados de entrada (imagens e ficheiro JSON com as anotações) num dicionário que contém informações sobre as imagens e as suas respetivas rotulagens. As imagens e anotações foram divididas em conjuntos de treino, validação e teste, correspondendo a 70%, 15% e 15% do total (Figura 56), respetivamente, de forma aleatória, conforme recomendado em estudos com os mesmos princípios base deste trabalho, como é o caso de Santos (2022).

3.4.1.2 YOLOv8

Relativamente ao conjunto de dados do *YOLOv8*, recorreu-se à funcionalidade de anotação do Roboflow, uma vez que esta plataforma apresenta uma elevada compatibilidade com o treino

de modelos YOLO, especialmente durante a exportação das etiquetas (Figura 55 (a)). Uma das principais vantagens desta ferramenta é a ampla variedade de formatos que consegue importar, ler e localizar (Figura 55 (b)), o que é particularmente relevante para este trabalho, uma vez que as anotações foram inicialmente realizadas no VIA antes da utilização do *YOLOv8*. Para além disso, a ferramenta de anotações do *Roboflow* inclui uma funcionalidade chamada *Smart Polygon*, que possibilita uma etiquetagem assistida. Com apenas um clique, é gerado um polígono de forma automática, facilitando significativamente o processo de anotação das imagens.

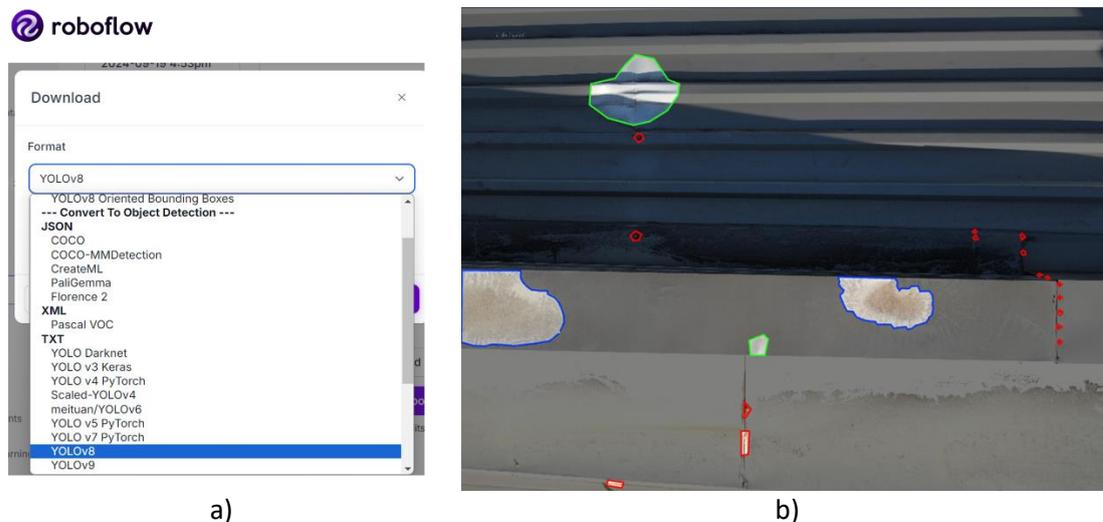


Figura 55 - Exportação das anotações (a) e ferramenta de anotações (b) no *Roboflow*

No YOLO, o registo do *dataset* resume-se a uma alteração de um ficheiro ("*data.yaml*") que é descarregado juntamente com as imagens e anotações. Essa modificação corresponde à indicação do número e nome das classes definidas e dos caminhos para as pastas que contêm os ficheiros descarregados, que se encontram separados em pastas de treino, validação e teste, com a mesma proporção adotada para o *Detectron2* (70%, 15% e 15%).

3.4.2 Treino

Os modelos *Detectron2* e *YOLOv8* caracterizam-se pela elevada precisão em deteção e segmentação de vários objetos. No entanto os algoritmos associados a essas redes enfrentam desafios quando se trata da segmentação de objetos pequenos e irregulares, como ocorre nas situações de corrosão, choques mecânicos e acumulação de água em painéis *sandwich*. Consequentemente, a abordagem adotada para os treinos, inicialmente para o *Detectron2* e depois também para o *YOLOv8*, dividiu-se em três etapas: recorte das imagens, aplicação de *Data Augmentation* e otimização de hiperparâmetros.

A apresentação de resultados dos treinos divide-se em deteção (“*det*”) e segmentação (“*segm*”), com maior destaque para a segunda, já que será relevante para o bom funcionamento da técnica de *Ray Casting*.

3.4.2.1 Recorte das imagens

Dado que as imagens utilizadas para treino foram todas capturadas pelo DJI M2EA, a resolução das mesmas é de 8000x6000 px. Para evitar sobrecarregar o custo computacional durante os treinos, tanto o *Detectron2* quanto o *YOLOv8* realizam automaticamente um redimensionamento das imagens de entrada para 1333x800px e 640x640px, respetivamente. Considerando este redimensionamento e a dificuldade desses modelos em detetar e segmentar objetos pequenos, a solução adotada foi recortar as imagens em 2000x2000px, dividindo-as em 12 partes. A escolha da resolução de 2000x2000px baseou-se em duas razões principais:

- O valor 2000 é um divisor comum de 8000 e 6000, permitindo dividir as imagens em partes iguais;
- O tamanho 2000x2000px não compromete o rendimento dos modelos durante o treino, mantendo os custos computacionais em níveis aceitáveis.

Uma vez que as 1570 imagens utilizadas para treino foram recortadas em 12 partes, gerou-se um impacto significativo no número de imagens enviadas para treino, passando para as 18840 imagens, contribuindo assim para uma melhoria dos modelos, tal como apresentado na Figura 56.

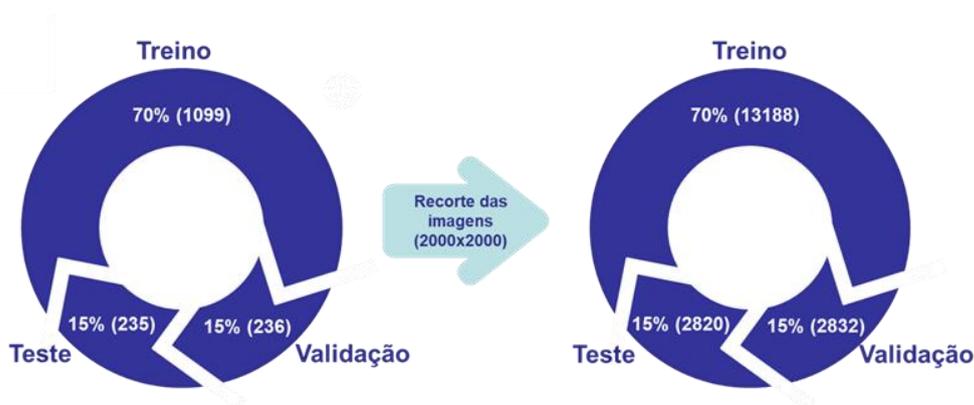


Figura 56 - Impacto do recorte 2000x2000 no número de imagens de treino

Em termos de métricas, observou-se um aumento em todas as principais métricas de desempenho, tanto na detecção como na segmentação nos dois modelos, excluindo a precisão média de detecção do *YOLOv8*. Destaca-se uma diferença notória nos valores de precisão com *Intersect over Union* de 50% (mAP50), tal como demonstrado na Tabela 5.

Tabela 5 - Comparação dos valores máximos obtidos entre as imagens originais e recortadas

Imagens originais (8000x6000)				Imagens recortadas (2000x2000)			
<i>Detectron2</i>		<i>YOLOv8</i>		<i>Detectron2</i>		<i>YOLOv8</i>	
Métrica	Valor Máximo	Métrica	Valor Máximo	Métrica	Valor Máximo	Métrica	Valor Máximo
AP (det)	12,32%	AP (det)	82,20%	AP (det)	26,04%	AP (det)	74,46%
AP50 (det)	39,33%	AP50 (det)	49,85%	AP50 (det)	49,73%	AP50 (det)	64,89%
AP75 (det)	22,10%	AP50-95 (det)	30,55%	AP75 (det)	23,54%	AP50-95 (det)	36,10%
AP (segm)	10,13%	AP (segm)	68,19%	AP (segm)	19,93%	AP (segm)	68,67%
AP50 (segm)	28,72%	AP50 (segm)	38,75%	AP50 (segm)	42,27%	AP50 (segm)	54,89%
AP75 (segm)	13,46%	AP50-95 (segm)	21,93%	AP75 (segm)	16,55%	AP50-95 (segm)	24,12%

Ao transferir a base de dados de imagens para o *Roboflow*, que permite visualizar as etiquetas, constatou-se que o recorte das imagens em 2000x2000 px desformatava as anotações. Quando esse recorte interseccionava as rotulagens, as anotações eram geradas pela união dos pontos pertencentes à imagem original, resultando em imprecisões.

A solução encontrada foi o desenvolvimento de um algoritmo que introduz pontos adicionais para cada anotação. Desta forma, quando a imagem é cortada, a anotação mantém uma quantidade suficiente de pontos, evitando desconfigurações. Essa abordagem ajudou a resolver a maioria dos problemas relacionados com a integridade das anotações, conforme ilustrado na Figura 57.

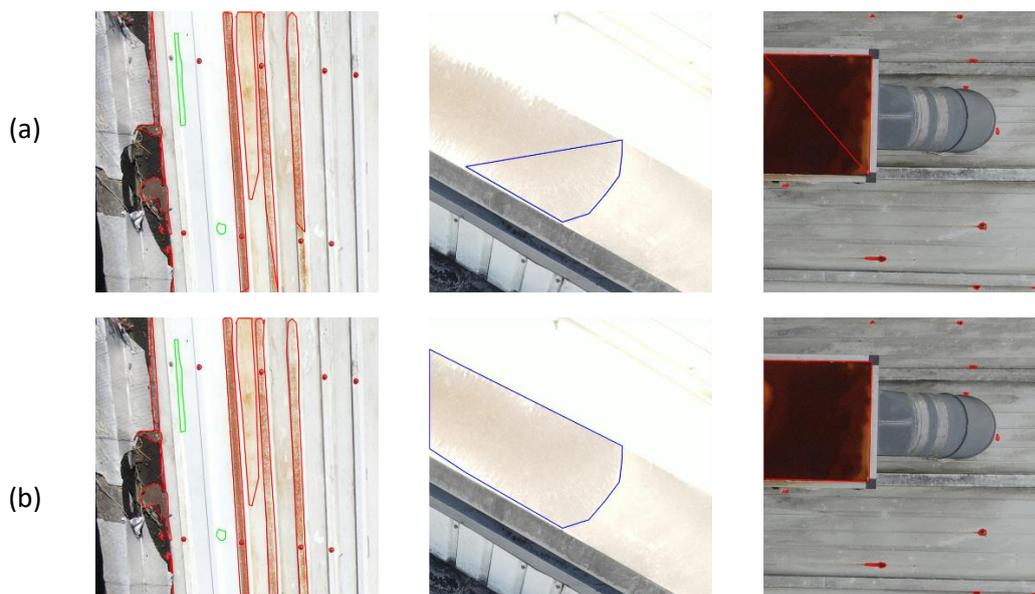


Figura 57 - Resultado da introdução de pontos (b) em anotações desconfiguradas (a)

Esta melhoria das etiquetas acabou por ter uma influência positiva em todas as métricas de treino, com subidas mais significativas no *Detectron2*, quando comparadas com os resultados anteriores. A precisão de IoU de 50% dos dois modelos, aproximaram-se, ficando com valores semelhantes, tal como apresentado na Tabela 6.

Tabela 6 - Resultados máximos dos treinos com anotações corrigidas

Métricas com imagens com correção das anotações (2000x2000)			
<i>Detectron2</i>		<i>YOLOv8</i>	
Métrica	Valor Máximo	Métrica	Valor Máximo
AP (det)	39,58%	AP (det)	81,34%
AP50 (det)	67,26%	AP50 (det)	70,70%
AP75 (det)	41,94%	AP50-95 (det)	44,23%
AP (segm)	35,35%	AP (segm)	76,14%
AP50 (segm)	63,97%	AP50 (segm)	63,24%
AP75 (segm)	33,99%	AP50-95 (segm)	32,47%

3.4.2.2 Data Augmentation

Devido à dificuldade de se obter uma base de dados extensa e do trabalho moroso associado à etiquetagem das imagens, no campo do ML e do DL, recorre-se a uma técnica denominada *Data Augmentation*. Esta técnica, tal como o nome indica, aumenta e diversifica a base de dados, sem haver a necessidade de anotar mais imagens. Neste caso em específico, as operações estão associadas a alterações em termos de brilho, contraste, rotação e saturação, tal como demonstrado na Figura 58.

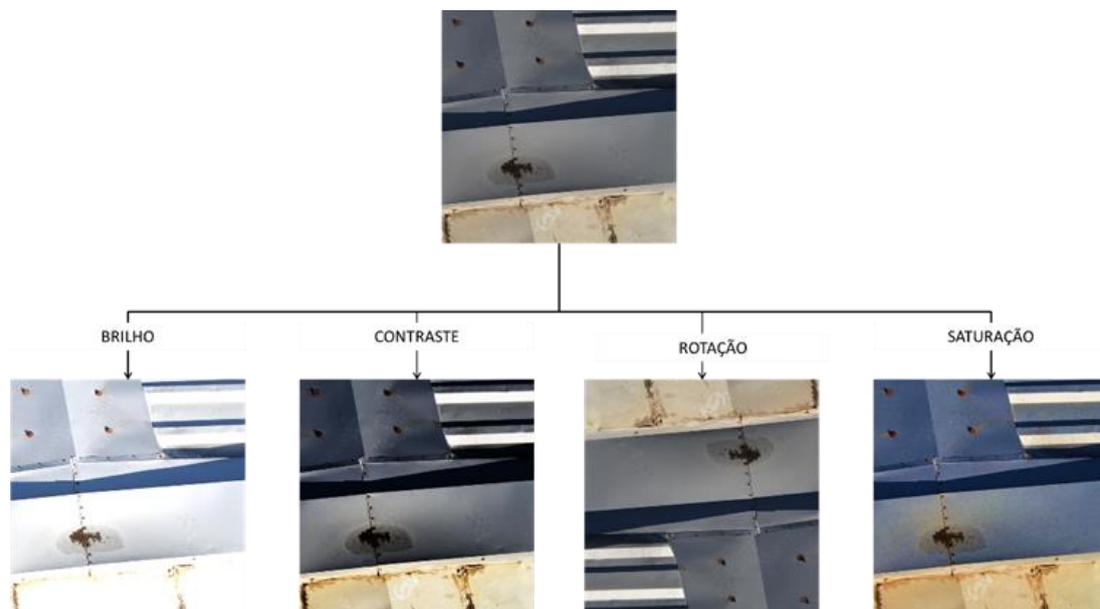


Figura 58 - Operações de *Data Augmentation*

As abordagens foram distintas para os dois modelos. Em relação ao *Detectron2*, o brilho, o contraste e a saturação foram definidos através da Equação (31), enquanto a rotação foi igual a 180°.

$$M_R = M_O(1 - \alpha) + M_T \cdot \alpha \quad (31)$$

onde M_R e M_O representam as imagens resultantes e originais, respetivamente; α representa o fator de intensidade escolhido e a aplicar; e M_T consiste na transformação aplicada.

A operação a realizar, assim como o fator de intensidade e a probabilidade de aplicação, são aspetos definidos pelo utilizador tendo em conta os seus intervalos, tal como apresentado na Tabela 7.

