



ESCOLA NAVAL



ta sainte & bien faire

Duarte Martins de Santo António

Utilização de UAV para Detecção Autónoma de Pontos de Impacto de Morteiro

Dissertação para obtenção do Grau de Mestre em Ciências Militares Navais, na especialidade de Fuzileiros



Escola Naval, 30 de agosto de 2025



ESCOLA NAVAL

talant de bi-faire



Duarte Martins de Santo António

Utilização de UAV para Detecção Autónoma de Pontos de Impacto de Morteiro

Dissertação para obtenção do Grau de Mestre em
Ciências Militares Navais, na especialidade de Fuzileiros

Orientação de: Bruno Duarte Damas
Nuno Alexandre Antunes Martins Pessanha Santos

O Aluno Mestrando,

O Orientador,

Duarte Santo António

Bruno Damas

Escola Naval, 30 de agosto de 2025

"The side that knows more and acts faster wins."

John Boyd

Agradecimentos

Expresso a minha gratidão a todos os que, de forma direta ou indireta, contribuíram para a sua concretização e que influenciaram de forma positiva a minha jornada académica.

Agradeço aos meus orientadores, Engenheiro Bruno Damas e Engenheiro Nuno Pessanha, pela orientação atenta e pela disponibilidade constante e incondicional.

Ao Major José Martins, manifesto a minha sincera gratidão pela cooperação e abertura demonstradas, viabilizando a recolha de imagens no exercício da Academia Militar.

Ao 2TEN FZ Konig Guerreiro, aos elementos do ISR e do pelotão de morteiros da Unidade de Apoio de Combate do Corpo de Fuzileiros, agradeço pela integração, pelo apoio e pelos ensinamentos partilhados durante a Instrução Operacional no Regimento de Artilharia nº5.

Aos meus filhos da escola, aos meus camaradas e imediatos. O vosso apoio, motivação e camaradagem foram fundamentais ao longo de todo este percurso.

Uma palavra especial de agradecimento à minha família, à Catarina e aos meus amigos próximos. São a minha âncora, força e motivação.

Resumo

A qualidade e velocidade da deteção de impactos e da correção de tiro constituem fatores determinantes para a eficácia do apoio de fogo. Esta função é tradicionalmente assegurada pelos Observadores Avançados (OAs), cuja atuação, embora essencial, está frequentemente condicionada por limitações no posicionamento, restrições de visibilidade impostas pelo terreno, dificuldades na localização exata dos pontos de impacto e exposição acrescida ao inimigo. A utilização de *Unmanned Aerial Vehicles* (UAV) surge como uma alternativa promissora, permitindo uma observação contínua e potencialmente automatizada. Esta dissertação propõe o desenvolvimento de um sistema de deteção autónoma de impactos de morteiro a partir de imagens aéreas obtidas por UAV. Para tal, exploram-se duas abordagens distintas: métodos clássicos de visão computacional, baseados em subtração de fundo e análise de movimento, e modelos baseados em *Deep Learning*, nomeadamente a arquitetura YOLOv8. O trabalho compreende a aquisição e anotação de um *dataset* específico, o desenvolvimento dos algoritmos, e a posterior avaliação e comparação dos seus desempenhos.

Palavras-chave: UAV, Observação Avançada, Deteção de Impactos, Visão Computacional

Abstract

The accuracy and speed of impact detection and fire correction are critical factors for effective fire support. This task is traditionally performed by Forward Observers (FOs), whose role, although essential, is often constrained by positioning limitations, reduced visibility due to terrain features, difficulties in precisely locating impact points, and increased exposure to enemy threats. The use of Unmanned Aerial Vehicles (UAVs) emerges as a promising alternative, enabling continuous and potentially automated observation. This dissertation proposes the development of an autonomous mortar impacts detection system based on aerial imagery captured by UAVs. Two distinct approaches are explored: classical computer vision methods, based on background subtraction and motion analysis, and deep learning-based models, namely the YOLOv8 architecture. The work includes the acquisition and annotation of a dedicated dataset, the development of the detection algorithms, and the subsequent evaluation and comparison of their performance.

Keywords: UAV, Forward Observation, Impacts Detection, Computer Vision

Índice

1	Introdução	3
1.1	Contexto Geral	3
1.2	Motivação e Pertinência do Tema para a Marinha Portuguesa	4
1.3	Formulação do Problema	5
1.3.1	Limitações da observação Humana no Terreno	5
1.3.2	Potencial da Utilização de UAVs para Apoio de Fogo	6
1.3.3	Desafios na deteção automática de Impactos	7
1.4	Objetivos	9
1.5	Estrutura da Dissertação	9
2	Enquadramento Teórico	11
2.1	Apoio de Fogo e o Papel dos OAs	11
2.1.1	Apoio de Fogo	11
2.1.2	Observadores Avançados	12
2.2	Visão Computacional	14
2.3	Métodos Clássicos de VC	16
2.3.1	<i>Gaussian Mixed Model</i>	16
2.3.2	Background Subtraction	17
2.3.3	Operações morfológicas	18
2.4	Modelos <i>Deep Learning</i> para Deteção de Objetos	20
2.4.1	Redes neuronais	20
2.5	<i>Multi-Object Tracking</i> (MOT)	25
2.6	Conclusão	27
3	Trabalho Relacionado	29
3.1	Algoritmos de <i>Fire and Smoke Detection</i>	30
3.2	Conclusão	34
4	Metodologia	35
4.1	Aquisição de Dados	36

4.1.1	Características do Equipamento Utilizado	36
4.1.2	Descrição dos Exercícios de Tiro	37
4.1.3	Fatores Operacionais e Ambientais	39
4.1.4	Caracterização do <i>dataset</i>	40
4.2	Visão Computacional Clássica	42
4.2.1	Justificação do método escolhido	42
4.2.2	<i>Pipeline</i>	44
4.3	Deteção com <i>Deep Learning</i>	47
4.3.1	Justificação do Método Escolhido	47
4.3.2	<i>Pipeline</i>	50
4.4	Seguimento de pontos de impacto (ByteTrack)	54
4.4.1	Justificação do Método	54
4.4.2	Associação de deteções	57
4.5	Conclusão	60
5	Resultados	61
5.1	Resultados obtidos com o método baseado em Background Subtraction	61
5.1.1	Testes experimentais e seleção de parâmetros	61
5.2	Resultados com o Algoritmo YOLOv8	65
5.3	Comparação dos Métodos	68
5.4	Discussão dos Resultados	72
5.5	Conclusão	73
6	Conclusões	75
6.1	Considerações Finais	75
6.2	Limitações e Trabalho Futuro	76
	Bibliografia	79

Lista de Figuras

1.1	Perspetiva do OA vs Perspetiva Aérea.	6
2.1	VC, ML, DL e visão humana	15
2.2	Problemas da VC.	16
2.3	Background Subtraction com GMM	18

2.4	Erosão. Retirado de OpenCV Development Team (2025).	19
2.5	Dilatação. Retirado de OpenCV Development Team (2025).	19
2.6	Opening. Retirado de OpenCV Development Team (2025).	20
2.7	Closing. Retirado de OpenCV Development Team (2025).	20
2.8	Neurónio artificial	21
2.9	Rede Neuronal Esquematisada	23
2.10	Classificação de Objetos com CNN	24
2.11	Funcionamento do <i>YOLOv3</i>	25
2.12	MOT com associações de alta e baixa confiança	26
3.1	Distribuição dos artigos encontrados por categoria.	30
3.2	Resultados da segmentação de U2-Net.	31
3.3	Framework of forest wildfire region detection.	32
3.4	Arquitetura do modelo YOLOv8 melhorado.	33
4.1	Esquema geral da metodologia aplicada.	35
4.2	Torre de Observação na Carreira de Tiro A7, CMSM.	38
4.3	Esquema representativo do <i>Pipeline</i> utilizado	44
4.4	Máscara binária realizada com Background Subtraction	45
4.5	Máscara binária após <i>Opening</i>	46
4.6	Máscara binária em sobreposição.	46
4.7	Estrutura do YOLOv8.	48
4.8	Exemplo de imagem e formato de anotação	50
4.9	Transformações aplicadas no <i>Data Augmentation</i>	51
4.10	Evolução das métricas.	53
4.11	Evolução das <i>losses</i> da validação durante o treino do modelo	54
4.12	MOT com associações de alta e baixa confiança	57
4.13	Exemplo adaptação <i>ByteTrack</i>	59
5.1	Resultados experimentais obtidos com abordagem VC clássica	62
5.2	Comparação de um frame de uma deteção com a <i>ground truth</i> .	64
5.3	Curva F1-Confidence obtida na validação do modelo YOLOv8.	66
5.4	Curva Precision-Confidence obtida na validação do modelo YOLOv8.	66
5.5	Curva Precision-Recall obtida na validação do modelo YOLOv8.	67
5.6	Curva Recall-Confidence obtida na validação do modelo YOLOv8.	67
5.7	Falso Positivo	69
5.8	Exemplo de TP. Método baseado em <i>Background Subtraction</i>	70
5.9	resultados com o método baseado em GMM	71
5.10	resultados com o método <i>YOLOv8</i>	71

Lista de Tabelas

3.1	Detecção de incêndios com métodos de deep learning baseados em UAVs	32
4.1	Especificações principais do UAV <i>DJI Air 3</i> . Adaptado de DJI (2023).	37
4.2	Sumário global do <i>dataset</i> utilizado neste trabalho	41
4.3	Detalhe dos vídeos do RA5 (identificação e uso no estudo)	41
4.4	Detalhe dos vídeos do CMSM (identificação e uso no estudo)	42
4.5	Comparação entre variantes do YOLOv8. Dados adaptados da documentação oficial da Ultralytics.	49
4.6	Caracterização do conjunto de dados anotado	52
4.7	Resumo dos parâmetros definidos para o <i>tracking</i>	59
5.1	Parâmetros definidos para o algoritmo MOG2 nos vídeos de teste	63
5.2	Desempenho do algoritmo MOG2 na detecção de impactos (vídeos de teste).	64
5.3	Resultados do YOLOv8 no conjunto de teste	65
5.4	Comparação dos dois métodos no vídeo de teste (7 impactos)	69
5.5	Caracterização dos vídeos de fontes <i>open source</i> utilizados na segunda fase de testes.	70

Lista de Acrónimos

AP	Average Precision.
AP@50	Average Precision at IoU 0.5.
AP@50–95	Average Precision at IoU entre 0.5 e 0.95.
BS	Background Subtraction (<i>Subtração de Fundo</i>).
CAS	Close Air Support (<i>Apoio Aéreo Aproximado</i>).
CMOS	Complementary Metal-Oxide-Semiconductor.
CMSM	Campo Militar de Santa Margarida.
CNN	Convolutional Neural Network (<i>Rede Neural Convolucional</i>).
CPU	Central Processing Unit.
DeepSORT	Deep Simple Online and Realtime Tracking.
DJI	Da-Jiang Innovations.
DL	Deep Learning (<i>Aprendizagem Profunda</i>).
EM	Expectation-Maximization.
FDC	Fire Direction Center.
FN	False Negative (<i>Falso Negativo</i>).
FO	Forward Observer (<i>Observador Avançado</i>).
FP	False Positive (<i>Falso Positivo</i>).
fps	Frames per Second.
FS	Fire Support (<i>Apoio de Fogo</i>).
FSCE	Fire Support Coordination Element.
FSO	Fire Support Officer.
FST	Fire Support Team.

GMM	Gaussian Mixture Model (<i>Modelo de Mistura de Gaussianas</i>).
GPU	Graphics Processing Unit.
HE	High Explosive.
IA	Inteligência Artificial.
IOP	Instrução Operacional.
IoU	Intersection over Union.
ISR	Intelligence, Surveillance and Reconnaissance
JFSE	Joint Fire Support Element.
mAP	Mean Average Precision.
ML	Machine Learning (<i>Aprendizagem Automática</i>).
MOG2	Mixture of Gaussians v2.
MOT	Multi-Object Tracking (<i>Rastreamento de Múltiplos Objetos</i>).
NATO	North Atlantic Treaty Organization.
NFS	Naval Fire Support (<i>Apoio de Fogo Naval</i>).
OA	Observador Avançado.
PCA	Principal Component Analysis (<i>Análise de Componentes Principais</i>).
PO	Posto de Observação.
RA5	Regimento de Artilharia n.º 5.
Re-ID	Re-identification.
SORT	Simple Online and Realtime Tracking.
TP	True Positive (<i>Verdadeiro Positivo</i>).
TP	Training Practice.

TTP	Táticas, Técnicas e Procedimentos.
UAC	Unidade de Apoio de Combate.
UAV	Unmanned Aerial Vehicle (<i>Veículo Aéreo Não Tripulado</i>).
VC	Visão Computacional.
YOLO	You Only Look Once.
YOLOv8	You Only Look Once, versão 8.

Capítulo 1

Introdução

Neste capítulo é apresentado o enquadramento do tema, começando pelo contexto geral da utilização dos morteiros como sistema de apoio de fogo e o papel crítico dos Observadores Avançados (OAs), bem como as suas limitações. Seguidamente, são discutidas a motivação e a pertinência da investigação para a Marinha Portuguesa, a formulação do problema, as dificuldades da observação humana, o potencial dos UAVs e os desafios da deteção automática de impactos. Por fim, são definidos os objetivos do trabalho e descrita a estrutura da dissertação.

1.1 Contexto Geral

O morteiro é um dos principais sistemas de apoio de fogo indireto dos Fuzileiros da Marinha Portuguesa, sendo a sua principal função apoiar a manobra em operações terrestres e anfíbias. Trata-se de um sistema de armas com elevada mobilidade, simplicidade de operação e capacidade de bater alvos a distâncias consideráveis, sem necessidade de linha de visão direta, sendo particularmente útil em terrenos complexos onde a artilharia convencional pode não ter acesso direto. “elevado ritmo de fogo, a trajectória balística, a ligeireza, a rápida entrada e saída de posição, a eficácia do fogo; são características específicas que fazem dos morteiros uma arma com papel único no campo de batalha” (Gil, 1998).

Contudo, os morteiros têm uma precisão intrinsecamente limitada, dada a natureza curva do seu tiro. O disparo é efetuado com base em tabelas balísticas e ajustado manualmente através da alteração da inclinação e direção do cano. A eficácia e rapidez do tiro dependem fortemente da qualidade e velocidade de ação dos OAs (Escola de Fuzileiros, s.d.). Os OAs são militares no terreno responsáveis por identificar alvos, comunicar coordenadas, observar os impactos dos disparos e reportar correções ao fogo. Estes operam, muitas vezes, em ambientes hostis e com visibilidade limitada, enfrentando desafios como a dificuldade em obter boas posições

de observação (sem comprometer a segurança), a falta de visibilidade da zona de impacto devido à geografia ou vegetação e os riscos associados à sua exposição ao inimigo.

Neste cenário, a utilização de *Unmanned Aerial Vehicles* (UAVs) surge como uma oportunidade promissora para reforçar as capacidades de observação avançada, oferecendo uma perspetiva aérea privilegiada, que ultrapassa os obstáculos naturais do terreno, permitindo uma visão clara da zona-alvo e dos impactos e, além disso, eliminam significativamente o risco humano associado à observação direta (Mohsan et al., 2022).

Para maximizar o seu potencial, os UAVs devem integrar algoritmos capazes de detetar os impactos de forma autónoma. No entanto, a deteção automática de impactos continua a ser um problema técnico exigente e pouco estudado em fontes abertas.

1.2 Motivação e Pertinência do Tema para a Marinha Portuguesa

A crescente utilização de veículos não tripulados para fins bélicos tem-se tornado uma tendência e um fator-chave da guerra dos dias de hoje. O novo conceito “Light and Fast” e de operações distribuídas adotado pelos Fuzileiros, beneficiaria de um sistema capaz de auxiliar o controlo dos seus fogos de apoio sem implicar um aumento significativo de efetivo. Com recursos humanos e financeiros limitados, é de vital importância maximizar as capacidades operacionais, garantindo maior precisão e tempos de resposta reduzidos para atender à complexidade do ambiente operacional. O sistema a desenvolver permitirá a deteção de pontos de impacto de morteiro, contribuindo para a correção de tiro de forma rápida e precisa, sem necessidade de aumentar significativamente os efetivos no terreno. Baseado no uso de UAV, alinha-se diretamente com as necessidades operacionais dos Fuzileiros, proporcionando apoio de fogo eficiente aos grupos de combate no terreno e contribuindo para uma “constituição ligeira e flexível, privilegiando o efeito de surpresa, superioridade da informação, conhecimento sobre a área de operações, a mobilidade, a manobra, a velocidade e a letalidade, potenciado” (Marinha Portuguesa, 2022).

1.3 Formulação do Problema

A observação avançada constitui um elemento fundamental na eficácia dos fogos indiretos, como a artilharia ou o apoio de fogo naval (*Naval Fire Support, NFS*), no caso das operações anfíbias. A capacidade de identificar com precisão os impactos dos projéteis e ajustar os parâmetros de tiro em tempo útil é determinante para o sucesso tático. Por sua vez, essa capacidade depende da presença de OAs posicionados em locais que lhes permitam ver claramente o alvo e a zona de impactos. No entanto, em muitos cenários, seja pela hostilidade do ambiente operacional ou pelas características do terreno, torna-se extremamente difícil, ou mesmo impossível, colocar estes observadores em posições eficazes e seguras. Em operações anfíbias, esta dificuldade é ainda mais acentuada durante as fases iniciais, quando não existem forças em terra com capacidade de observação direta.

Para além das limitações físicas impostas pelo terreno, os observadores avançados, por controlarem poder de fogo significativo, são considerados alvos prioritários para o inimigo. Assim, os desafios na deteção dos impactos de morteiro em ambientes complexos tornam evidente a necessidade de recorrer a soluções tecnológicas que possam colmatar as limitações humanas no terreno (Department of the Army, 1991; Guest, 2023; Pinto, 2022).

A utilização de UAVs para a observação avançada pode surgir como uma grande vantagem tática, contudo a deteção automática de impactos, para além de ser um problema pouco estudado e divulgado em documentos de fontes abertas, apresenta inúmeros desafios devido à natureza dos impactos.