No que respeita ao *YOLOv8*, apesar de existir uma maior diversidade de operações, foram consideradas apenas as quatro seleccionadas no *Detectron2*, com o intuito de existir um termo de comparação entre os dois modelos. Em termos da aplicação das operações, no *Detectron2* existem duas possibilidades:

- **Seleção das operações no Roboflow:** antes da exportação da base de dados, é possível escolher as operações a serem aplicadas, o que permite visualizar o resultado da aplicação dessas operações e os respetivos valores seleccionados (Figura 59);

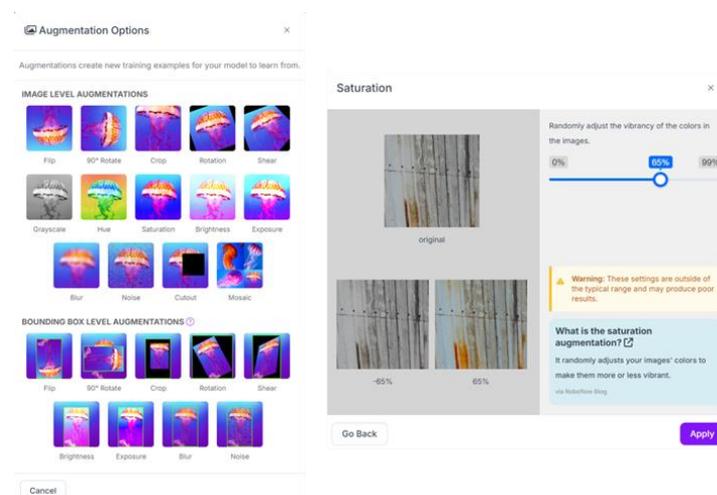


Figura 59 - Operações de *Data Augmentation* no *Roboflow*

- **Definição dos valores de brilho, contraste, saturação e rotação:** no ficheiro de exportação ("*data.yaml*"), foram definidos os valores correspondentes ao brilho (*hsv_v*), contraste (*hsv_h*), saturação (*hsv_s*) e rotação (*flipud*).

Tabela 7 - Intervalos e valores otimizados das operações de *Data Augmentation*

Operação	α	Probabilidade		Valores otimizados		
		<i>Detectron2</i>	YOLO	<i>Detectron2</i>		YOLO
				α	Probabilidade	Probabilidade
Brilho	[1,1 e 1,3]	[0,1 e 0,5]	[0,1 e 1,0]	1,1	0,5	0,4
Contraste	[1,2 e 1,4]			1,2	0,5	0,015
Saturação	[1,2 e 1,4]			1,2	0,5	0,7
Rotação	180°		[-180° e 180°]	180°	0,5	0°

Foram realizados vários treinos em ambos os modelos com o objetivo de avaliar o impacto das operações de *Data Augmentation*, ajustando os valores de cada operação. Os resultados das métricas mais elevadas estão registados na Tabela 8, enquanto a Tabela 7 apresenta os intervalos e os valores otimizados característicos das operações de *Data Augmentation*. É importante destacar que os resultados máximos do *YOLOv8* correspondem aos mesmos da Tabela 6, uma vez que os valores otimizados para as operações de *Data Augmentation* foram, na verdade, os valores padrão (*default*).

Tabela 8 - Resultados máximos dos treinos com *Data Augmentation* aplicado

Métricas com imagens com <i>Data Augmentation</i> aplicado			
<i>Detectron2</i>		<i>YOLOv8</i>	
Métrica	Valor Máximo	Métrica	Valor Máximo
AP (det)	38,92%	AP (det)	81,34%
AP50 (det)	67,55%	AP50 (det)	70,70%
AP75 (det)	39,95%	AP50-95 (det)	44,23%
AP (segm)	34,65%	AP (segm)	76,14%
AP50 (segm)	64,72%	AP50 (segm)	63,24%
AP75 (segm)	32,92%	AP50-95 (segm)	32,47%

3.4.2.3 Otimização de hiperparâmetros

Um aspeto essencial do treino está relacionado com a escolha adequada dos valores dos principais hiperparâmetros associados aos modelos, visando alcançar valores máximos de mAP50. Embora existam várias dezenas de hiperparâmetros disponíveis, tanto no *Detectron2* (Facebook AI Research, 2021) quanto no *YOLOv8* (ULTRALYTICS, 2023), foram selecionados os mesmos ou semelhantes, quando disponíveis, para proporcionar um termo de comparação entre os dois modelos. Os hiperparâmetros escolhidos foram:

- **Dimensão máxima e mínima das imagens:** refere-se aos tamanhos máximos e mínimos que as imagens podem ter durante o treino. Independentemente de haver um recorte,

todas as imagens são redimensionadas para estes valores antes de serem introduzidas no modelo. Esta configuração impacta a precisão do modelo e o custo computacional envolvido;

- **Número de *threads*:** trata-se da quantidade de threads de trabalho para carregamento de dados, influenciando a velocidade do pré-processamento e a alimentação do modelo;
- **Número de fases congeladas:** refere-se ao congelamento das primeiras n camadas da rede, impedindo que sejam treinadas. Essa prática é adequada para *fine-tuning* ou *transfer learning*.
- **Número de Rol's por imagem:** número de *Rol* por cada imagem iterada;
- **Limiar de sobreposição da *Region of Interest*:** limite de sobreposição de IoU, para que uma *Rol* seja considerada como *background* (caso a mesma seja menor que o valor definido) ou instância positiva (caso seja a mesma seja maior ou igual que o valor definido);
- **Número máximo de iterações/épocas:** consiste na quantidade de vezes que as amostras passam pelo algoritmo, otimizando os pesos sinápticos. No caso de *Detectron2* são iterações, já no *YOLOv8* são definidas épocas, sendo que cada uma realiza um número de iterações consoante o tamanho do *dataset*;
- **Paciência:** apenas aplicável no *YOLOv8*, representa o número de épocas sem melhorias nas métricas de validação, que fazem parar o treino antes do número de épocas definidas, evitando assim uma sobrecarga no custo computacional;
- **Taxa de aprendizagem:** velocidade de convergência do modelo, para a qual, no caso do *YOLOv8*, pode ser selecionado o valor inicial e final;
- **Momento:** representa a velocidade de alteração dos pesos sinápticos da rede para a convergência do modelo;
- **Período de validação:** intervalo de iterações para que as métricas de precisão sejam calculadas, como forma de validação do treino;

- **Dimensão do *batch*:** número de imagens fornecidas ao modelo, durante cada iteração. No *YOLOv8*, definindo o valor como -1, a dimensão do *batch* é gerada automaticamente, de acordo com o GPU disponível.

Tendo em conta todos estes hiperparâmetros, procedeu-se a uma otimização dos mesmos, diminuindo e aumentando o valor de cada um deles e percebendo o impacto que essas alterações tinham nas métricas de treino, tendo-se obtido os valores finais apresentados na Tabela 9.

Tabela 9 - Valores iniciais e otimizados para os treinos do *Detectron2* e do *YOLOv8*

Hiperparâmetro	Valores iniciais (<i>default</i>)		Valores otimizados (<i>best</i>)	
	<i>Detectron2</i>	YOLO	<i>Detectron2</i>	YOLO
Modelo	-	yolov8x-seg.pt	X_101_32x8d_FPN_3x	yolov8x-seg.pt
Dimensão máxima e mínima das imagens	1333x1333 e 800x800	640x640	2000x2000	1000x1000
Número de <i>threads</i>	4	8	4	8
Número de fases congeladas	2	None	2	None
Número de Rol's por imagem	512	-	400	-
Limiar de sobreposição da <i>ROI</i>	0,5	-	0,57	-
Número máximo de iterações/épocas	40000 iterações	100 épocas	40000 iterações	50 épocas
Paciência	-	100 épocas	-	10 épocas
Taxa de aprendizagem	0,001	0,01	0,001	0,001429 - > 0,0001
Período de validação	5000	-1	500	-1
Momento	0,9	0,937	0,9	0,9
Dimensão do <i>batch</i>	16	16	2	-1

As métricas obtidas nos treinos realizados com os valores dos hiperparâmetros otimizados presentes na Tabela 10, foram os valores mais elevados obtidos durante todo o trabalho e estão apresentados na Tabela 10. Embora as alterações nos hiperparâmetros não tenham sido

substanciais e, no caso do *Detectron2*, as melhorias nas métricas tenham sido ligeiras, a mudança das imagens de 640x640 *px* para 1000x1000 *px* no *YOLOv8* teve um impacto significativo, resultando num aprimoramento considerável nas métricas.

Tabela 10 - Valores máximos obtidos para o *Detectron2* e para o *YOLOv8* com os hiperparâmetros otimizados

Métricas com hiperparâmetros <i>otimizados</i>			
<i>Detectron2</i>		<i>YOLOv8</i>	
Métrica	Valor Máximo	Métrica	Valor Máximo
AP (det)	38,70%	AP (det)	78,50%
AP50 (det)	68,36%	AP50 (det)	73,30%
AP75 (det)	39,77%	AP50-95 (det)	45,44%
AP (segm)	34,54%	AP (segm)	74,90%
AP50 (segm)	64,98%	AP50 (segm)	68,32%
AP75 (segm)	32,34%	AP50-95 (segm)	35,75%

3.4.3 Teste e inferências

A fase de teste foi conduzida utilizando os modelos finais, que foram treinados com os melhores hiperparâmetros listados na Tabela 9. Os resultados obtidos refletiram as métricas mais altas, conforme apresentado na Tabela 10. À semelhança dos treinos, os testes do *Detectron2* e do *YOLOv8*, são avaliados através de métricas de precisão, que estão presentes na Tabela 11.

Tabela 11 - Valores máximos de teste obtidos para os melhores modelos

Métricas de testes e inferências			
<i>Detectron2</i>		<i>YOLOv8</i>	
Métrica	Valor Máximo	Métrica	Valor Máximo
AP (det)	43,47%	AP (det)	82,50%
AP50 (det)	72,55%	AP50 (det)	79,90%
AP75 (det)	45,06%	AP50-95 (det)	50,80%
AP (segm)	38,88%	AP (segm)	78,80%
AP50 (segm)	68,54%	AP50 (segm)	72,50%
AP75 (segm)	37,92%	AP50-95 (segm)	40,30%

Relativamente às inferências realizadas em imagens de teste (imagens que não constituem a base de dados de treino), estão demonstrados na Figura 60 alguns exemplos de segmentação realizados com os melhores modelos obtidos para o *Detectron2* e para o *YOLOv8*, mostrando também as anotações realizadas nessas mesmas imagens (*Ground Truth*). Estes cinco exemplos foram testados com confiança igual ou superior a 50% e podem ser divididos em:

- Inferências *multi damage* (corrosão e choque mecânico) bem sucedidas com todas as corrosões de diversos tamanhos e formatos identificadas, tal como demonstrado na Figura 60 (A);
- A Figura 60 (B) apresenta um caso em que existe uma diferença significativa relativamente à luz incidente na cobertura e conseqüentemente nos danos, mas mesmo assim tanto o *Detectron2* como o *YOLOv8*, conseguem identificar com precisão as corrosões e choques mecânicos;
- Semelhante ao exemplo anterior, em que existe sombra e sol no painel *sandwich*, a Figura 60 (C) apresenta um exemplo em que as corrosões possuem pequenas dimensões. Apesar de segmentar quase todas as corrosões, o modelo do *YOLOv8* mostrou ser menos eficaz na zona com sombra;

- Uma segmentação com os três tipos de danos trabalhados (corrosão, choque mecânico e acumulação de água) surge na Figura 60 (D) com precisões elevadas;
- Por fim, surge a Figura 60 (E) que demonstra a qualidade de segmentação dos dois modelos ao identificar apenas danos (corrosões neste caso) nos painéis sandwich e descartando esses mesmos danos fora da envolvente do ativo.

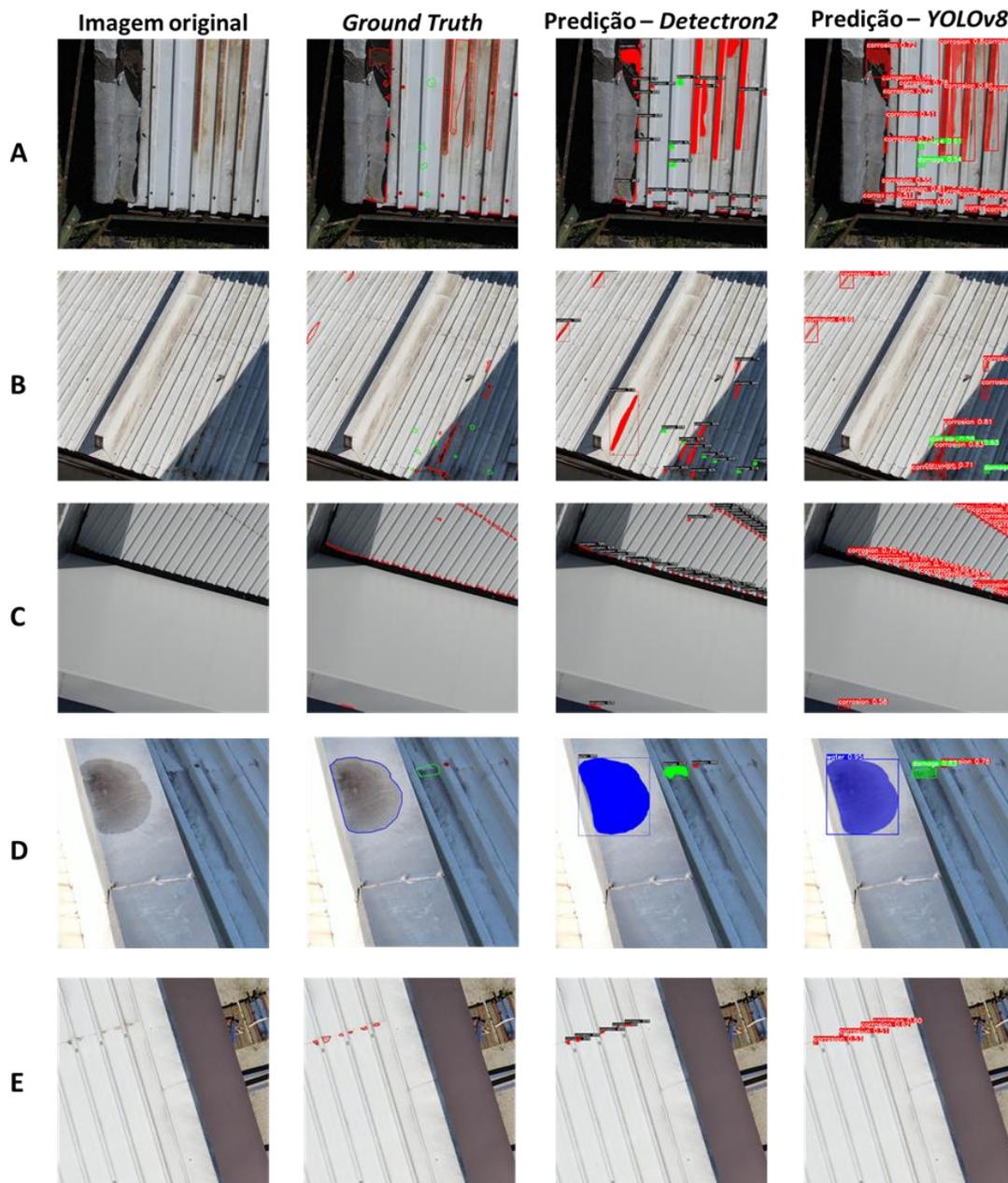


Figura 60 - Exemplos de inferências dos modelos *Detectron2* e *YOLOv8*

3.5 Ray Casting

A escassez de informações sobre a tecnologia de *Ray Casting* e a sua aplicabilidade na Engenharia Civil levou ao desenvolvimento de uma metodologia destinada à obtenção de um modelo fotogramétrico com patologias, sem qualquer base prévia. Essa abordagem resultou num processamento caracterizado por uma divisão em fases detalhadas, como ilustrado na Figura 61.