Nesta secção serão abordadas as limitações inerentes à observação humana no terreno, o potencial dos UAVs para a observação e deteção de impactos, os desafios da deteção automática de impactos e a escassez de literatura em fontes abertas sobre o tópico.

1.3.1 Limitações da observação Humana no Terreno

Limitações no Posicionamento

As limitações do posicionamento dos OAs representam um primeiro grande desafio na correção de fogos. Muitas vezes, a geografia do terreno impede que o OA encontre um Posto de Observação (PO) de onde veja completamente o alvo e a zona dos impactos. A altimetria e vegetação limitam drasticamente os campos de observação. A doutrina propõe a coordenação de múltiplas equipas de OAs para

mitigar estas limitações, contudo, a realidade do combate nem sempre o permite (Department of the Army, 1991; Exército Português, 2004; NATO Standardization Office, 2015). Além disso, a necessidade de discrição e proteção influencia o posicionamento: um PO com excelente campo de observação é inconcebível se expuser o OA ao fogo inimigo. Os manuais táticos referem explicitamente que POs em cumes ou estruturas proeminentes devem ser evitados, por terem alta probabilidade de serem alvo do inimigo (Department of the Army, 1991; Escola de Fuzileiros, s.d.; Exército Português, 2004). Os OAs têm de equilibrar a obtenção de uma boa capacidade de observação com a sua cobertura e sobrevivência. Esta realidade conduz a posicionamentos sub-ótimos no âmbito da observação.

Perceção e perspetiva

Mesmo quando bem posicionados e equipados com meios óticos avançados, os OAs enfrentam limitações de perceção inerentes à observação ao nível do solo. A falta de uma perspetiva tridimensional compromete a avaliação da profundidade e da distância relativa entre o alvo e o ponto de impacto (Department of the Army, 1991; Exército Português, 2004; NATO, 2020; U.S. Department of the Army, 2024). Um impacto ligeiramente curto ou longo pode parecer alinhado com o alvo, levando a interpretações erradas do erro associado ao tiro, como ilustrado na Figura 1.1.

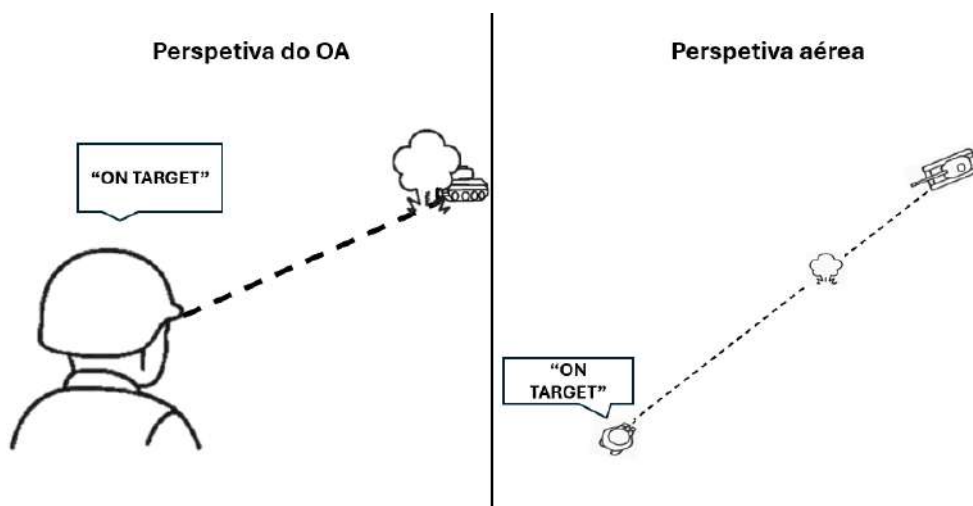


FIGURA 1.1: Perspetiva do OA vs Perspetiva Aérea.

1.3.2 Potencial da Utilização de UAVs para Apoio de Fogo

No contexto das limitações físicas e perceptuais da observação humana no terreno, os UAVs surgem como uma ferramenta potencialmente promissora para apoio aos fogos indiretos. A capacidade dos UAVs de operar acima dos obstáculos

do terreno permite-lhes obter uma perspectiva aérea da zona-alvo, restaurando a componente tridimensional perdida pela observação ao nível do solo.

Além disso, o uso de UAVs reduz o risco humano. A função de observação pode ser desempenhada sem expor militares a posições vulneráveis. Os UAVs podem operar sobre áreas perigosas ou inacessíveis, e se forem neutralizados, não resultam em baixas.

Conflitos recentes, como a guerra na Ucrânia, demonstraram a eficácia dos UAVs na aquisição de alvos e no ajuste de fogos, funcionando como observadores avançados aéreos (Reuters, 2024). A capacidade destes sistemas de fornecer informação em tempo real permite uma regulação mais precisa dos fogos indiretos, potenciando significativamente a eficácia das operações de apoio de fogo. Esta realidade levou à sua crescente incorporação nas doutrinas militares contemporâneas, onde os UAVs são hoje parte integrante do processo de aquisição de alvos, utilizando os seus sensores para melhorar a letalidade e a rapidez na resposta de fogo, (Army, 2021; Centre, 2023; for European Policy Analysis, 2024).

Além disso, o desenvolvimento de capacidades de inteligência artificial e visão computacional tem vindo a reforçar o potencial dos UAVs, permitindo a automatização da deteção e designação de alvos, particularmente útil em teatros de operações complexos e dinâmicos (Army, 2024; Pinto, 2022). Ao contrário do modelo com um operador a olhar para o vídeo em tempo real, a deteção automática assinala os eventos com níveis de confiança, segue-os e georreferencia-os, encurtando o ciclo *sensor-to-shooter* e minimizando o risco de erros humanos (Crifasi, 2025).

1.3.3 Desafios na deteção automática de Impactos

Apesar do elevado potencial da utilização de UAVs para apoio aos fogos indiretos através de uma observação aérea e contínua, a sua plena integração operacional depende da superação dos desafios associados à deteção automática de impactos. As próximas sub-seções abordam mais detalhadamente cada um destes desafios.

Variabilidade dos Impactos

Os impactos de morteiros apresentam uma grande variabilidade visual. As suas formas, cores e dimensões dependem fortemente de fatores como o tipo de munição utilizado, as características do terreno e as condições atmosféricas. Essa variabilidade dificulta a criação de sistemas robustos de deteção, pois o “padrão” de um impacto não é único.

Influência do Ambiente Envolverte

Outra dificuldade significativa na detecção de impactos em vídeo é a distinção entre os sinais da explosão (especialmente o fumo resultante) e elementos do ambiente envolvente que possuem aparência semelhante. O sistema de detecção pode ser confundido por nuvens, sombras, poeira do terreno ou até movimento de vegetação que se assemelham ao aspeto do fumo de um impacto. Este problema é comum aos algoritmos e estudos de detecção de fumo. Wang et al. (2024) observam que este é um dos fatores que mais prejudica a precisão de sistemas de detecção baseados em visão computacional.

Análise Temporal Específica

Ao contrário de objetos estacionários ou eventos prolongados, um impacto de morteiro é um evento transitório e de curta duração. A detonação e os primeiros momentos do impacto ocorrem em frações de segundo, produzindo mudanças muito rápidas no campo visual. É necessário uma análise *frame a frame* para não perder o momento crítico. Isso apresenta dois problemas principais: o requisito computacional elevado e a necessidade de algoritmos adequados para alterações súbitas de luminosidade, aparecimento rápido de fumo/poeira e mudanças bruscas entre *frames*, ocorrendo em questão de milissegundos.

Dispersão Rápida da Nuvem Resultante do Impacto

Logo após a explosão inicial e a formação de uma nuvem de fumo e detritos, essa nuvem tende a dissipar-se rapidamente no ambiente. Seja pela ação da gravidade (que faz os detritos caírem) ou, principalmente, pela ação do vento e turbulência do ar, a pluma de fumo de um impacto não permanece estática: ela expande-se, sofre desvios e enfraquece em questão de segundos. Isso dificulta a identificação do ponto exato de impacto e a confirmação visual do evento pouco tempo depois de ocorrido. Essa rápida dissipação significa que um observador humano ou algoritmo que não detectar o evento nos instantes iniciais terá muita dificuldade em fazê-lo posteriormente, quando se tratar de uma fina névoa dispersa. Outro efeito da dispersão rápida é a incerteza na localização exata do impacto devido ao desvio causado pelo vento.

1.4 Objetivos

O objetivo principal deste trabalho é desenvolver e avaliar soluções para a detecção automática de impactos de artilharia. Para isso, serão estudadas duas abordagens distintas: uma baseada em técnicas clássicas de visão computacional e outra utilizando *deep learning*. Será também objetivo a aquisição de um conjunto de dados com imagens reais de impactos, que permita treinar e avaliar os algoritmos propostos. Ambos os métodos serão comparados em termos de precisão e robustez, de forma a determinar a abordagem mais adequada para cenários reais.

1.5 Estrutura da Dissertação

O Capítulo 1 apresenta o problema, a motivação, os objetivos e a organização do documento. O Capítulo 2 estabelece o enquadramento teórico essencial. O Capítulo 3 revê os trabalhos relacionados. O Capítulo 4 descreve a metodologia da detecção dos impactos, isto é, a aquisição e preparação dos dados, as duas abordagens de detecção e a associação das detecções. O Capítulo 5 apresenta, discute e compara os resultados obtidos. O Capítulo 6 encerra com conclusões, limitações e perspectivas de trabalho futuro.