Figura 61 - Metodologia de *Ray Casting*

1. **Fotogrametria:** este primeiro passo, ainda anterior ao processo de *Ray Casting*, é essencial para o seu funcionamento. Para o efeito, recorreu-se ao *iTwin Capture Modeler Master* com o objetivo de exportar uma nuvem de pontos, uma *mesh* e ainda um ficheiro *.xml* que contém informações relativas às imagens como:
 - Dimensões da imagem (altura e largura);
 - Distância focal;
 - Pontos principais (x e y);
 - Parâmetros de distorção (k1, k2, k3, p1 e p2);
 - Parâmetros de rotação (M_00-M_02, M_10-M_12, M_20-M_22, cx, cy e cz).

O processamento das imagens dos casos de estudo (pala de um edifício industrial da NACEX e quadro com alvos) resultou na obtenção da nuvem de pontos (*point cloud*) e da malha (*mesh*) – no caso da pala do edifício industrial na Figura 51 e na Figura 48, e no caso do quadro com alvos na Figura 62 - sendo este último utilizado apenas para validação da etapa final (número 11 da Figura 61).

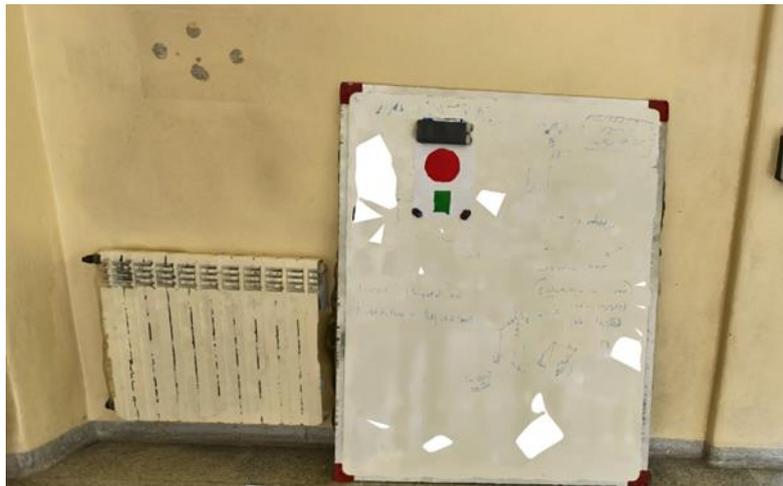


Figura 62 - Mesh do quadro com alvos

2. **Retificação das imagens:** a câmara do UAV DJI M3M tem uma lente *fisheye*, que provoca distorções nas imagens capturadas, resultando em desvios na projeção de 2D para 3D. Para resolver essa questão, foi desenvolvido um código que corrige a distorção com base no posicionamento e nos parâmetros das imagens obtidas por qualquer câmara de *drone*, conforme ilustrado na Figura 63.



Figura 63 - Exemplos de remoção da distorção em imagens

Para além da distorção, outra técnica de retificação aplicada às imagens foi a equalização. Utilizando a função *Contrastive Limited Adaptive Equalization*, buscou-se reduzir o brilho das

imagens. Esta equalização permite realçar os choques mecânicos, diminuir significativamente as sombras e equilibrar os brancos, especialmente em relação à reflexão solar, como demonstrado na Figura 64.



Figura 64 - Exemplos de equalização em imagens

3. **Realização de inferências:** este passo baseia-se na aplicação do melhor modelo obtido durante os treinos do *Detectron2* para realizar inferências nas imagens utilizadas no processamento da nuvem de pontos e da *mesh*. Na Figura 65 são apresentados dois exemplos de imagens que passaram pelo processo de inferência. Esta etapa envolve procedimentos como o recorte das imagens em 2000×2000 *px*, a realização de inferências nas imagens recortadas e, por fim, a reconstrução da imagem original a partir das imagens de 2000×2000 *px* que foram inferidas. Um aspeto a realçar é o facto de ter sido necessário realizar uma adaptação que permitisse o bom funcionamento da tecnologia para qualquer tipo de imagem. Como as imagens provenientes do DJI M3M têm uma resolução de 5280×3956 *px*, foi introduzido um passo intermédio que redimensiona todas as imagens de entrada para 8000×6000 *px* e posteriormente realiza o recorte das mesmas.

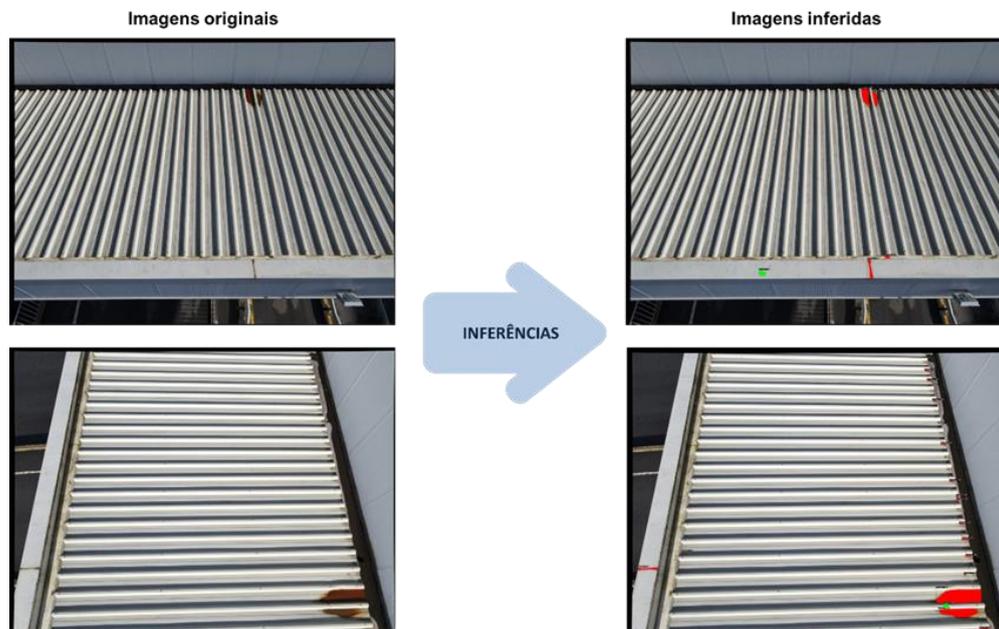


Figura 65 - Exemplos de imagens inferidas

4. **Ray Casting inicial:** o *Ray Casting* inicial baseia-se na projeção dos danos resultantes das inferências na *mesh* do modelo fotogramétrico. Para isso, é necessário remover o *background* das imagens através de um filtro RGB que mantém as cores vermelha (255,0,0), verde (0, 255,0) e azul (0, 0, 255) puras (cores definidas para as inferências dos três danos), tal como demonstra o exemplo da Figura 66. Posto isto, as imagens são projetadas e as cores das máscaras interseitam os pontos correspondente da *mesh*, gerando uma nuvem de pontos só de inferências, que ao sobrepor-se à nuvem de pontos, forma um modelo fotogramétrico com patologias (Figura 67).



Figura 66 - Remoção do *background* com filtro RGB

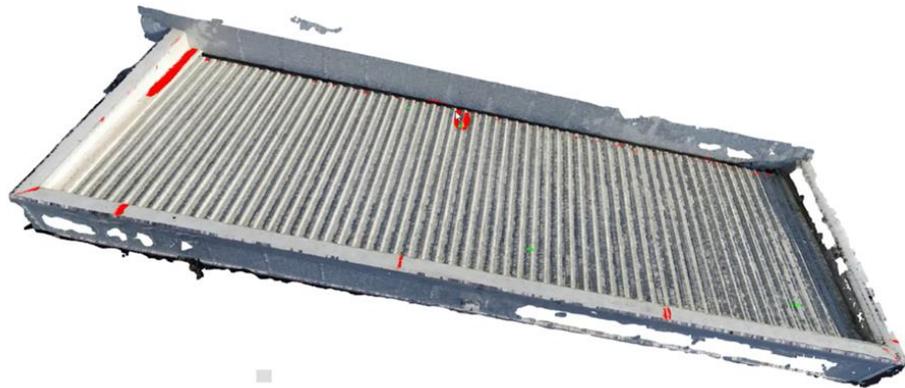


Figura 67 - Resultado do Ray Casting inicial

5. **Clustering dos danos:** recorrendo à técnica de agrupamento *DBSCAN*, os danos presentes são divididos por *clusters* com base na densidade dos pontos. Para cada um destes *clusters* é retirado um centróide (x,y,z) , assim como uma reta normal a este centróide. Esta técnica depende da definição de dois parâmetros: o número mínimo de pontos necessários para que eles se unam e formem um *cluster* (*minPontos*) e a distância máxima entre dois pontos para que sejam considerados vizinhos (*eps*). Para este trabalho, foram escolhidos os valores de 1 e 0,05, respetivamente. Posto isto, a técnica assume um ponto central e avalia os pontos de fronteira (caso a distância para o ponto central seja menor ou igual que a *eps*) e os pontos de ruído (caso a distância para o ponto central seja maior que a *eps*), tal como explicado na Figura 68 (a). O resultado da aplicação da *DBSCAN* na pala do edifício industrial está representado na Figura 68 (b) (Pandey, 2020).

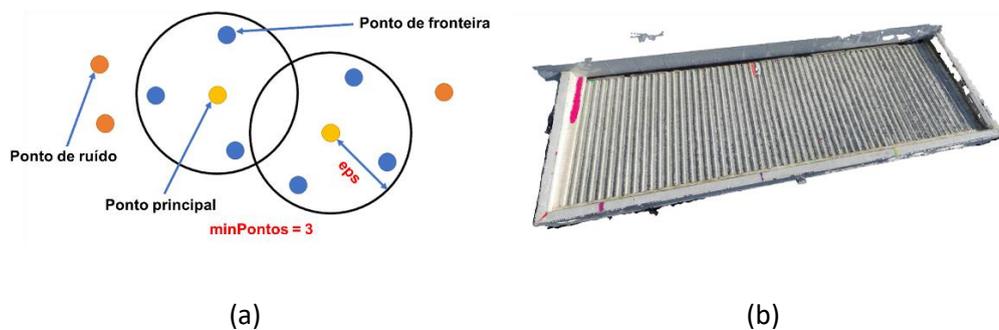


Figura 68 - Explicação da *DBSCAN* (a) (Pandey, 2020) e resultado na pala do edifício industrial da NACEX (b)

6. **Melhor imagem por dano:** com o intuito de obter os danos com a maior precisão e confiança possível, foi concebido um processo de seleção das cinco melhores imagens por cada dano. Essa seleção apoia-se em três critérios:

- O primeiro critério prende-se com a qualidade das imagens, isto é, só avançam para o segundo critério imagens com um valor de contraste acima de 3.5;
- O segundo critério consiste na seleção das quinze imagens mais próximas do dano, levando em consideração o centróide do dano e as coordenadas associadas a cada imagem. Este processo é ilustrado na Figura 69 (a), onde são apresentadas as quinze posições (representadas por círculos vermelhos) relativas a um dano específico (indicado pela triângulo verde);
- O terceiro e último critério (Figura 69 (b)) define as cinco imagens mais ortogonais (bolas vermelhas) ao dano (bola verde).

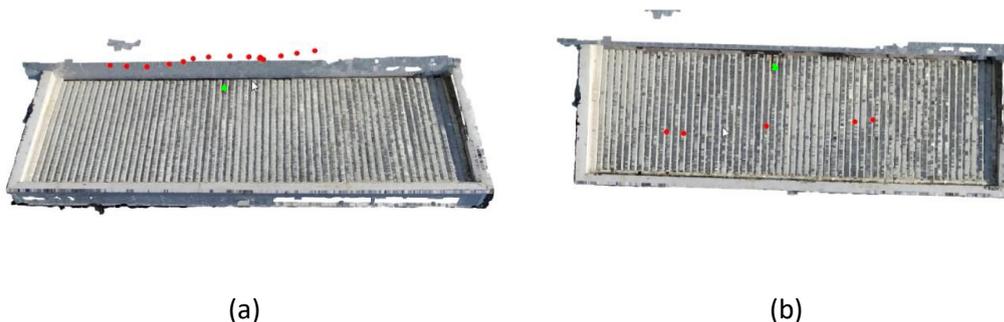
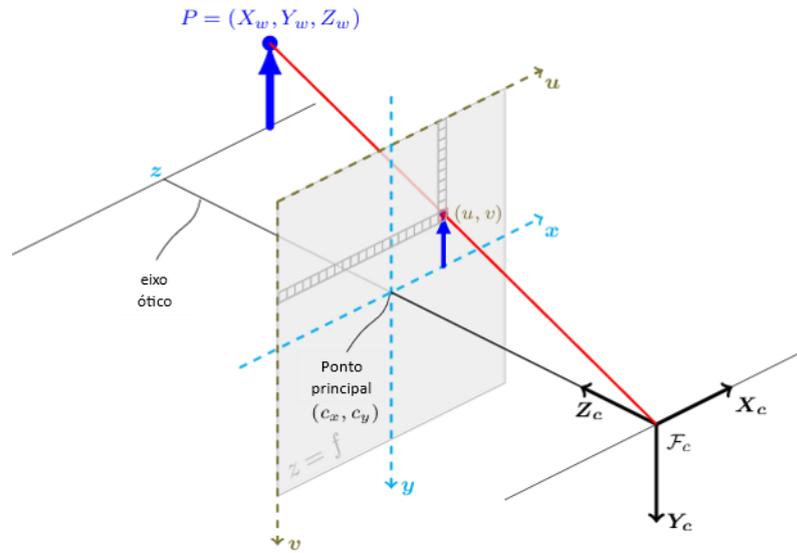


Figura 69 - Exemplos das quinze imagens mais próximas (a) e das cinco imagens mais ortogonais (b) a um determinado dano específico

7. **Projeção do centróide (3D→2D):** de forma a obter o tipo do dano, foi necessário projetar as coordenadas 3D dos centróides de cada dano e convertê-las para o plano 2D da imagem. A Figura 70 ilustra a projeção de um ponto 3D (P_w) numa imagem, representada pelo ponto $p(u, v)$. Esta projeção é concretizada através da interseção entre uma reta que une o centro da câmara (P_c) e o ponto 3D (P_w) e um plano que representa a imagem (Doxygen, 2024).


 Figura 70 - Modelo de câmara *pinhole* (Doxygen, 2024)

As funções utilizadas recorrem ao modelo de câmara *pinhole* em que a transformação projetiva sem distorção se calcula através das equações (32) e (33).

$$s p = A [R|t] P_w \quad (32)$$

$$p = A P_c \quad (33)$$

onde s representa a escala arbitrária da transformação, A (ver também Equação (34)) é a matriz intrínseca da câmara, R e t representam a rotação e a translação, respetivamente, e P_c engloba o sistema de coordenadas da câmara.

$$A = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (34)$$

onde f_x e f_y compõem a distância focal e o ponto principal é representado pelas coordenadas c_x e c_y . Resultando na Equação (35):

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} \quad (35)$$

onde u e v são as coordenadas de p e X_c, Y_c e Z_c as de P_c .

A matriz $[R|t]$ consiste no produto matricial entre uma transformação projetiva e uma transformação homogênea. A transformação projetiva 3 por 4 mapeia pontos 3D representados em coordenadas da câmara para pontos 2D no plano da imagem e representados em coordenadas de câmara normalizadas ($x' = \frac{X_c}{Z_c}$ e $y' = \frac{Y_c}{Z_c}$), tal como demonstrado na Equação (36):

$$Z_c \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1000 \\ 0100 \\ 0010 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} \quad (36)$$

Por sua vez, a transformação homogênea é composta pelos parâmetros extrínsecos R e t e corresponde à mudança do sistema de coordenadas mundial (w) para o da câmara (c), segundo a Equação (37):

$$P_c = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} P_w \Leftrightarrow \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11}r_{12}r_{13}t_x \\ r_{21}r_{22}r_{23}t_y \\ r_{31}r_{32}r_{33}t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \quad (37)$$

A combinação destas duas transformações, equivale à transformação projetiva dos pontos 3D em coordenadas mundiais em pontos 2D num plano de uma imagem, tendo em consideração as coordenadas da câmara (Equação (38)):

$$Z_c \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = [R|t] \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11}r_{12}r_{13}t_x \\ r_{21}r_{22}r_{23}t_y \\ r_{31}r_{32}r_{33}t_z \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \quad (38)$$

A junção das equações para parâmetros intrínsecos e extrínsecos ($s p = A [R|t]P_w$), resulta na Equação (39):

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11}r_{12}r_{13}t_x \\ r_{21}r_{22}r_{23}t_y \\ r_{31}r_{32}r_{33}t_z \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \quad (39)$$

Como $Z_c \neq 0$, a Equação (39), fica como a Equação (40):

final é o valor mais elevado da confiança da instância, garantindo que a imagem escolhida seja a que apresenta a melhor representação do dano identificado.