Capítulo 2

Enquadramento Teórico

Este capítulo apresenta os conceitos teóricos mais importantes enquadrados com o desenvolvimento de uma solução para a deteção automática de impactos de artilharia com recurso a imagens de UAVs. É abordada a importância da observação avançada no apoio de fogo e as abordagens mais relevantes no domínio da visão computacional, incluindo métodos clássicos e *machine learning*, para a deteção de impactos.

2.1 Apoio de Fogo e o Papel dos OAs

O apoio de fogo, *Fire Support (FS)*, engloba o uso de fogos indiretos e outros efeitos de armas combinadas para apoiar a manobra em combate. Na doutrina militar da NATO, o apoio de fogo inclui artilharia, morteiros, *Naval Fire Support (NFS)* e *Close Air Support (CAS)* integrados para alcançar os objetivos do comandante (NATO Standardization Office, 2015). Um apoio de fogo eficaz proporciona profundidade crítica e poder destrutivo no campo de batalha, mas o seu sucesso depende fortemente da qualidade da aquisição de alvos e da precisão do controlo de fogos. É aqui que os OAs desempenham um papel fundamental. A observação avançada, o processo de identificar alvos, pedir e regular os fogos indiretos, é considerada operacionalmente indispensável para um apoio de fogo eficaz.

2.1.1 Apoio de Fogo

No contexto nacional, o apoio de fogo “engloba o emprego coordenado do conjunto dos órgãos de aquisição de objetivos, das armas de tiro direto, indireto (morteiros, Artilharia de Campanha e artilharia naval) e das operações aéreas, em proveito da manobra da força”. Esta capacidade “constitui a parcela essencial da

globalidade dos fogos à disposição do Comandante”, destacando-se pela “flexibilidade de emprego, a prontidão de resposta e a capacidade de fazer sentir os seus efeitos a grande distância” (Exército Português, 2004).

Segundo a North Atlantic Treaty Organization (2023), o apoio de fogo é a aplicação de fogos, coordenada com a manobra das forças, para destruir, neutralizar ou suprimir o inimigo. De acordo com a publicação aliada AArtyP-5 - *NATO Fire Support Doctrine*, um apoio de fogo eficaz requer uma doutrina comum e uma estrutura que permita às forças multinacionais coordenar os fogos de forma integrada. Ao nível tático, a North Atlantic Treaty Organization (NATO) emprega *Joint Fire Support Elements (JFSE)* em cada escalão para planear e coordenar os fogos. Por exemplo, ao nível de uma brigada ou divisão, um *JFSE* planeia e coordena os efeitos da artilharia, aviação e apoio de fogo naval, enquanto ao nível de companhia um *JFSE* mais reduzido (composto frequentemente apenas por uma equipa do *Fire Support Officer (FSO)*, OAs e um controlador aéreo avançado) apoia diretamente o comandante da companhia. A doutrina da NATO sublinha que o apoio de fogo deve ser planeado centralmente, mas com capacidade de execução descentralizada, de forma a responder rapidamente às necessidades do comandante da manobra (NATO Standardization Office, 2015).

O Manual FM 3-09 do Exército dos EUA, *Fire Support and Field Artillery Operations* (2024), descreve de forma semelhante o apoio de fogo como a integração de fogos indiretos e outros meios para atingir alvos em apoio às operações. O apoio de fogo é inerentemente conjunto e multidomínio, combinando fogos *surface-to-surface* (artilharia, rockets, morteiros), fogos aéreos e capacidades de aquisição de alvos para criar os efeitos desejados. A doutrina dos EUA salienta que o comandante pode empregar apoio de fogo orgânico (morteiros do batalhão, unidades de artilharia) bem como meios conjuntos (*CAS* e *NFS*) para materializar os efeitos sobre o inimigo. Tanto na abordagem nacional, da NATO como dos EUA, a aquisição oportuna de alvos e a coordenação são cruciais, razão pela qual o papel dos observadores e dos sensores é repetidamente enfatizado como parte integrante do sistema de *Fire Support*.

2.1.2 Observadores Avançados

Na doutrina nacional, segundo o Estado-Maior do Exército (2012, citado por Pereira (2024, p. 32)), o OA tem como missão localizar, identificar, classificar e priorizar os objetivos, proceder ao pedido de tiro e, por fim, observar e regular o tiro.

Na doutrina da NATO, o OA, ou *Forward Observer* (FO), é definido como o elemento responsável por localizar alvos, ajustar o fogo e reportar os efeitos dos fogos. A publicação AArtyP-01 especifica que “*observers perform target location and adjustment of fire and are responsible for reporting the effects of fire*” (NATO, 2020, p. 4-7). O OA atua integrado numa *Fire Support Team* (FST), em coordenação com o *Fire Support Coordination Element* (FSCE). Essa coordenação é descrita como “*essential to ensure timely and effective fire support*” (NATO, 2020, p. 2-3). As tarefas principais do OA incluem a identificação e localização de alvos, solicitação de fogos, ajustar e regular os fogos e garantir a sua coordenação (NATO Standardization Office, 2015). O OA deve integrar todos os meios de *FS* disponíveis. Isto requer proficiência no plano de apoio de fogo e na utilização dos procedimentos corretos como o *Call for Fire*. O OA controla os fogos, enviando dados do alvo e correções em relação à *Fire Direction Center* (FDC) e observando os efeitos. Uma vez solicitado o fogo sobre um alvo, geralmente com a mensagem padrão do procedimento *NATO call for fire*, (incluindo a localização do alvo, descrição, tipo de efeito pretendido, etc.), a unidade de artilharia ou morteiro realiza os disparos iniciais. O OA observa o impacto (visualmente ou por sensores) para verificar se os disparos atingem o alvo. Caso contrário, transmite correções (*left/right, add/drop*) para conduzir os fogos até ao alvo. Este processo iterativo é conhecido como *fire adjust*. Trata-se de uma competência essencial dos OAs, que devem ajustar o fogo com rapidez e precisão, minimizando o tempo e as munições necessárias para obter os efeitos desejados. Quando as correções necessárias são efetuadas, passam a executar o “*fire for effect*” (disparo em volume total).

Importa salientar que os OAs também desempenham um papel fundamental no controlo de fogos, para além do ajuste. Contribuem para a gestão dos fogos de apoio, prevenindo fratricídio e danos colaterais ao manter consciência situacional das localizações amigas e ao fazer cumprir as *fire support coordination measures* (como *no-fire lines* ou áreas de fogo restrito), conhecendo a posição dessas linhas em relação aos alvos, ((NATO Standardization Office, 2015)).

Importância da Observação Avançada

Historicamente, a introdução de OAs dedicados melhorou significativamente a eficácia da artilharia. Antes da existência do sistema de OA, a artilharia na Primeira Guerra Mundial era frequentemente controlada a partir da retaguarda por ou por planos de tiro pré-definidos, o que a tornava relativamente inflexível e pouco adequada às necessidades da infantaria na linha da frente. Durante o período entre guerras e na Segunda Guerra Mundial, os exércitos perceberam que era essencial

posicionar observadores junto à linha da frente. Na experiência dos EUA durante a Segunda Guerra Mundial, “as equipas de observadores avançados foram o elo vital” que permitiu um novo nível de integração entre a artilharia e a infantaria, (Walker, 2009). Ao transferir a responsabilidade de ajustar os fogos do comandante na retaguarda para o OA na linha da frente, o fogo da artilharia tornou-se mais responsivo, deliberado e coordenado com a situação tática. Este facto foi descrito como uma “mudança monumental” no comando e controlo da artilharia (Walker, 2009). Esta realidade permanece válida até hoje, as operações continuam a depender de observadores para combinar os efeitos dos fogos de longo alcance com a manobra das forças terrestres.

Impacto na Letalidade e no Ritmo das Operações

A observação avançada afeta diretamente a letalidade e o ritmo do apoio de fogo. Um OA competente pode encurtar drasticamente o ciclo sensor-atirador. Na prática, isto significa mais alvos atingidos num menor espaço de tempo e capacidade de explorar oportunidades. Por exemplo, se uma patrulha de infantaria encontra forças inimigas, na presença de OAs, pode imediatamente solicitar fogos indiretos, muitas vezes apanhando o inimigo de surpresa ou fixando-o. Sem um OA, a mesma unidade poderá ter de recuar ou depender apenas de fogo direto, perdendo a oportunidade de empregar o poder de fogo superior da artilharia. Um OA experiente também pode reduzir a quantidade de munições gastas, diminuindo o número de correções necessárias e aumentando assim a eficiência do apoio de fogo. Unidades que integram eficazmente OAs (e outros observadores, como equipas de reconhecimento ou UAVs), conseguem detetar e atingir o inimigo mais rapidamente. Esta agilidade é especialmente crítica em ambientes modernos com adversários móveis e dispersos. A presença de observadores aéreos e terrestres dedicados permite “encurtar a *kill chain*, reduzir a perda de alvos e aumentar a eficácia” das *fire missions* contra alvos de oportunidade (Sims et al., 2021). Em resumo, a observação avançada é um multiplicador de força para a letalidade e rapidez.

2.2 Visão Computacional

A Visão Computacional (VC) é um campo da Inteligência Artificial (IA), que visa dotar os computadores da capacidade de extrair automaticamente informação de imagens digitais (Brownlee, 2019). A um nível mais abstrato, procura inferir características do mundo real a partir de dados visuais, de forma semelhante à visão

humana (Khan et al., 2018; Prince, 2012). A VC é, assim, fundamental na análise automática de grandes volumes de dados visuais.

As abordagens clássicas de VC baseiam-se em técnicas de processamento de imagem que exigem a extração manual de características visuais relevantes. Para isso, utilizam algoritmos específicos para detecção de bordas, contornos ou pontos de interesse, a partir dos quais se extraem descritores que resumem a informação visual da imagem de forma estável, mesmo quando há mudanças na escala, rotação ou iluminação. Estas representações são, posteriormente, utilizadas como entrada para algoritmos de aprendizagem automática convencionais (Dalal & Triggs, 2005; Lowe, 2004; Srinivas & Babu, 2016).

Em contraste, a VC moderna, impulsionada pelos avanços na Inteligência Artificial (IA), baseia-se sobretudo em técnicas de aprendizagem profunda (*deep learning*), recorrendo a *Convolutional Neural Networks* (CNNs). Estes modelos têm a capacidade de aprender automaticamente representações de alto nível a partir de grandes volumes de imagens, eliminando a necessidade de definir manualmente as características, (LeCun et al., 2015). Como resultado, os métodos baseados em Deep Learning (DL) têm demonstrado um desempenho significativamente superior em diversas tarefas de visão por computador, estabelecendo-se como a abordagem dominante no estado da arte (Krizhevsky et al., 2012; Voulodimos et al., 2018).

A Figura 2.1 ilustra esquematicamente a relação entre a VC, a aprendizagem automática (*Machine Learning* – ML), a DL e a visão humana, evidenciando como a VC se posiciona na interseção entre a percepção humana e os métodos de IA (Khan et al., 2018).

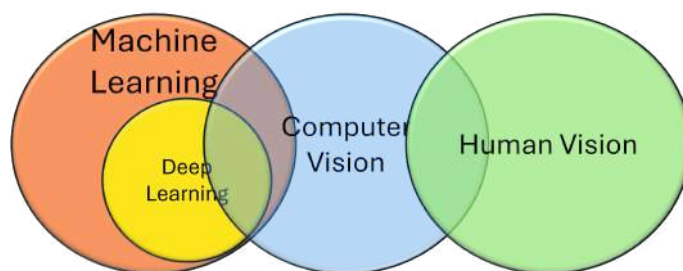


FIGURA 2.1: Relação entre Visão Computacional, *Machine Learning*, *Deep Learning* e a visão humana. Adaptado de (Khan et al., 2018).

Os problemas da visão computacional podem ser, de forma geral, organizados em três categorias principais: classificação, detecção e segmentação, em que classificação corresponde a atribuir uma categoria a uma imagem inteira (ex.: fumo

vs. sem fumo), deteção consiste em localizar e classificar objetos dentro da imagem (ex.: identificar e desenhar bounding boxes sobre um impacto) e segmentação refere-se à divisão da imagem em regiões distintas (ex.: separar a área de fumo do restante cenário), conforme ilustrado na Figura 2.2.

Classificação	Deteção	Segmentação
		
O que está na imagem é fumo?	O que está na imagem e onde está?	Que píxeis pertencem ao objeto?

FIGURA 2.2: Problemas da VC.

2.3 Métodos Clássicos de VC

No contexto desta dissertação, são particularmente relevantes os métodos clássicos de subtração de fundo baseados em modelos de mistura de Gaussianas (*Gaussian Mixed Model*), ajustados pelo algoritmo *Expectation-Maximization* (EM), bem como as operações morfológicas aplicadas à segmentação de impactos em imagens.

2.3.1 *Gaussian Mixed Model*

Os *Gaussian Mixed Model* (GMMs) têm origem na estatística clássica, onde foram desenvolvidos para representar distribuições compostas por múltiplos subgrupos.

Os GMMs pressupõem que os dados são gerados por uma mistura de K distribuições gaussianas, onde cada componente da mistura é definido por uma média (μ), uma matriz de covariância (Σ) e um peso (π) que reflete a contribuição do componente para a distribuição global dos dados (Bishop, 2006). A equação que define a densidade de probabilidade de um GMM é:

$$p(x) = \sum_{k=1}^K \pi_k \cdot \mathcal{N}(x \mid \mu_k, \Sigma_k) \quad (2.1)$$

onde $\mathcal{N}(x \mid \mu_k, \Sigma_k)$ representa uma distribuição gaussiana com média μ_k e matriz de covariância Σ_k , e onde se tem $\sum_{k=1}^K \pi_k = 1$.

Esta fórmula permite modelar dados com clusters de forma elíptica e variâncias distintas, uma vantagem notável comparativamente a métodos como o k-means, que assume clusters esféricos de igual dimensão (McLachlan & Peel, 2000).

Algoritmo de Expectation-Maximization (EM)

Como não se sabe a que componente gaussiano cada dado pertence, a estimação dos parâmetros de um GMM é realizada com o algoritmo Expectation-Maximization (EM). Este método alterna entre duas etapas: na etapa Expectation (E-step), calcula-se a probabilidade de cada ponto pertencer a cada componente e, na etapa Maximization (M-step), esses valores são usados para atualizar os parâmetros do modelo (médias, covariâncias e pesos) de forma a melhor ajustar os dados observados.

O algoritmo EM aumenta a verossimilhança a cada iteração, mas não garante encontrar a melhor solução possível (ótimo global), podendo ficar preso em soluções sub ótimas (ótimos locais). Por isso, a escolha dos valores iniciais e do número de componentes K tem um impacto significativo no desempenho final do modelo (Bishop, 2006).

2.3.2 Background Subtraction

No contexto da subtração de fundo, cada pixel é modelado como uma mistura de distribuições gaussianas, representando as diferentes intensidades observadas ao longo do tempo. As componentes mais frequentes correspondem tipicamente ao fundo, enquanto variações menos recorrentes podem indicar objetos em movimento.

Cada pixel da imagem é modelado como uma mistura de K distribuições gaussianas, em que cada componente representa uma situação (ex: iluminação normal, sombra, reflexo). O modelo atualiza-se, adaptando-se a variações graduais. Para um novo pixel da imagem, verifica-se se ele se ajusta a alguma das distribuições gaussianas modeladas. Se sim, ele é classificado como fundo e os parâmetros do GMM são atualizados. Se não, o pixel é considerado objeto em movimento (*foreground*).

Esta abordagem permite distinguir entre alterações graduais, como variações de iluminação, e mudanças súbitas causadas por eventos de interesse, como impactos ou movimento de fumo. O modelo é atualizado de forma contínua, permitindo adaptar-se a novas condições e melhorar a robustez da segmentação (Zivkovic, 2004).

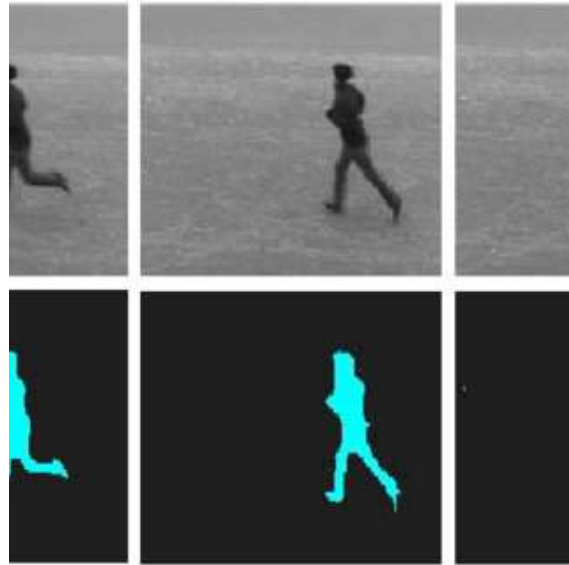


FIGURA 2.3: Background Subtraction com GMM. A linha superior apresenta frames de uma sequência em execução e a segunda linha mostra o resultado da subtração de fundo. Retirado de Bakheet et al. (2012)

2.3.3 Operações morfológicas

As operações morfológicas são transformações aplicadas a imagens binárias que utilizam um *structuring element*, denominado *kernel*, para modificar a forma dos objetos presentes na imagem (OpenCV Development Team, 2025). O *kernel* é uma pequena matriz binária (por exemplo, 3×3) que define a vizinhança analisada em cada pixel. A forma e o tamanho do kernel, bem como o tipo de operação, determinam o efeito da transformação na imagem.

Existem diferentes tipos de *kernel*, consoante a aplicação:

- **Retangular:**

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

- **Cruz:**

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

- **Elíptico** (aproximação binária):

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad \text{ou} \quad \begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$

As principais operações morfológicas são:

- **Erosão**: elimina pixels nas bordas dos objetos, útil para remover ruído.



FIGURA 2.4: Erosão. Retirado de OpenCV Development Team (2025).

- **Dilatação**: adiciona pixels às bordas, preenchendo falhas.