10. **Ray Casting final:** Nesta fase, apenas a imagem selecionada no passo anterior é utilizada, caracterizando-se por um *Ray Casting* isolado ou por instância. Assim, é escolhida a melhor imagem para cada um dos *clusters*, permitindo uma análise mais precisa e detalhada dos danos identificados, tal como demonstrado na Figura 73.



Figura 73 - Exemplo de corrosão em *Ray Casting* final

11. **Gestão de dados:** de forma a organizar toda a informação, é gerada uma lista que serve como uma gestão dos dados. Nesta lista, são incorporadas todas as informações necessárias, conforme o desejado pelo utilizador. Em seguida, apresenta-se um exemplo do que poderá ser considerado, desde o caminho da melhor imagem original e tratada para cada *cluster*, até ao centróide do dano, as medidas x e y do dano, a área e o tipo do dano, tudo disposto numa tabela em *MS-Excel*.

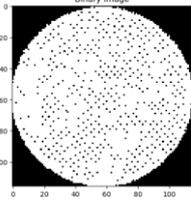
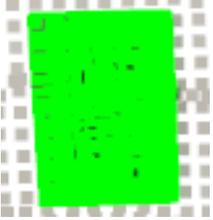
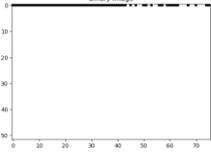
Com o intuito de validar a obtenção das medidas x e y e a área do dano, foi realizada uma experiência em ambiente controlado, mais concretamente com um quadro que continha alvos de medidas conhecidas, como um círculo e um quadrado (Figura 62). Para tal é necessário:

- Proceder a uma filtragem dos pontos de ruído pertencentes aos clusters;
- Recorrer a uma técnica (RANSAC) que estabelece a melhor representação plana do cluster, facilitando assim a análise 2D;
- Usar o *Principal Component Analysis* (PCA) de modo a encontrar os eixos

principais;

- Criar uma imagem binária, através dos pontos projetados que são previamente escalados e normalizados para *pixels*;
- Calcular as medidas (Tabela 12) com recurso a multiplicação entre as dimensões das imagens binárias e as resoluções calculadas anteriormente.

Tabela 12 - Resultados das medidas da experiência em ambiente controlado

Cluster	Imagem binária	Valores reais	Valores obtidos	Erro (%)
		$x = 12.80 \text{ cm}$	$x = 12.99 \text{ cm}$	1.5%
		$y = 12.80 \text{ cm}$	$y = 13.02 \text{ cm}$	1.7%
		$A = 128.7 \text{ cm}^2$	$A = 128.2 \text{ cm}^2$	0.4%
		$x = 8.70 \text{ cm}$	$x = 8.58 \text{ cm}$	1.4%
		$y = 5.85 \text{ cm}$	$y = 5.80 \text{ cm}$	0.9%
		$A = 50.9 \text{ cm}^2$	$A = 50.6 \text{ cm}^2$	0.5%

Tal como demonstrado na Tabela 12, as percentagens de erro não excedem os 1,7%, satisfazendo assim as necessidades de precisão do projeto, que visa avaliar a área correspondente a cada tipo de dano.

4 Aplicação da metodologia a um ativo industrial

Com o objetivo de validar toda a metodologia concebida, optou-se por um caso de estudo de um ativo industrial com uma área significativa de painéis *sandwich* danificados, focando-se especialmente em corrosões e choques mecânicos. Esta escolha permitiu aplicar a metodologia em contexto real, proporcionando uma análise detalhada e prática da eficácia das técnicas desenvolvidas na deteção e avaliação dos danos.

4.1 Memória descritiva

O activo em estudo (Figura 74) corresponde a um edifício com fins industriais, que se encontra devoluto, localizado na cidade de Santo Tirso, mais concretamente na zona industrial do Arco. Caracteriza-se por ter um perímetro de aproximadamente 245 m, uma área de 3000 m² e uma altura máxima de 16.50 m. A sua envolvente, que inclui a cobertura e as quatro fachadas, é inteiramente revestida por painéis *sandwich*.



Figura 74 - Imagem geral do ativo industrial em estudo

4.2 Aquisição de dados

À semelhança do que aconteceu com as restantes campanhas experimentais, foi necessário obter dados do edifício com recurso a um TLS (*BLK360 G1*) e a dois UAV's (*DJI M2EA* e *DJI M3M*).

4.2.1 BLK360 G1, Cyclone Field e iPad

Seguindo as boas práticas do uso do *BLK360 G1*, previamente à utilização do equipamento em campo, é necessário definir a quantidade e a localização das estações a realizar e dos alvos a georreferenciar. Devido à distância e aos custos associados a uma visita ao local para uma tarefa desta natureza, decidiu-se proceder a esta marcação das estações e dos alvos através do *Google Maps*. Conforme apresentado na Figura 75, as estações foram posicionadas estrategicamente em um padrão de *zig-zag*, com espaçamento de 10 a 15 metros, considerando as dimensões do ativo e os acessos ao seu redor. Procurando maximizar a eficiência da campanha e otimizar o tempo, prevendo-se a necessidade de se realizar 33 estações. Para além disso, com a ajuda do topógrafo foi definida a necessidade da fixação de 19 alvos.



Figura 75 - Localização das estações e dos alvos

O recurso ao *BLK360 G1* dividiu-se em três fases, representadas na Figura 76:

- a) Montagem do equipamento, verificando se o mesmo está em segurança para entrar em funcionamento;
- b) Colocação e fixação dos *B&W targets* nas localizações estabelecidas;
- c) Realização dos levantamentos digitais, seguindo o mapeamento realizado para os locais das estações.



(a)



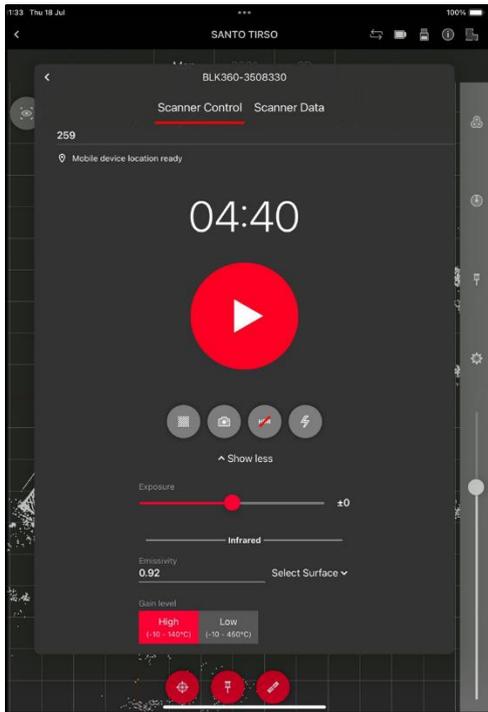
(b)



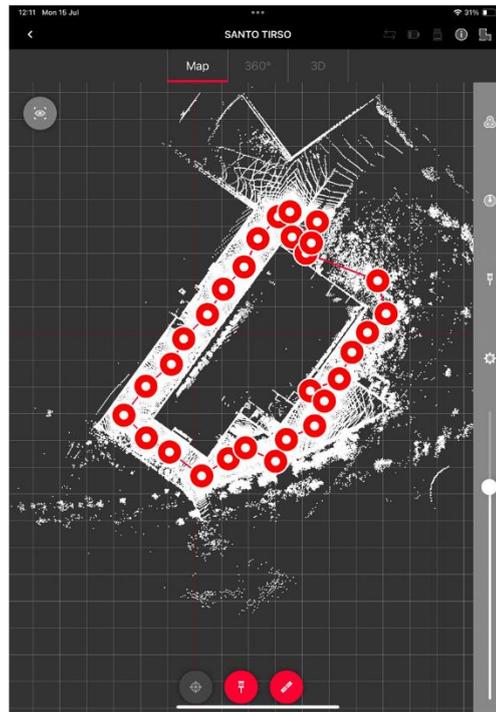
(c)

Figura 76 - Montagem do *BLK360 G1* (a), colocação dos alvos (b), localização das estações (c)

A utilização do *Cyclone Field 360* em conjunto com o *iPad* durante a operação do *BLK360 G1* facilita não apenas o trabalho em campo, mas também o processamento posterior dos dados. Esta abordagem permite definir diversos parâmetros dos levantamentos, como a escolha da densidade, a ativação da captura de imagens, HDR e *flash*, conforme ilustrado na Figura 77 (a). A campanha terminou ao fim de 4 horas, resultando num total de 29 estações interligadas através de *links* (Figura 77 (b)), com a nuvem de pontos resultante ilustrada na Figura 78.



(a)



(b)

Figura 77 - Definição dos parâmetros dos *scans* (a) e resultado da campanha (b)



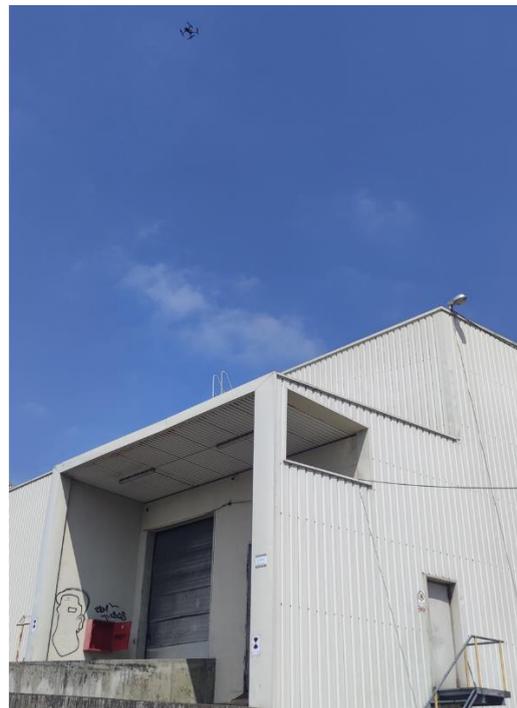
Figura 78 - Nuvem de pontos resultante do pré-processamento do levantamento efetuado com o *BLK360 G1*

4.2.2 DJI M2EA e DJI M3M

A operação com os *drones* foi acompanhada dos respetivos comandos de controlo (Figura 79 (a)), que permitem a definição de uma área de voo, de modo a realizar uma captura de imagens e um voo de forma automática, garantindo a sobreposição desejada das imagens. A captura das imagens da cobertura foi realizada através de um voo automático (Figura 79 (b)), no entanto, a localização do ativo industrial com outros armazéns e vegetação na sua envolvente, fez com que não fosse possível o voo automático nas fachadas.



(a)



(b)

Figura 79 - Utilização do comando de controlo (a) e voo automático (b)

O levantamento com drones ficou marcado pela presença de gaiivotas que dificultaram o voo, fazendo com que a campanha se estendesse por mais de 5 horas, capturando um total de 7769 imagens. Na Figura 80 são apresentados 4 exemplos de imagens do ativo.



Figura 80 - Exemplos de imagens do ativo capturadas pelos drones

4.3 Processamento de dados e criação da nuvem de pontos e da *mesh*

O processamento de dados do *BLK360 G1* é realizado no *software* da *Leica Cyclone Register 360*, enquanto as imagens que compõem a fotogrametria são processadas no *iTwin Capture Modeler*. O recurso ao *CloudCompare* surge com o intuito de se obter a fusão entre as duas nuvens de pontos geradas. A presente secção apresenta os passos do processo do caso do estudo e os respetivos resultados.

4.3.1 *Cyclone Register 360*

O processamento no *Cyclone Register 360* baseou-se nas seguintes 4 etapas:

1. **Melhoria dos *links*:** devido à geometria do ativo industrial e da distância entre estações, este aprimoramento prendeu-se com a adição de apenas 13 *links*, resultando num total de 42 *links* quando comparado com o trabalho em campo, tal como é possível observar na Figura 81;

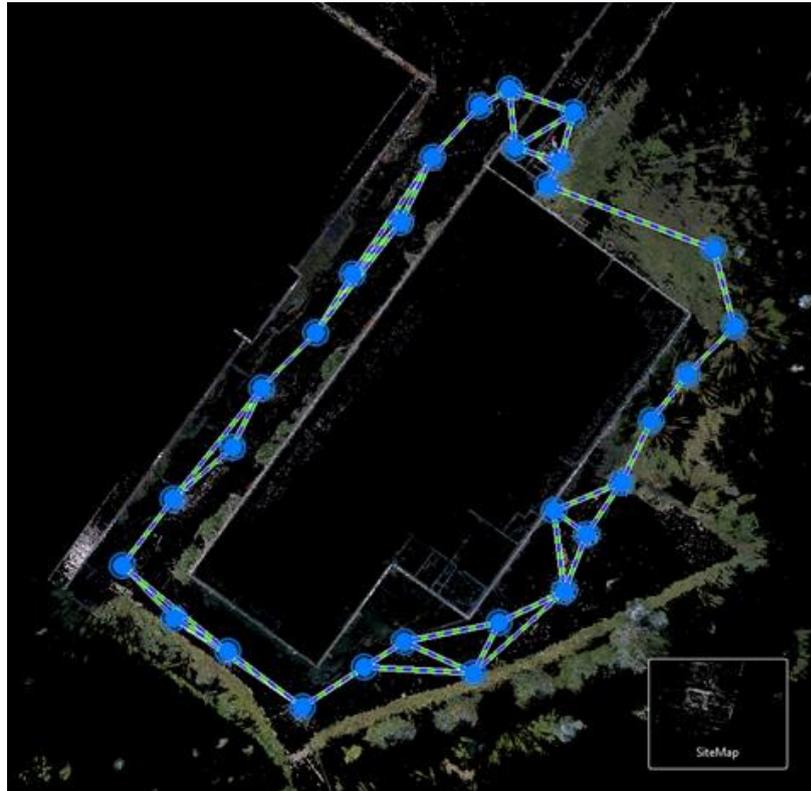
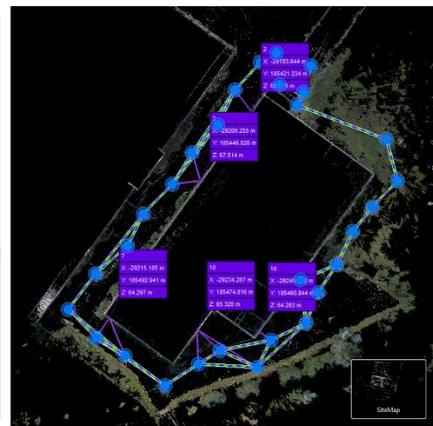


Figura 81 - Melhoria dos links do caso de estudo no Cyclone Register 360

2. **Georreferenciação:** foram marcados e georreferenciados os 19 B&W targets no software (Figura 82 (a)), mas para a criação da nuvem de pontos apenas foram considerados os GCP's com menos de 0.01 m de erro (Figura 82 (b));



(a)



(b)

Figura 82 - Marcação dos alvos (a) e identificação dos GCP's utilizados (b)

3. **Limpeza e controlo de erros:** após a limpeza da nuvem de pontos, não só a nível de visualização, mas também em termos de *Cloud to Cloud*, a nuvem de pontos ficou com um *Bundle Error* de 0.003 m, um *Overlap* de 64% e ainda uma *Strength* de 73% (Figura 83);

Bundle Error 0.003 m ✓	
Overlap 64 % ✓	Strength 73 % ✓
Cloud-to-Cloud	Target Error

Figura 83 - Excerto do relatório final do *Cyclone Registe 360*

4. **Exportação:** terminado o processamento dos dados da tecnologia LiDAR, procedeu-se à exportação do resultado da nuvem de pontos, apresentado na Figura 84. O tipo de ficheiro escolhido para a exportação foi o *.las*, ficando com cerca de 4.22 GB.