FIGURA 2.5: Dilatação. Retirado de OpenCV Development Team (2025).

- **Opening**: erosão seguida de dilatação, remove ruído mantendo a forma dos objetos.
- **Closing**: dilatação seguida de erosão, preenche pequenas lacunas.
- **Gradiente morfológico**: diferença entre dilatação e erosão, destaca contornos.
- **Top-hat**: diferença entre a imagem original e o seu opening, realça pequenos detalhes claros.

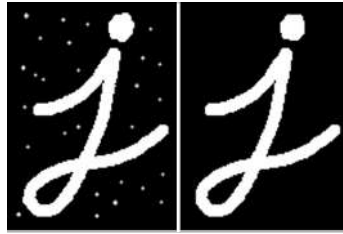


FIGURA 2.6: Opening. Retirado de OpenCV Development Team (2025).



FIGURA 2.7: Closing. Retirado de OpenCV Development Team (2025).

- **Black-hat:** diferença entre closing e a imagem original, destaca falhas escuras no fundo.

2.4 Modelos *Deep Learning* para Detecção de Objetos

A deteção de objetos constitui uma tarefa central no domínio da visão por computador, consistindo na identificação e localização de elementos de interesse em imagens ou sequências de vídeo. Os modelos modernos de deteção, baseados em CNNs, têm demonstrado elevada eficácia nesta tarefa, destacando-se, em particular, a série *You Only Look Once* (YOLO) (Redmon et al., 2016).

2.4.1 Redes neuronais

As redes neuronais são modelos de ML inspirados no cérebro humano, compostos por numerosas unidades interligadas designadas por neurónios artificiais. Estas unidades organizam-se em camadas e constituem uma estrutura capaz de aproximar funções complexas que relacionam as entradas e saídas (por exemplo, relacionar uma imagem com a categoria a que pertence). Em termos simples, uma rede neuronal aprende a executar uma tarefa ajustando internamente um conjunto de parâmetros (pesos) de forma a aproximar a relação desejada entre os dados de entrada e os resultados esperados, (Goodfellow et al., 2016a).

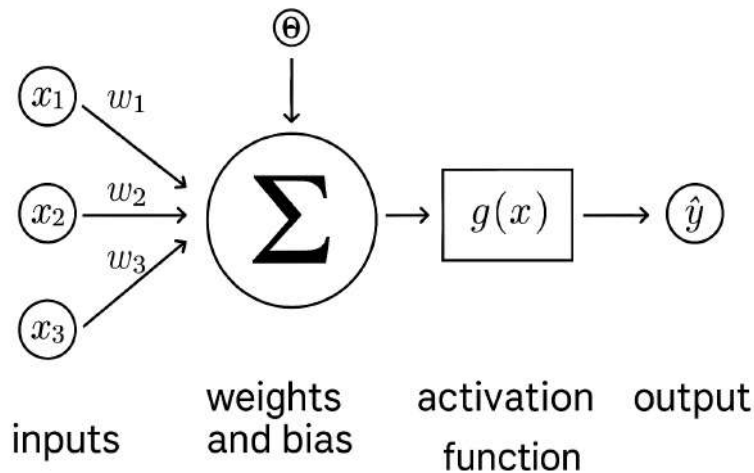


FIGURA 2.8: Neurónio artificial esquematizado. Adaptado de (Bishop, 2006).

Um neurónio artificial, unidade elementar de uma rede neuronal, recebe múltiplas entradas x_1, x_2, \dots, x_n , cada uma associada a um peso correspondente w_1, w_2, \dots, w_n , que determina a importância relativa de cada entrada. Essas entradas ponderadas são somadas, juntamente com um termo de bias Θ , formando a soma ponderada total. Esta soma é então passada por uma função de ativação $g(x)$, que introduz não-linearidade ao sistema e permite que a rede aprenda padrões complexos. O resultado final do neurónio é a saída \hat{y} , que pode representar, por exemplo, uma classificação, uma previsão numérica ou a ativação de outro neurónio em camadas seguintes da rede.

Numa rede *feedforward* típica, a informação flui num único sentido, da camada de entrada até à camada de saída, passando, quando existem, por camadas intermédias denominadas camadas ocultas. Durante o processo de aprendizagem supervisionada, a rede é exposta a múltiplos exemplos (pares entrada-saída esperada), ajustando os seus pesos para que a saída produzida se aproxime do valor correto (por exemplo, a classe real da imagem), (Goodfellow et al., 2016a).

O ajuste é guiado pelos erros obtidos. A rede calcula o erro da predição em cada exemplo utilizando uma função de custo e aplica um algoritmo de optimização (retropropagação do erro) para actualizar iterativamente os pesos no sentido de minimizar esse erro (Goodfellow et al., 2016a). As redes feedforward constituem a base de muitas aplicações modernas de inteligência artificial.

Arquitetura: A arquitetura de uma rede define como os neurónios estão

organizados e interligados. Uma rede feedforward simples inclui uma camada de entrada (que recebe os dados, como os píxeis de uma imagem), uma ou mais camadas ocultas (que extraem padrões) e uma camada de saída (que fornece o resultado final, como a classe prevista), (Goodfellow et al., 2016a). O número de camadas (profundidade), de neurónios por camada (largura) e a forma como se conectam definem a arquitetura.

Pesos sinápticos: As ligações entre neurónios têm valores associados, os pesos, que representam a influência dessas ligações. Durante o treino, os pesos são ajustados automaticamente para reduzir o erro obtido, ((Goodfellow et al., 2016a)).

Função de ativação: Depois de receber os valores das entradas, cada neurónio aplica uma função de ativação para decidir se deve ou não ser ativado. Esta função é não linear, o que permite à rede resolver problemas mais complexos. Sem ela, a rede só conseguiria aprender relações simples. Uma das funções mais usadas é a *Rectified Linear Unit* (ReLU), que devolve zero para valores negativos e mantém os valores positivos. Nas redes convolucionais, estas funções ajudam as camadas a aprender, de forma gradual, padrões visuais cada vez mais complexos: desde contornos até objetos inteiros (Goodfellow et al., 2016a).

Treino (função de custo e otimização): O treino da rede consiste em ajustar os pesos para minimizar uma função de custo, que mede a discrepância entre a saída produzida e a esperada. Utiliza-se, geralmente, o método do gradiente descendente estocástico com retropropagação do erro. O gradiente indica a direção em que os pesos devem ser ajustados para reduzir o erro. Este processo é repetido ao longo de várias épocas (passagem completa por todo o conjunto de treino) até que a rede atinja um desempenho satisfatório (Goodfellow et al., 2016a).

Taxa de aprendizagem: A taxa de aprendizagem (*learning rate*) é um hiperparâmetro que controla o tamanho dos ajustes efetuados nos pesos. Valores muito altos podem provocar instabilidade, enquanto valores baixos podem tornar o treino demasiado lento. Segundo Goodfellow et al. (2016), é um dos parâmetros mais críticos a ajustar numa rede neural. Em prática, são utilizados valores típicos como 0.01 ou 0.001, que podem ser adaptados ao longo do treino.

Redes Neurais Convolucionais

As CNNs são especialmente eficazes em tarefas de visão computacional, como a classificação e a deteção de objetos. Nestes modelos, a imagem é processada

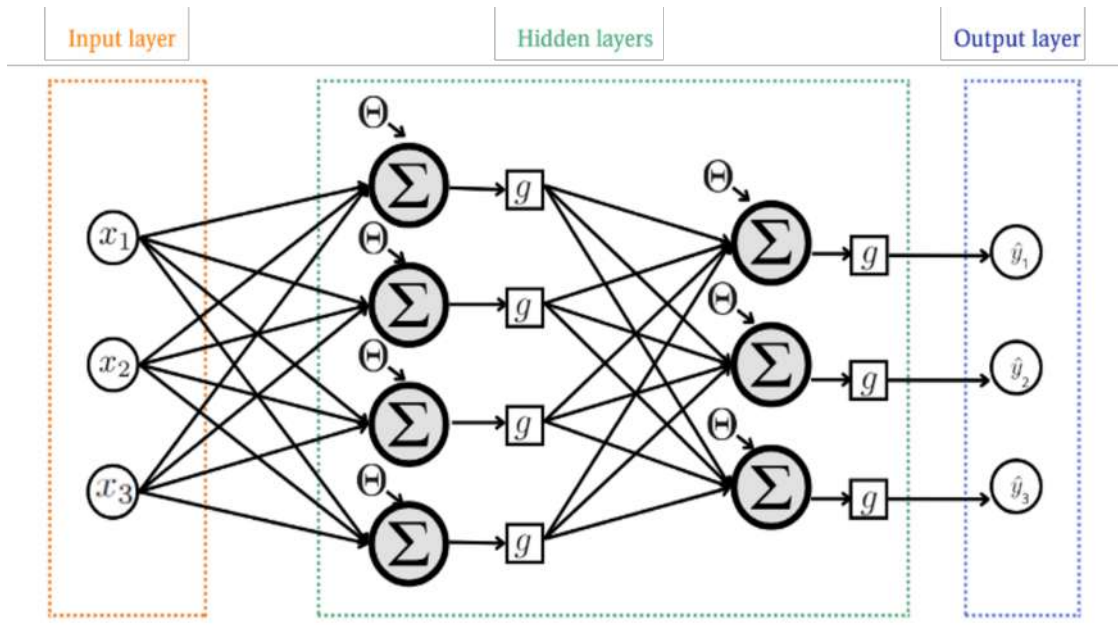


FIGURA 2.9: Rede Neuronal Esquematizada

por camadas convolucionais, que atuam como filtros responsáveis por extrair características visuais. As primeiras camadas detetam padrões simples, como arestas ou texturas, enquanto as camadas mais profundas captam formas mais complexas, como objetos ou estruturas específicas.

Na tarefa de classificação, a rede indica a que categoria pertence uma imagem (por exemplo, “pessoa” ou “leão”). Já na detecção de objetos, o modelo estima simultaneamente as classes presentes e as coordenadas espaciais dos objetos na imagem, representadas por *bounding boxes*.

Modelos como as *Region-based Convolutional Neural Network* (R-CNN) dividem este processo em duas etapas: primeiro propõem regiões candidatas com objetos e, depois, classificam o conteúdo de cada região (Goodfellow et al., 2016a). A R-CNN, apresentada por Ross Girshick et al. em 2014, deu origem a variantes mais rápidas e eficientes, como a Fast R-CNN, Faster R-CNN, e posteriormente a arquiteturas em tempo real, como a família YOLO.

YOLO

O modelo original *YOLO*, proposto por Redmon *et al.* em 2016, reformulou a tarefa de detecção de objetos como um problema de regressão direta, prevendo simultaneamente as *bounding boxes* e as respectivas classes num único passo (Redmon et al., 2016). Esta abordagem permitiu analisar a imagem de forma global e realizar

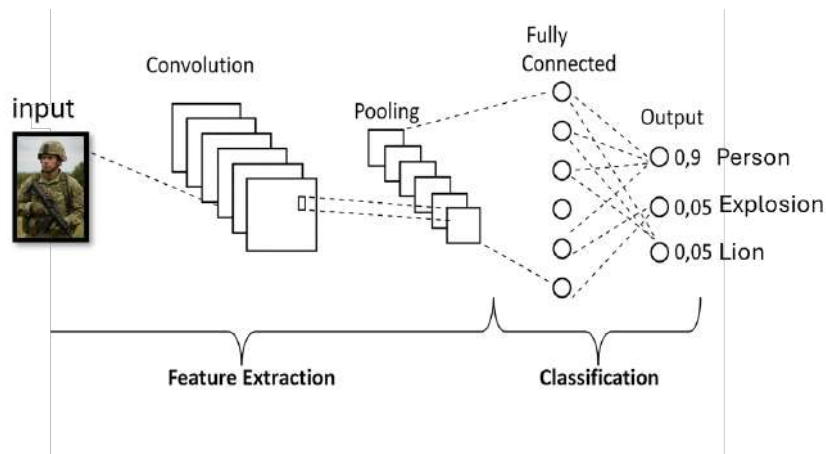


FIGURA 2.10: Classificação de Objetos com CNN. Adaptado de (Goodfellow et al., 2016b).

a detecção de forma eficiente, tornando o método adequado para aplicações em tempo real.

Inicialmente, a imagem era dividida numa grelha regular, e a rede previa, para cada célula, os objetos localizados dentro da sua área. No entanto, essa grelha rígida limitava a detecção de objetos pequenos ou próximos entre si. Para ultrapassar essas limitações, versões posteriores como o *YOLOv3* introduziram detecção em múltiplas escalas (*multi-scale detection*), combinando grelhas de diferentes resoluções, o que melhorou significativamente a robustez e precisão do modelo (Cong et al., 2023).

Desde então, várias versões têm sido desenvolvidas, incorporando melhorias tanto ao nível da arquitetura como das estratégias de treino. Por exemplo, o *YOLOv5*, trouxe vantagens práticas como a implementação em PyTorch e a existência de modelos adaptados a diferentes recursos computacionais (*nano, small, etc.*) (Cong et al., 2023).

Mais recentemente, o *YOLOv8*, lançado pela Ultralytics, introduziu uma abordagem *anchor-free*, dispensando as *bounding boxes* de referência pré-definidas. Esta alteração permite uma maior flexibilidade e simplifica o processo de treino, facilitando a detecção de objetos com dimensões e proporções variadas (Cong et al., 2023; Yaseen, 2024). Esta versão também trouxe melhorias na estrutura da rede e nos mecanismos de treino, tendo sido a escolhida para este trabalho devido à sua boa relação entre precisão e eficiência.

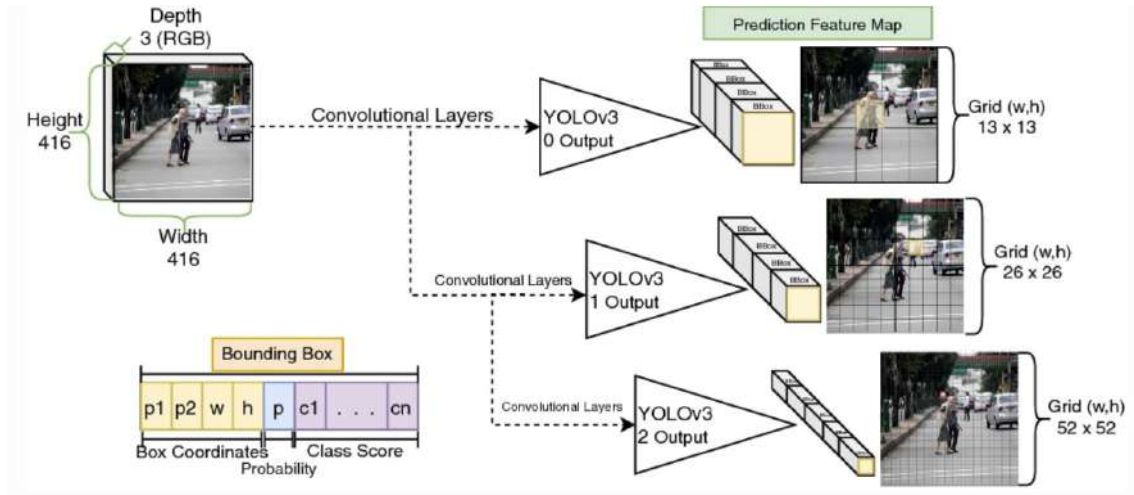


FIGURA 2.11: Funcionamento do *YOLOv3*: a imagem é dividida em grelhas, e cada célula prevê simultaneamente as *bounding boxes* e as classes dos objetos. Retirado de (Sarudalf3, s.d.)

2.5 Multi-Object Tracking (MOT)

Multi-Object Tracking (MOT) é uma tarefa da visão computacional que consiste em detetar e seguir vários objetos ao longo de uma sequência de vídeo, atribuindo uma identidade consistente a cada um (Hassan et al., 2024). O processo baseia-se geralmente em duas fases: a deteção dos objetos em cada *frame*, e a associação dessas deteções entre *frames* consecutivos (Hassan et al., 2024). Os métodos de MOT podem ser divididos em duas categorias principais: *trackers* baseados totalmente em *deep learning* e *trackers* clássicos baseados em associação de deteções (Hassan et al., 2024).

Os *trackers* baseados em *deep learning* utilizam arquiteturas diversas para modelar a correlação temporal entre os *frames*. Estes métodos conseguem aprender a dinâmica do movimento e manter a identidade dos objetos ao longo do tempo sem regras explícitas de associação (Hassan et al., 2024).

Os *trackers* clássicos, conhecidos como *trackers by detection*, baseiam-se na combinação de um detetor externo com algoritmos tradicionais de previsão e associação. Após a deteção dos objetos em cada *frame*, estes métodos utilizam filtros como o de Kalman para prever a próxima posição de cada objeto e resolvem o problema de associação com algoritmos como o Húngaro (Hassan et al., 2024). Exemplos comuns incluem o *Simple Online and Realtime Tracking (SORT)*, *Deep Simple Online and Realtime Tracking (DeepSORT)* e *ByteTrack* (Zhang et al., 2021, 2022). Este tipo de abordagem é mais leve, mais fácil de implementar e amplamente

utilizado em aplicações em tempo real (Hassan et al., 2024).

Trackers clássicos baseados em filtros de Kalman são comuns em sistemas com poucos recursos, pois são simples e rápidos. O *SORT* combina este filtro com o algoritmo Húngaro, sendo eficaz em muitos cenários, embora sofra com trocas de identidade quando há oclusão. Filtros de partículas são usados em contextos com movimento não linear ou ruído não-gaussiano, mas tendem a perder precisão com oclusões, (Hassan et al., 2024).