Figura 84 - Resultado da nuvem de pontos do *BLK360 G1*

4.3.2 *iTwin Capture Modeler*

No software *iTwin Capture Modeler*, o processamento dos dados provenientes dos drones foi estruturado em três etapas. A primeira delas envolve a georreferenciação dos alvos (Figura 85), onde todos os GCP's levantados pelo topógrafo foram marcados. Para este processo, utilizou-se o sistema referencial de coordenadas ETRS89 / Portugal TM06 (EPSG:3763) juntamente com a altura do geóide EGM96 (EPSG:5773).

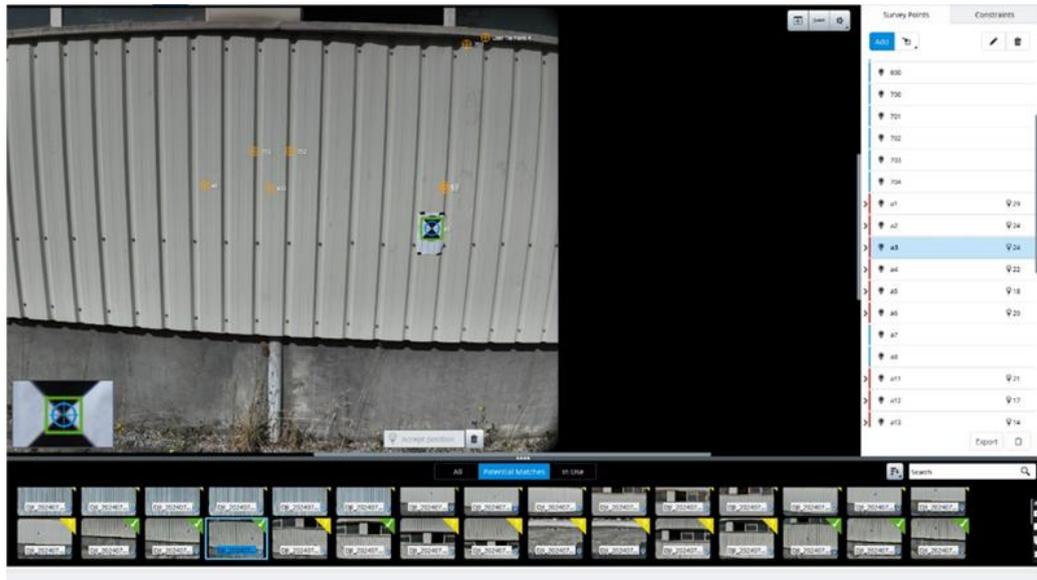
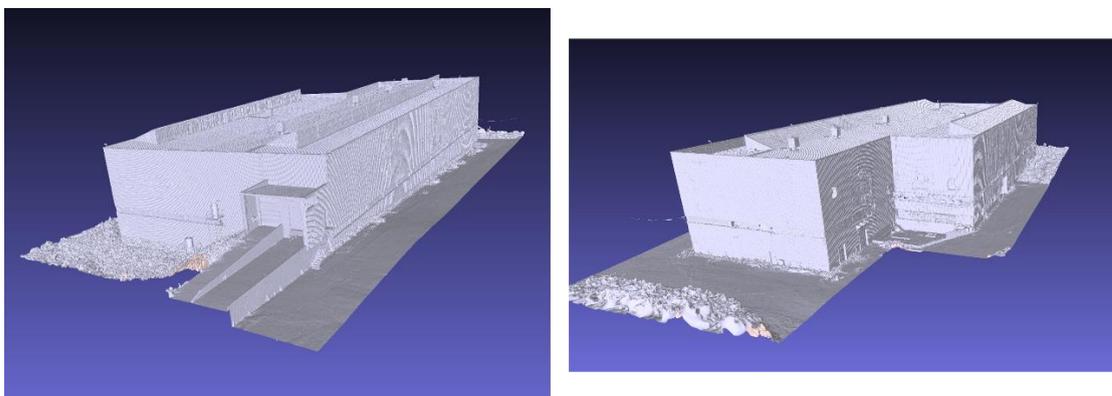


Figura 85 - Georreferenciação dos alvos no *iTwin Capture Modeler*

Posteriormente, são definidos os parâmetros de aerotriangulação das imagens tendo em consideração os *Ground Control Points* georreferenciados. Relativamente à criação da nuvem de pontos, destacou-se pela necessidade de alteração do valor da *point sample distance* de 1 pixel (*default*) para 0.01 m, com o objetivo de não gerar um ficheiro *.las* muito pesado e impossível de trabalhar, ficando com 1.32 GB.

Por fim, a *mesh* criada (4.6 GB) não teve qualquer tipo de alteração em termos de parâmetros *default* do *iTwin Capture Modeler*, resultando no modelo presente na Figura 86 (a) e (b).



(a)

(b)

Figura 86 - Perspetivas da *mesh* do caso de estudo resultante do *iTwin Capture Modeler*

4.3.3 CloudCompare

Após a obtenção das nuvens de pontos geradas pela tecnologia LiDAR, utilizando o *BLK360 G1* (Figura 87 (a)), e da nuvem de pontos resultante da fotogrametria realizada com UAVs (Figura 87 (b)), foi possível realizar uma análise comparativa entre as duas abordagens. Essa comparação permitiu avaliar a precisão e a eficácia de cada método na captura de dados espaciais, contribuindo para uma melhor compreensão das características do ativo em estudo.

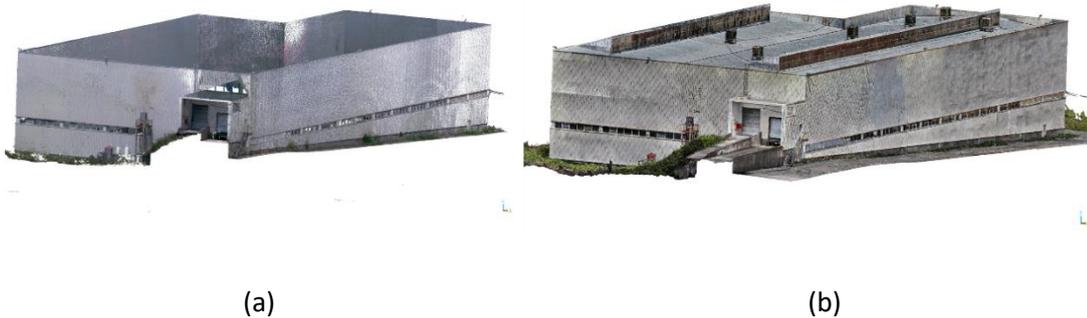


Figura 87 - Nuvens de pontos do LiDAR (a) e da fotogrametria (b) do caso de estudo

O processamento no *CloudCompare*, baseou-se na aplicação da tecnologia ICP, com o intuito de fundir as duas nuvens de pontos apresentadas anteriormente na Figura 87, resultando no modelo com 11.56 GB que surge na Figura 88.



Figura 88 - Nuvem de pontos do caso de estudo final resultante da ICP

4.4 Aplicação do modelo de identificação de anomalias

A presente secção tem como objetivo demonstrar a eficácia dos melhores modelos treinados, tanto do *Detectron2* quanto do *YOLOv8*, através da realização de inferências em várias imagens do ativo industrial, capturadas por drones. A Figura 89, a Figura 90 e a Figura 91 ilustram alguns exemplos dessas inferências em planos diferentes (fachadas e cobertura), evidenciando a capacidade dos modelos em identificar e classificar os danos presentes nas imagens.

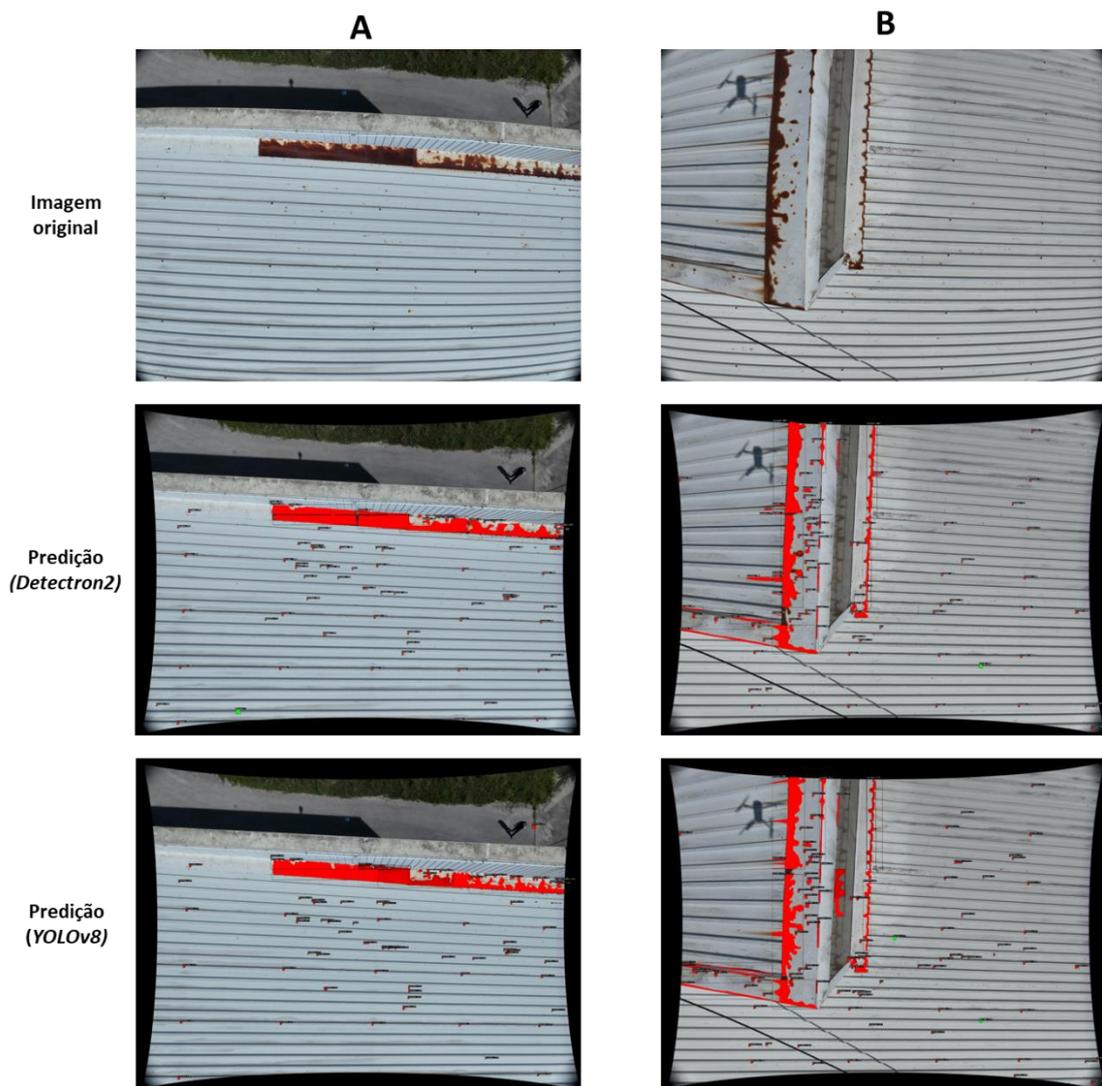


Figura 89 - Exemplos de predições *Detectron2* e *YOLOv8* em imagens da cobertura do caso de estudo

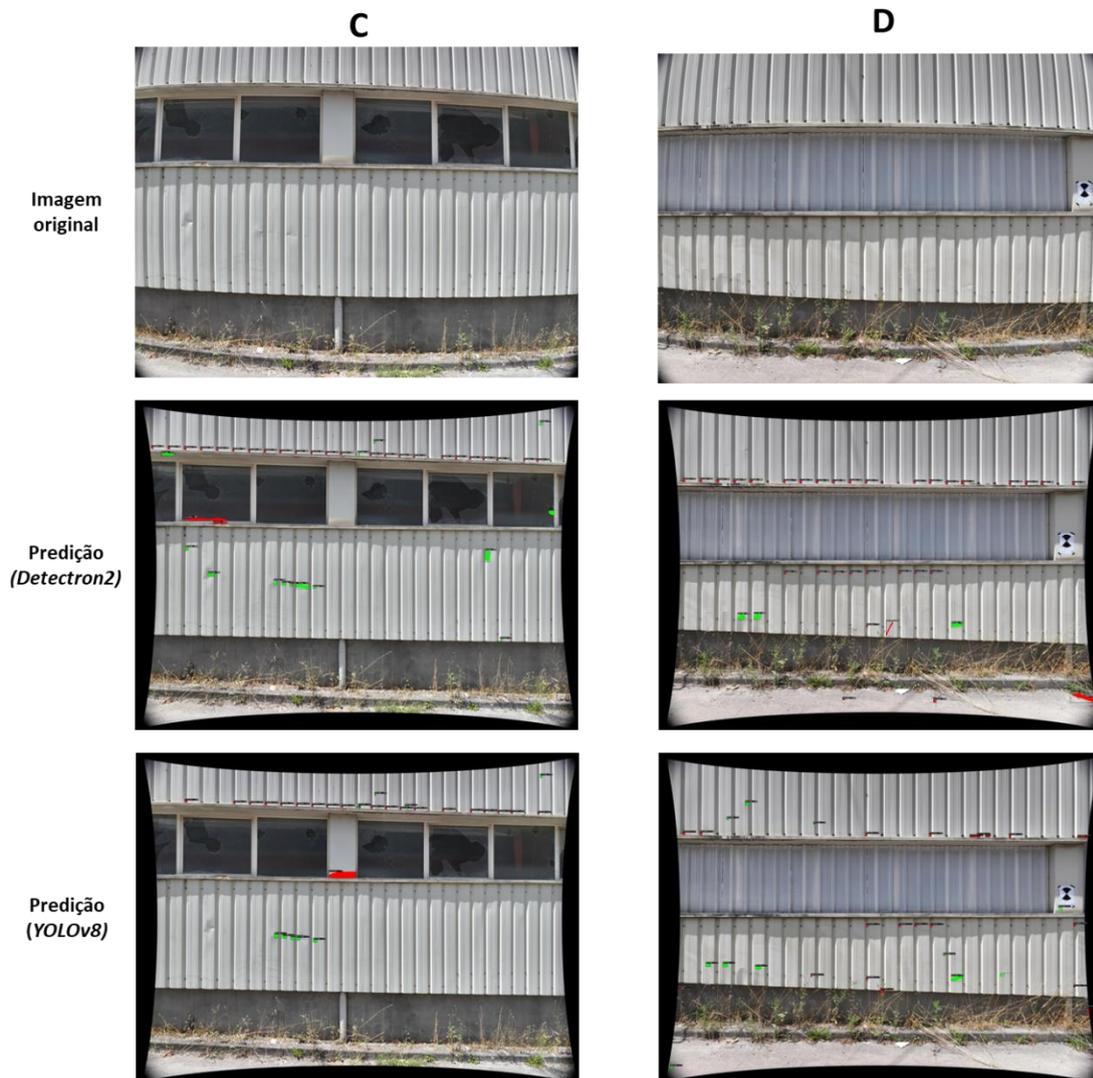


Figura 90 - Exemplos de predições *Detectron2* e *YOLOv8* em imagens da fachada Este do caso de estudo

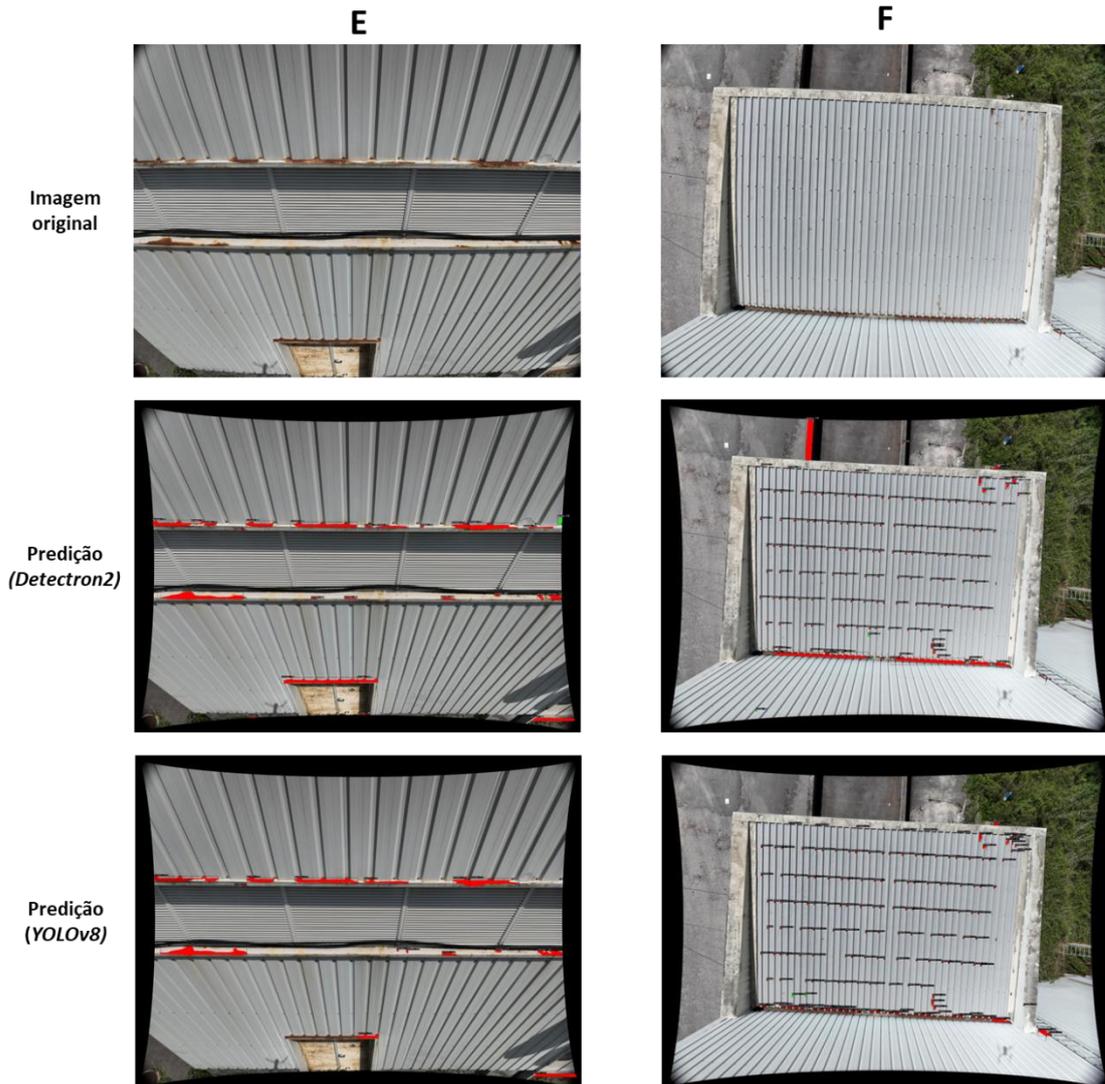


Figura 91 - Exemplos de predições *Detectron2* e *YOLOv8* em imagens da fachada Oeste e Sul do caso de estudo

4.5 Modelo fotogramétrico com patologias

Por fim, a etapa final consiste no desenvolvimento de um modelo fotogramétrico do ativo industrial (nuvem de pontos) com patologias, aplicando a técnica de projeção do *Ray Casting*. Neste modelo as anomalias identificadas são identificadas com as cores vermelhas, verdes e azuis, de modo a localizar e a facilitar a visualização dos diferentes tipos de anomalias, corrosão, choques mecânicos e acumulação de águas, respetivamente.

A maior dificuldade da aplicação do *Ray Casting* prendeu-se com as imitações dos meios computacionais para suportar um processo completo desta natureza com uma nuvem de pontos e uma *mesh* de grande escala (cerca de 20 GB). Assim, a solução passou por dividir a estrutura em 5 partes, a cobertura e as 4 fachadas, aplicando a técnica em separado cada um destes elementos. Seguidamente são apresentados os resultados finais, divididos por zonas para melhor visualização, assim como indicadores que caracterizam cada uma das anomalias.

- **Fachada Norte**



Figura 92 - Modelo da fachada Norte

A Figura 92 apresenta o modelo da fachada Norte com um pormenor de uma zona com inferências de corrosão em 3 parafusos e na esquadria metálica da porta. Esta fachada possui 0,279 m² de corrosões e 0,022 m² de choques mecânicos.

- Fachada Este

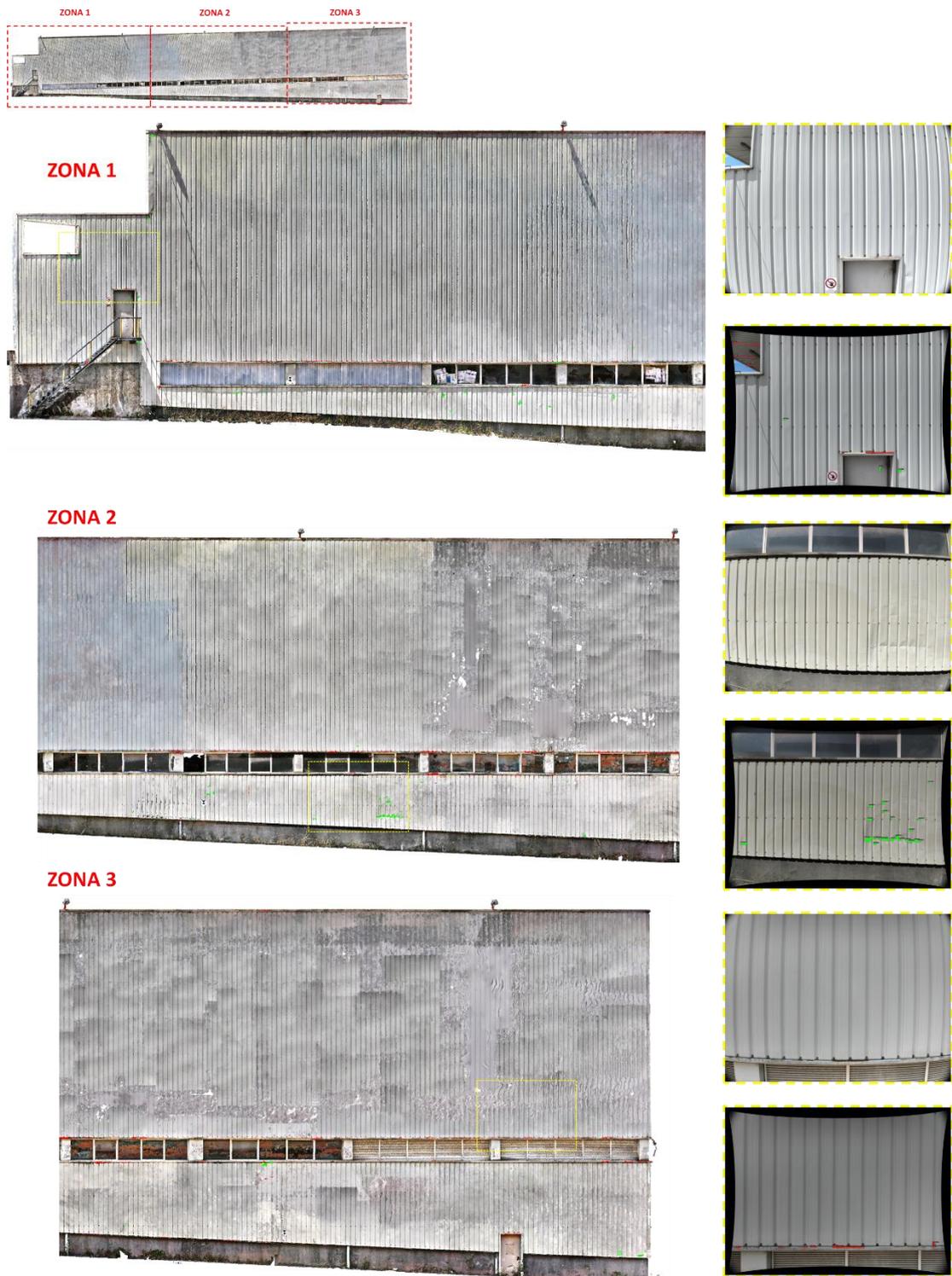


Figura 93 - Modelo da fachada Este

Por sua vez, a Figura 93 divide o modelo da fachada Norte em 3 zonas, com inferências de

corrosão e de choques mecânicos. A fachada possui 0,087 m² de corrosão e 0,045 m² de choques mecânicos.

- **Fachada Oeste**



ZONA 1



ZONA 2



Figura 94 - Modelo da fachada Oeste



Figura 95 - Modelo da fachada Oeste (continuação)

A Figura 94 e a Figura 95 revelam o modelo da fachada Oeste dividido em 4 zonas, com vários pormenores que apresentam corrosões de várias formas, nomeadamente nas esquadrias de portas e janelas. A fachada possui $0,611 \text{ m}^2$ de corrosão e $0,227 \text{ m}^2$ de choques mecânicos.

- Fachada Sul



Figura 96 - Modelo da fachada Sul

A fachada Sul (Figura 96) tem a particularidade de incluir três objetos salientes (uma botija, uma tubagem de circulação de ar e umas escadas de acesso à cobertura) e ainda uma pala. Esta fachada apresenta 0,125 m² de corrosão e 0,029 m² de choques mecânicos.

- Cobertura

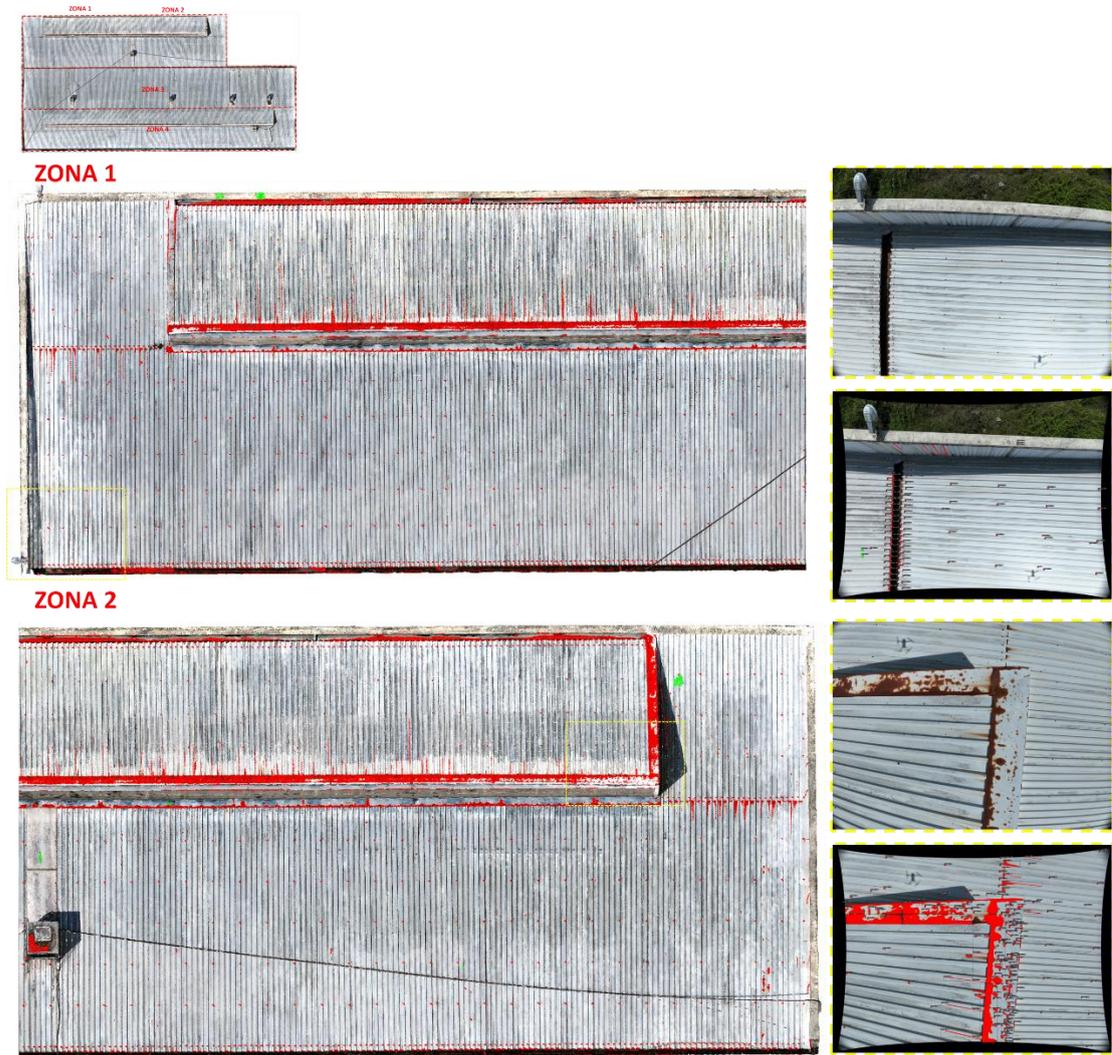


Figura 97 - Modelo da cobertura

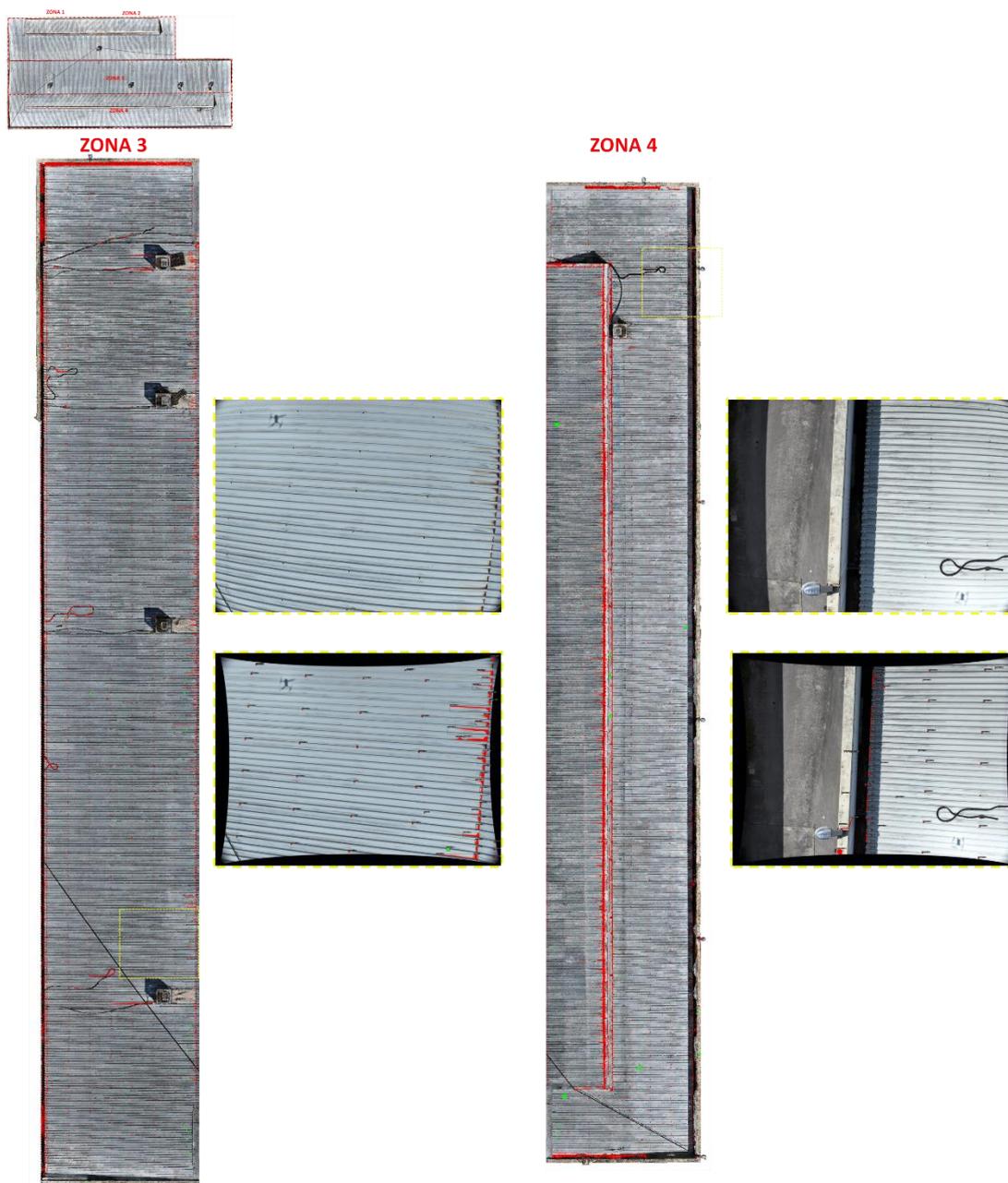


Figura 98 - Modelo da cobertura (continuação)

A cobertura, presente na Figura 97e na Figura 98, foi a zona mais bem sucedida no caso de estudo em termos de identificação exímia dos 3 tipos de anomalias. No entanto, devido ao número elevado de *clusters* e à sua dimensão, o processo de cálculo tornou-se bastante moroso, impossibilitando assim a obtenção dos valores referentes à gestão de dados da cobertura.

5 Conclusões e desenvolvimentos futuros

5.1 Conclusões

Com o intuito de solucionar os entraves associados aos tradicionais métodos de inspeção e de monitorização de edifícios, este trabalho apresenta uma solução capaz de automatizar o processo de inspeção de um ativo industrial, sem comprometer a eficácia desse processo. A metodologia proposta inicia-se na aquisição de dados, com recurso a um TLS e dois UAV's, que proporcionam uma reconstrução tridimensional do ativo através de tecnologias avançadas como LiDAR e fotogrametria. Para além disso, os drones fornecem imagens que auxiliam e constituem o *dataset* de um algoritmo de IA em visão computacional de deteção automática de corrosões, choques mecânicos e acumulações de água em painéis *sandwich*. De forma a tornar o processo de inspeção totalmente automático, foi desenvolvida uma técnica de projeção baseada em *Ray Casting*, com a capacidade de mapear os danos segmentados nas imagens pelo algoritmo concebido no modelo 3D do ativo industrial.

Ferramentas como LiDAR e fotogrametria são fundamentais para inspecionar e monitorizar edifícios. O LiDAR permite a criação de modelos 3D detalhados, com o TLS exigindo boas práticas de calibração e processamento para otimizar os resultados. Por sua vez, a utilização do MLS proporciona detalhes que o *BLK360 G1* não captura, possibilitando uma integração eficiente com *softwares* especializados. Relativamente à fotogrametria, a utilização do DJI M2EA e do DJI M3M permite realizar a reconstrução 3D através da sobreposição de imagens capturadas de diferentes ângulos, enquanto o uso de sistemas RTK e GNSS auxiliam na georreferenciação dos modelos, reduzindo a necessidade de vários GCPs. A metodologia MVS permite a densificação da nuvem de pontos embora apresente limitações no processamento de superfícies lisas ou refletoras. Por fim, a fusão destas tecnologias combina as vantagens de cada uma delas, com o objetivo de gerar modelos mais precisos e completos.

Tendo em conta o estado da arte relativo à segmentação por instâncias, os algoritmos de *deep learning* utilizados foram o *Mask R-CNN*, recorrendo à biblioteca *Detectron2*, e o YOLO, através da sua melhor versão (*YOLOv8*), realizando uma comparação de valores máximos das métricas entre os dois. A base de dados de treino do algoritmo é constituída por 1570 imagens com resolução 8000x6000 pixels, que foram transformadas em 18840 imagens com 2000x2000 pixels, que por sua vez foram divididas em conjuntos de treino, validação e teste, com uma

proporção de 70%, 15% e 15%, respetivamente. As anotações das classes (corrosão, choque mecânico e acumulação de água) nas imagens foram realizadas no *software* VIA para o caso do *Detectron2* e na ferramenta de anotação do *Roboflow Universe* para o *YOLOv8*. De modo a obter os valores máximos de métricas possíveis, foram aplicadas operações de *Data Augmentation* e realizada uma análise de sensibilidade, permitindo identificar os valores otimizados dos hiperparâmetros.

Ambos os algoritmos apresentam um desempenho robusto, mesmo em condições de identificação desfavoráveis, conseguindo detetar e segmentar os três tipos de dano. Observou-se uma melhor aptidão para a identificação de corrosões e de choques mecânicos, razão esta que pode ser explicada pelo facto de existir um número bastante superior de etiquetas destas anomalias quando comparadas com a acumulação de água.

Em termos de métricas, no treino foram atingidas precisões com *Intersection over Union* de 50%, de 68,1% para a deteção e 64,6% para a segmentação, no caso do *Detectron2*. Relativamente ao treino da *YOLOv8*, foram obtidas precisões de 73,3% e 68,3%, para a deteção e segmentação, respetivamente. Apesar das métricas terem sido superiores no caso da *YOLOv8*, verificou-se que as inferências eram relativamente melhores no caso do *Detectron2*. Uma vez que é a qualidade das inferências que permite melhorar as etapas seguintes, optou-se por adotar o melhor modelo do *Detectron2* para realizar o processo de *Ray Casting*.

No que respeita à aplicação em imagens de teste da *Detectron2*, as precisões com IoU de 50%, atingiram os 72,55% para a deteção e 68,54% para a segmentação, enquanto na *YOLOv8*, foram 79,90% e 72,50%, para a deteção e segmentação respetivamente.

No que respeita à análise dos resultados do *Ray Casting*, é importante destacar que esta fase se revela bastante complexa, já que engloba uma série de técnicas e pressupostos fundamentais para a sua aplicação. Apesar de, de uma forma geral, a metodologia demonstrar robustez, verificou-se que, em edifícios de elevada complexidade, pode surgir um desafio significativo relacionado com o custo computacional, especialmente na parte do *clustering* dos danos. Este problema é mais acentuado ainda durante o processamento no *software Itwin*, que se torna bastante exigente ao operar com um elevado volume de dados, como as 7769 imagens inseridas do caso de estudo. Para tentar mitigar esses desafios e otimizar o desempenho do sistema, a solução adotada foi a divisão do edifício em diferentes planos, ou seja, a compartimentação do edifício em 4 fachadas e cobertura. Ao tratar cada um destes elementos

de forma individual, é possível não apenas reduzir a carga computacional, mas também identificar com precisão a localização de cada dano, algo que seria inviável se o edifício fosse avaliado de forma holística. Outro desafio que terá de ser tratado com um maior cuidado será a questão da projeção dos danos, especialmente quando são edifícios com elevada complexidade e ainda quando na sua envolvente estão presentes outros objetos. A presença destes objetos salientes pode fazer com que os raios sejam projetados de forma incorreta, indicando pontos que, na verdade, não correspondem a danos reais. Prevê-se que para resolver este problema, seja necessário, para cada uma das imagens, analisar a distância da câmara aos diversos pontos projetados e avaliar quais as variações dessas distâncias. Caso se identifiquem variações acentuadas, torna-se fundamental ter em atenção essa imagem específica, mantendo apenas os pontos projetados que apresentem um padrão consistente. Para além disso, é importante destacar a limitação do cálculo dos *clusters* em zonas com danos de grandes dimensões, como é o caso da cobertura. Contudo, destaca-se a eficácia da segmentação e do mapeamento para a cobertura. Sendo que é nessa zona que se encontram a maioria dos danos a identificar em edifícios industriais.

De uma forma muito sucinta, concluiu-se que a metodologia aqui proposta é bastante robusta quando aplicável em contexto real. No entanto, é necessário um aperfeiçoamento de certas etapas à medida que vão sendo testados edifícios mais complexos.

5.2 Desenvolvimentos futuros

O trabalho desenvolvido ao longo da presente dissertação caracteriza-se por um elevado grau de complexidade, sendo importante considerar a continuidade e exploração das ideias propostas. Os trabalhos futuros poderão passar por explorar a aplicação de novos procedimentos que aprimorem os resultados obtidos ou a forma como os mesmos são apresentados. Seguidamente, são apresentados alguns aspetos a melhorar na metodologia proposta:

- Aprimoramento da tecnologia utilizada no *Ray Casting* para realização dos *clusters*, correspondentes aos danos, de modo a tornar o processo mais rentável em termos computacionais;
- Aumentar o número de tipos de dano a identificar através do algoritmo de inteligência artificial, como por exemplo fissuras, desgaste dos painéis e acumulação de resíduos;

- Alargar o âmbito de aplicação da metodologia de identificação de danos a outro tipo de soluções de cobertura (de membranas impermeabilizantes, de betão, de fibrocimento ou de telhas), para além dos painéis *sandwich*;
- Implementar um mapa de cores que avalie o nível de risco de degradação, através de uma classificação de número de danos por metro quadrado ou outro critério semelhante;
- Criação de um *dashbord* que permita identificar e posteriormente localizar os diversos danos no modelo 3D. Para além disso, seria interessante criar uma interface gráfica que possibilite a visualização dos modelos fotogramétricos com patologias ao longo dos anos, tirando conclusões acerca da evolução das anomalias.

Referências bibliográficas

AGGARWAL, Charu C. - **Neural Networks and Deep Learning**. Cham : Springer International Publishing, 2018. ISBN 978-3-319-94462-3.

ALJAAFARI, Nura - **Ichthyoplankton Classification Tool using Generative Adversarial Networks and Transfer Learning**

ALLEN PRECISION EQUIPMENT - **USED Leica BLK360 G1** [Em linha], atual. 2023. [Consult. 4 out. 2024]. Disponível em WWW:<URL:https://www.allenprecision.com/shop/13-750569-used-leica-blk360-g1-5972#attr=>.

ALSHAWABKEH, Yahya; BAIK, Ahmad - Integration of photogrammetry and laser scanning for enhancing scan-to-HBIM modeling of Al Ula heritage site. **Heritage Science**. . ISSN 20507445. 11:1 (2023). doi: 10.1186/s40494-023-00997-2.

ARAÚJO, Aline Moura - **Detecção e destaque em vídeo de objetos utilizando YOLO**. Paraíba : Universidade Federal da Paraíba, 2022

ARAÚJO, F. H.; CARNEIRO, A. C.; SILVA, R. R. - Redes neurais convolucionais com tensorflow: Teoria e prática. Em **Livro Anais - Artigos e Minicursos**. Piauí : [s.n.]v. 1. p. 382–406.

BESL, P. J.; MCKAY, Neil D. - A method for registration of 3-D shapes. **IEEE Transactions on Pattern Analysis and Machine Intelligence**. . ISSN 0162-8828. 14:2 (1992) 239–256. doi: 10.1109/34.121791.

BISHOP, Christopher M. - **Pattern Recognition and Machine Learning**. 1. ed. New York : Springer New York, 2006

BŁASZCZAK-BAK, Wioleta *et al.* - Measurement methodology for surface defects inventory of building wall using smartphone with light detection and ranging sensor. **Measurement**. . ISSN 02632241. 219:2023) 113286. doi: 10.1016/j.measurement.2023.113286.

BOCHKOVSKIY, Alexey; WANG, Chien-Yao; LIAO, Hong-Yuan Mark - YOLOv4: Optimal Speed and Accuracy of Object Detection. (2020).

BOUZIANI, M.; CHAABA, H.; ETTARID, M. - EVALUATION of 3D BUILDING MODEL USING TERRESTRIAL LASER SCANNING and DRONE PHOTOGRAMMETRY. Em **International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives**. [S.l.] : International Society for Photogrammetry and Remote Sensing, 7 Out. 2021

BRE, Facundo; GIMENEZ, Juan M.; FACHINOTTI, Víctor D. - Prediction of wind pressure coefficients on building surfaces using artificial neural networks. **Energy and Buildings**. . ISSN 03787788. 158:2018) 1429–1441. doi: 10.1016/j.enbuild.2017.11.045.

BULSARI, A. - Some analytical solutions to the general approximation problem for feedforward neural networks. **Neural Networks**. . ISSN 08936080. 6:7 (1993) 991–996. doi: 10.1016/S0893-6080(09)80008-7.

BURGER, Wilhelm; BURGE, Mark J. - **Digital Image Processing: An Algorithmic Introduction Using Java**.

[S.l.] : Springer, 2012

CABRAL, Rafael De Araújo - **Automated Visual Inspection of Railway Bridges through Computer Vision and Artificial Intelligence**. Porto : Faculdade de Engenharia do Porto, Out. 2023

CAO, Lin *et al.* - A Global Structure and Adaptive Weight Aware ICP Algorithm for Image Registration. **Remote Sensing**. . ISSN 2072-4292. 15:12 (2023) 3185. doi: 10.3390/rs15123185.

CHATZISTAMATIS, S. *et al.* - Fusion of TLS and UAV photogrammetry data for post-earthquake 3D modeling of a cultural heritage Church. Em **International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives**. [S.l.] : International Society for Photogrammetry and Remote Sensing, 6 Mar. 2018

CHEN, Kaiqi *et al.* - BackTracer: Improving ray-casting 3D target acquisition by backtracking the interaction history. **International Journal of Human Computer Studies**. . ISSN 10959300. 176:2023). doi: 10.1016/j.ijhcs.2023.103045.

CHOWDHURY, Ayesha Munira; MOON, Sungwoo - Generating integrated bill of materials using mask R-CNN artificial intelligence model. **Automation in Construction**. . ISSN 09265805. 145:2023). doi: 10.1016/j.autcon.2022.104644.

DELIRY, Sayed Ishaq; AVDAN, Uğur - Accuracy of Unmanned Aerial Systems Photogrammetry and Structure from Motion in Surveying and Mapping: A Review. **Journal of the Indian Society of Remote Sensing**. . ISSN 0255-660X. 49:8 (2021) 1997–2017. doi: 10.1007/s12524-021-01366-x.

DETECTRON2 CONTRIBUTORS - **Benchmarks** [Em linha], atual. 2019. [Consult. 4 out. 2024]. Disponível em WWW:<URL:https://detectron2.readthedocs.io/en/latest/notes/benchmarks.html#benchmarks>.

DÍAZ-GARCÍA, Jesús *et al.* - Progressive ray casting for volumetric models on mobile devices. **Computers and Graphics (Pergamon)**. . ISSN 00978493. 73:2018) 1–16. doi: 10.1016/j.cag.2018.02.007.

DJI - **Mavic 2 Enterprise Advanced - Specs** [Em linha], atual. 2023. [Consult. 4 out. 2024]. Disponível em WWW:<URL:https://enterprise.dji.com/mavic-2-enterprise-advanced/specs>.

DJI - **DJI Mavic 3M - Specs** [Em linha], atual. 2024. [Consult. 4 out. 2024]. Disponível em WWW:<URL:https://ag.dji.com/mavic-3-m/specs>.

DOXYGEN - **Camera Calibration and 3D Reconstruction** [Em linha], atual. 2 jun. 2024. [Consult. 4 out. 2024]. Disponível em WWW:<URL:https://docs.opencv.org/4.x/d9/d0c/group__calib3d.html>.

FABRIS, Massimo; FONTANA GRANOTTO, Pietro; MONEGO, Michele - Expeditious Low-Cost SfM Photogrammetry and a TLS Survey for the Structural Analysis of Illasi Castle (Italy). **Drones**. . ISSN 2504446X. 7:2 (2023). doi: 10.3390/drones7020101.

FACEBOOK AI RESEARCH'S - **detectron2** [Em linha], atual. 15 nov. 2021. [Consult. 4 out. 2024]. Disponível em WWW:<URL:https://github.com/facebookresearch/detectron2>.

FAUSTA FIORILLO; MARCO LIMONGIELLO; CECILIA BOLOGNESI - Integrazione dei dati acquisiti con sistemi image-based e range-based per una rappresentazione 3D efficiente. Em **42th INTERNATIONAL CONFERENCE OF REPRESENTATION DISCIPLINES TEACHERS. CONGRESS OF UNIONE ITALIANA PER IL DISEGNO. PROCEEDINGS 2020. LINGUAGGI, DISTANZE, TECNOLOGIE**. [S.l.] : FrancoAngeli srl, 2021

- FERREIRA, Marcos Vinícius Dos Santos *et al.* - Deep Learning: Uma Introdução às Redes Neurais Convolucionais. Em **Anais Eletrônicos ENUCOMP 2017**. [S.l.] : Fundação Universidade Estadual do Piauí-FUESPI, 2017. p. 793–812.
- GEEKSFORGEES - **Full form of YOLO** [Em linha], atual. 10 jul. 2024. [Consult. 4 out. 2024]. Disponível em WWW:<URL:https://www.geeksforgeeks.org/full-form-of-yolo/>.
- GIRSHICK, Ross *et al.* - Rich feature hierarchies for accurate object detection and semantic segmentation. 2013).
- GIRSHICK, Ross *et al.* - Rich feature hierarchies for accurate object detection and semantic segmentation. 2013).
- GIRSHICK, Ross - Fast R-CNN. 2015).
- GOODENOUGH, Adam A.; BROWN, Scott D. - DIRSIG5: Next-Generation Remote Sensing Data and Image Simulation Framework. **IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing**. . ISSN 1939-1404. 10:11 (2017) 4818–4833. doi: 10.1109/JSTARS.2017.2758964.
- GRUPO ACRE PORTUGAL - **Software de visualização e alinhamento de nuvens Leica Cyclone FIELD 360** [Em linha], atual. 18 set. 2019. [Consult. 4 out. 2024]. Disponível em WWW:<URL:https://grupoacre.com.pt/catalogo-productos/software-de-processamento-de-nuvens-leica-cyclone-field-360/>.
- HAYKIN, Simon S. .. - **Neural networks and learning machines**. [S.l.] : Prentice Hall, 2009. ISBN 9780131471399.
- HE, Kaiming *et al.* - Deep Residual Learning for Image Recognition. Em **2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. [S.l.] : IEEE, Jun. 2016. ISBN 978-1-4673-8851-1
- HE, Kaiming *et al.* - Mask R-CNN. 2017).
- HOSSEINZADEH, Masih *et al.* - Multi-dimensional CFD-Mask R-CNN and CFD-watershed segmentation approach for multiphase non-catalytic gas-solid reactions: A case study for hydrogen reduction of porous iron oxide pellets. **Chemical Engineering Journal**. . ISSN 13858947. 495:2024). doi: 10.1016/j.cej.2024.153474.
- JIANG, Nan *et al.* - A Fusion Method Using Terrestrial Laser Scanning and Unmanned Aerial Vehicle Photogrammetry for Landslide Deformation Monitoring under Complex Terrain Conditions. **IEEE Transactions on Geoscience and Remote Sensing**. . ISSN 15580644. 60:2022). doi: 10.1109/TGRS.2022.3181258.
- KOVANIČ, Ludovít *et al.* - Combined ground-based and UAS SfM-MVS approach for determination of geometric parameters of the large-scale industrial facility – Case study. **Measurement: Journal of the International Measurement Confederation**. . ISSN 02632241. 216:2023). doi: 10.1016/j.measurement.2023.112994.
- KRIZHEVSKY, Alex; SUTSKEVER, Ilya; HINTON, Geoffrey E. - ImageNet classification with deep convolutional neural networks. **Communications of the ACM**. . ISSN 0001-0782. 60:6 (2017) 84–90. doi: 10.1145/3065386.

- LECUN, Y. *et al.* - Handwritten digit recognition with a back-propagation network. Em **Advances in Neural Information Processing Systems**. [S.l.] : Morgan Kaufmann, 1990v. 2. p. 396–404.
- LEICA GEOSYSTEMS AG - **BLK360 TECNOLOGIA** [Em linha], atual. 2024. [Consult. 4 out. 2024]. Disponível em WWW:<URL:https://shop.leica-geosystems.com/br/pt-BR/leica-blk/blk360/blk360-g1-technology?srsItd=AfmBOopMTBRWUjzNclE5txuMmHxLVkPE31BjMsSgj6zO_I6Gchlxe1NK>.
- LIBRALÃO, Giampaolo Luiz *et al.* - **Técnicas de Aprendizado de Máquina para análise de imagens oftalmológica**. São Paulo : [s.n.]
- LIN, Tsung-Yi *et al.* - Feature Pyramid Networks for Object Detection. (2016).
- LIU, Chi Yun; CHOU, Jui Sheng - Bayesian-optimized deep learning model to segment deterioration patterns underneath bridge decks photographed by unmanned aerial vehicle. **Automation in Construction**. . ISSN 09265805. 146:2023). doi: 10.1016/j.autcon.2022.104666.
- LUHMANN, Thomas; CHIZHOVA, Maria; GORKOVCHUK, Denys - Fusion of UAV and terrestrial photogrammetry with laser scanning for 3D reconstruction of historic churches in Georgia. **Drones**. . ISSN 2504446X. 4:3 (2020) 1–18. doi: 10.3390/drones4030053.
- NIGRI, Ilana - **Comparação entre controles look-and-move e servo-visual utilizando transformadas SIFT em manipuladores do tipo eye-in-hand**. Rio de Janeiro : Centro Técnico Científico da Pontifícia Universidade Católica do Rio de Janeiro, 3 Jun. 2009
- NWAOGU, Janet Mayowa *et al.* - Application of drones in the architecture, engineering, and construction (AEC) industry. **Automation in Construction**. . ISSN 09265805. 150:2023) 104827. doi: 10.1016/j.autcon.2023.104827.
- OLIVEIRA, André Estevam Costa - **Detecção e segmentação de edificações utilizando a arquitetura Mask R-CNN na confecção de mapas de uso e cobertura do solo**. Brasília : Universidade de Brasília, 2022
- OLIVEIRA, R. *et al.* - Laser Scanner in Construction Quality Control of Steel Industrial Buildings. **Practice Periodical on Structural Design and Construction**. . ISSN 1084-0680. 28:4 (2023). doi: 10.1061/PPSCFX.SCENG-1389.
- PANDEY, Pallavi - **DBSCAN Clustering** [Em linha], atual. 22 out. 2020. [Consult. 4 out. 2024]. Disponível em WWW:<URL:https://machinelearninggeek.com/dbscan-clustering/>.
- PANTOJA-ROSETO, B. G.; ACHANTA, R.; BEYER, K. - Damage-augmented digital twins towards the automated inspection of buildings. **Automation in Construction**. . ISSN 09265805. 150:2023) 104842. doi: 10.1016/j.autcon.2023.104842.
- PEDOEM, Jonathan; HUANG, Rachel; CHEN, Cuixian - YOLO-LITE: A Real-Time Object Detection Algorithm Optimized for Non-GPU Computers. 1:2018).
- PHUNG, V. H.; RHEE, E. J. - A deep learning approach for classification of cloud image patches on small datasets. **Journal of Information and Communication Convergence Engineering**. 16:2018) 173–178. doi: 10.6109/jicce.2018.16.3.173.
- RAY, H. *et al.* - Ray casting architectures for volume visualization. **IEEE Transactions on Visualization**

- and Computer Graphics**. . ISSN 10772626. 5:3 (1999) 210–223. doi: 10.1109/2945.795213.
- REDMON, Joseph *et al.* - You Only Look Once: Unified, Real-Time Object Detection. 2015) 779–788.
- REN, Shaoqing *et al.* - Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. 3:2015).
- ROBOFLOW UNIVERSE - **YOLOv8: A New State-of-the-Art Computer Vision Model** [Em linha], atual. 2023. [Consult. 4 out. 2024]. Disponível em WWW:<URL:https://yolov8.com/>.
- ROUGERON, Gilles; GARREC, Jérémie Le; ANDRIOT, Claude - Optimal positioning of terrestrial LiDAR scanner stations in complex 3D environments with a multiobjective optimization method based on GPU simulations. **ISPRS Journal of Photogrammetry and Remote Sensing**. . ISSN 09242716. 193:2022) 60–76. doi: 10.1016/j.isprsjprs.2022.08.023.
- ROUGERON, Gilles; GARREC, Jérémie Le; ANDRIOT, Claude - Optimal positioning of terrestrial LiDAR scanner stations in complex 3D environments with a multiobjective optimization method based on GPU simulations. **ISPRS Journal of Photogrammetry and Remote Sensing**. . ISSN 09242716. 193:2022) 60–76. doi: 10.1016/j.isprsjprs.2022.08.023.
- SANTOS, Rafael Lemos Dos - **Identificação de anomalias em edifícios industriais com base em técnicas de inteligência artificial e imagens obtidas com veículos aéreos não tripulados**. Ouro Preto : Escola de Minas da Universidade Federal de Ouro Preto, Nov. 2022
- SHUKLA, M.; ABDELRAHMAN, M. - Artificial neural networks based steady state security analysis of power systems. Em **Thirty-Sixth Southeastern Symposium on System Theory, 2004. Proceedings of the**. Atlanta : IEEE, 16 Mar. 2004. ISBN 0-7803-8281-1
- SILVA, Tayná Cristina Sousa. - **Deteção e Classificação de Derramamento de Óleo na Superfície Oceânica Baseada em Aprendizagem Profunda via Algoritmo Yolo**. Maranhão : Universidade Federal do Maranhão, 2024
- SILVER, David *et al.* - Mastering the game of Go without human knowledge. **Nature**. . ISSN 0028-0836. 550:7676 (2017) 354–359. doi: 10.1038/nature24270.
- SIMONYAN, Karen; ZISSERMAN, Andrew - Very Deep Convolutional Networks for Large-Scale Image Recognition. 2014).
- SON, Seung Woo *et al.* - Integrating UAV and TLS approaches for environmental management: A case study of a waste stockpile area. **Remote Sensing**. . ISSN 20724292. 12:10 (2020). doi: 10.3390/rs12101615.
- SUPERDATASCIENCE TEAM - **Convolutional Neural Networks (CNN): Step 3 - Flattening** [Em linha], atual. 18 ago. 2018. [Consult. 4 out. 2024]. Disponível em WWW:<URL:https://www.superdatascience.com/blogs/convolutional-neural-networks-cnn-step-3-flattening>.
- SZEGEDY, Christian *et al.* - Rethinking the Inception Architecture for Computer Vision. 2015).
- TANG, Pingbo *et al.* - Surveying, Geomatics, and 3D Reconstruction. Em **Infrastructure Computer Vision**. [S.l.] : Elsevier, 2020. p. 13–64.

- TARIQ, Abdullah *et al.* - Heritage preservation using aerial imagery from light weight low cost Unmanned Aerial Vehicle (UAV). Em **2017 International Conference on Communication Technologies (ComTech)**. [S.l.] : IEEE, Abr. 2017. ISBN 978-1-5090-5984-3
- TOWARDS DATA SCIENCE - **Techniques for handling underfitting and overfitting in Machine Learning** [Em linha], atual. 5 jun. 2021. [Consult. 4 out. 2024]. Disponível em WWW:<URL:https://towardsdatascience.com/techniques-for-handling-underfitting-and-overfitting-in-machine-learning-348daa2380b9>.
- UIJLINGS, J. R. R. *et al.* - Selective Search for Object Recognition. **International Journal of Computer Vision**. . ISSN 0920-5691. 104:2 (2013) 154–171. doi: 10.1007/s11263-013-0620-5.
- ULTRALYTICS - **GitHub - ultralytics/ultralytics: NEW - YOLOv8 🚀 in PyTorch > ONNX > OpenVINO > CoreML > TFLite** [Em linha], atual. 11 set. 2022. [Consult. 4 out. 2024]. Disponível em WWW:<URL:https://github.com/ultralytics/ultralytics>.
- VIEIRA, Pablo De Abreu *et al.* - **DETECÇÃO DE DOENÇAS PULMONARES UTILIZANDO APRENDIZADO PROFUNDO EM IMAGENS DE RAIOS-X**
- WANG, Qian; TAN, Yi; MEI, Zhongya - Computational Methods of Acquisition and Processing of 3D Point Cloud Data for Construction Applications. **Archives of Computational Methods in Engineering**. . ISSN 1134-3060. 27:2 (2020) 479–499. doi: 10.1007/s11831-019-09320-4.
- WINGTRA ONE - **Image quality** [Em linha], atual. 19 jun. 2019. [Consult. 4 out. 2024]. Disponível em WWW:<URL:https://wingtra.com/mapping-drone-wingtraone/image-quality/>.
- WINIWARTER, Lukas *et al.* - Virtual laser scanning with HELIOS++: A novel take on ray tracing-based simulation of topographic full-waveform 3D laser scanning. **Remote Sensing of Environment**. . ISSN 00344257. 269:2022) 112772. doi: 10.1016/j.rse.2021.112772.
- WU, Jianxin - Introduction to Convolutional Neural Networks. Em [Em linha] Disponível em WWW:<URL:https://api.semanticscholar.org/CorpusID:36074296>.
- WU, Yu-Chen; FENG, Jun-Wen - Development and Application of Artificial Neural Network. **Wireless Personal Communications**. . ISSN 0929-6212. 102:2 (2018) 1645–1656. doi: 10.1007/s11277-017-5224-x.
- XU, Xiangyang *et al.* - Crack Detection and Comparison Study Based on Faster R-CNN and Mask R-CNN. **Sensors**. . ISSN 1424-8220. 22:3 (2022) 1215. doi: 10.3390/s22031215.
- XU, Yingying *et al.* - Automatic defect detection and segmentation of tunnel surface using modified Mask R-CNN. **Measurement: Journal of the International Measurement Confederation**. . ISSN 02632241. 178:2021). doi: 10.1016/j.measurement.2021.109316.
- YE, X. W.; JIN, T.; YUN, C. B. - A review on deep learning-based structural health monitoring of civil infrastructures. **Smart Structures and Systems**. . ISSN 17381991. 24:5 (2019) 567–585. doi: 10.12989/sss.2019.24.5.567.
- ZEILER, Matthew D.; FERGUS, Rob - Visualizing and Understanding Convolutional Networks. 3:2013).

ZHAO, Qian; ZHOU, Liangchen; LV, Guonian - A 3D modeling method for buildings based on LiDAR point cloud and DLG. **Computers, Environment and Urban Systems**. . ISSN 01989715. 102:2023) 101974. doi: 10.1016/j.compenvurbsys.2023.101974.

ZHONG, B. *et al.* - Mapping computer vision research in construction: Developments, knowledge gaps and implications for research. **Automation in Construction**. . ISSN 09265805. 107:2019). doi: 10.1016/j.autcon.2019.102919.