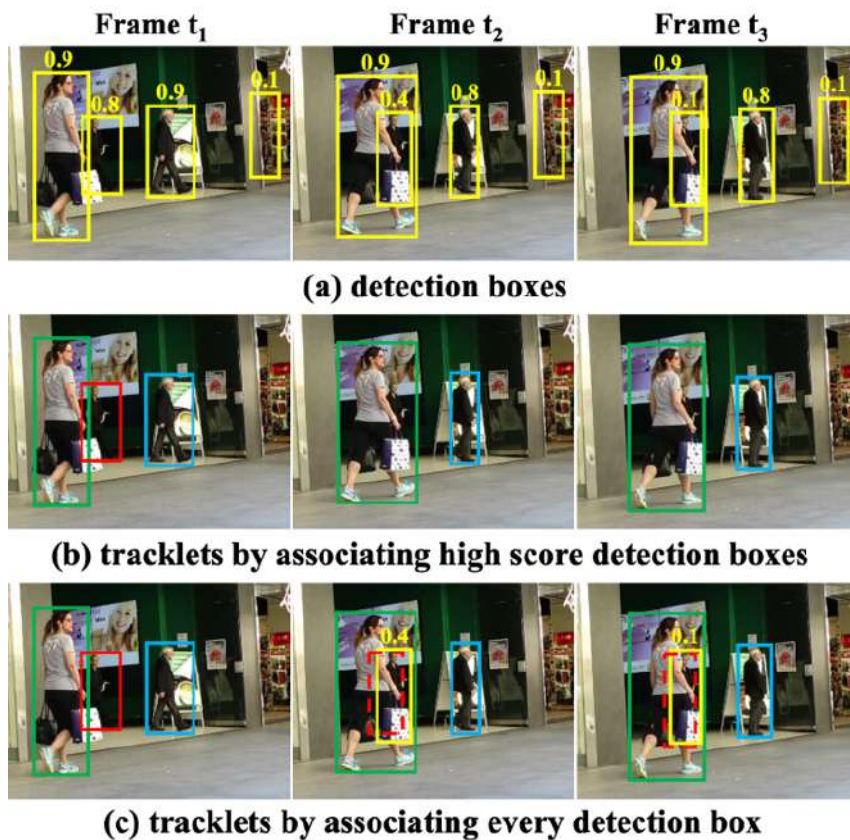


FIGURA 2.12: MOT com associações de alta e baixa confiança

Uma característica que distingue alguns *trackers* modernos é a forma como tratam as detecções de baixa confiança. Métodos tradicionais como o *SORT* descartam essas detecções, o que pode levar à perda de objetos difíceis de detectar de forma consistente. Em contrapartida, algoritmos como o *ByteTrack* integram essas detecções com pontuação abaixo do limiar de confiança principal, tentando associá-las a trajetórias já existentes. Esta estratégia é suportada pela evidência de que “most methods simply throw away low-score detection boxes, which brings non-negligible true object missing and fragmented trajectories” (Zhang et al., 2022). O *ByteTrack*

realiza uma segunda fase de associação entre deteções de baixa confiança e trajetórias não associadas, recuperando objetos ocluídos sem aumentar significativamente os falsos positivos, ((Zhang et al., 2022)).

2.6 Conclusão

Em síntese, este capítulo apresentou os conceitos fundamentais que enquadram a deteção automática de impactos de artilharia utilizando UAVs, recorrendo exclusivamente a técnicas de visão computacional, destacando o papel da observação avançada no apoio de fogo, os métodos clássicos de visão computacional e as abordagens baseadas em ML. Estes elementos fornecem a base teórica necessária para o desenvolvimento e avaliação dos métodos explorados nos capítulos seguintes.

Capítulo 3

Trabalho Relacionado

A detecção automática de impactos de artilharia utilizando UAVs representa uma aplicação emergente e altamente especializada dentro do campo da visão computacional. Contudo, a revisão de bibliografia revelou uma escassez significativa de artigos e trabalhos científicos focados especificamente na detecção automática de impactos de artilharia.

Foi conduzida uma pesquisa bibliográfica de forma abrangente, com o apoio da plataforma SciSpace, que realizou buscas em múltiplas bases de dados e repositórios científicos reconhecidos, incluindo IEEE Xplore, SpringerLink, ScienceDirect, Taylor & Francis, Wiley Online Library, Scopus, Web of Science, arXiv, e Google Scholar. Foram considerados artigos publicados em inglês e português, sem restrições temporais, abrangendo revistas científicas com revisão por pares, atas de conferências e publicações especializadas na área da defesa. Os termos de pesquisa utilizados incluíram “artillery impact detection”, “blast point detection”, “UAV artillery support” e “impact detection UAV”. Apesar do amplo alcance da pesquisa, dos 112 artigos analisados, apenas 27 abordam o uso de UAVs no apoio à artilharia ou em operações de fogos indiretos. Este número reduz-se substancialmente quando se foca a detecção automática de impactos, com apenas 5 artigos a referir o tema e apenas 2 a apresentar métodos concretos de detecção de pontos de impacto, conforme a Figura 3.1.

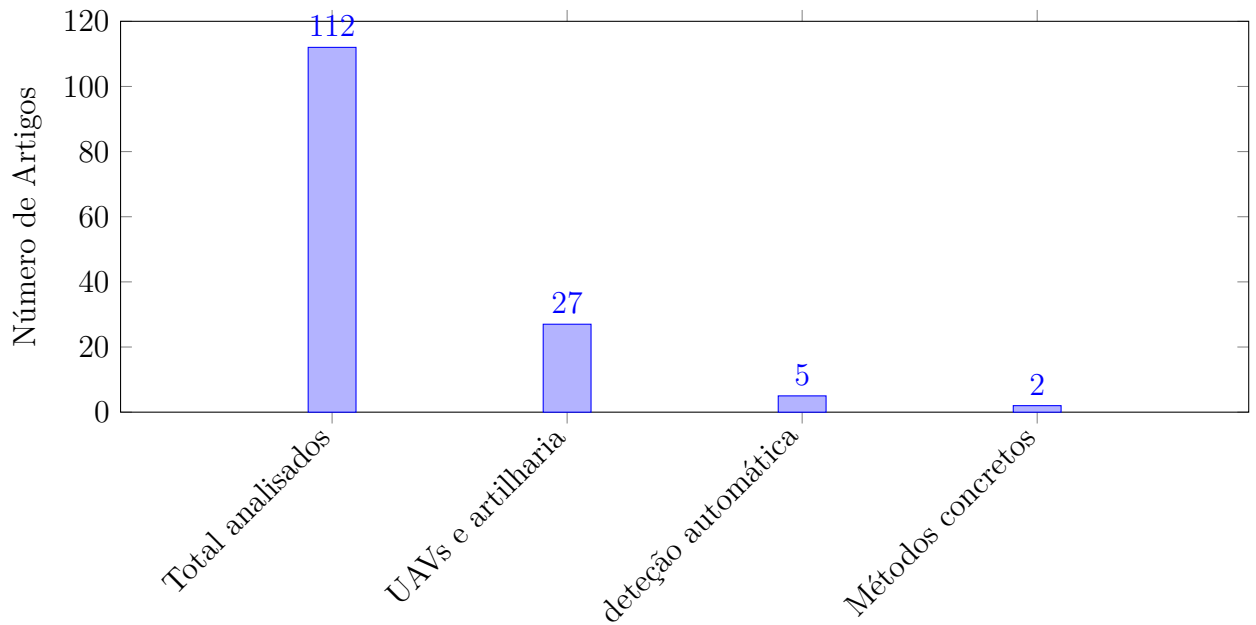


FIGURA 3.1: Distribuição dos artigos encontrados por categoria.

Entre os poucos trabalhos identificados, destacam-se os dois estudos com metodologias concretas para a deteção do ponto de explosão. Contudo, a falta de acesso integral aos textos completos por se encontrarem disponíveis apenas sob diretoria de uma base de dados científica chinesa, limitou a análise dos documentos. O primeiro adota uma abordagem clássica onde modela a sequência de vídeo como uma matriz e aplica *Principal Component Pursuit* (PCA) para separar o fundo do primeiro plano, seguindo-se análise por PCA e critérios de consistência temporal para isolar o ponto de explosão (Qin & Yuan, 2013). O segundo reformula o problema como segmentação por instâncias e propõe uma variante melhorada do *Segmenting Objects by Locations v2* (SOLOv2), orientada para alvos de pequena escala e com desempenho próximo de tempo real, gerando máscaras do *burst point* diretamente nas imagens (Lian et al., 2024). Estes dados evidenciam uma escassez significativa de literatura científica na utilização de UAVs para deteção automática de impactos.

3.1 Algoritmos de *Fire and Smoke Detection*

A deteção de impactos de artilharia com recurso a UAVs constitui um domínio de investigação recente e ainda escassamente documentado na literatura científica. Dado que a bibliografia dedicada à deteção automática de impactos é praticamente inexistente, recorreu-se à análise de trabalhos centrados na deteção de fogo e de fumo em cenários florestais, dada a semelhança visual e dinâmica com o

fenómeno em estudo. A detecção de fogo e fumo tem sido largamente investigada no contexto da prevenção de incêndios florestais, sendo este um campo consolidado no qual é possível encontrar abordagens relevantes, metodologias modernas e soluções implementáveis em UAVs.

Ghali et al. (2021) realizam uma análise comparativa entre as CNNs U-Net, U2-Net e EfficientSeg para a tarefa de segmentação de fogo. A U2-Net é uma variante mais profunda da U-Net, com capacidade aumentada para detecção de contornos detalhados. As funções de perda aplicadas foram *Dice Loss* e *Cross Binary Entropy Dice Loss*. Entre os modelos testados, o U2-Net demonstrou maior precisão na delimitação das regiões afetadas pelo fogo. Na Figura 3.2 é possível observar alguns dos resultados obtidos.

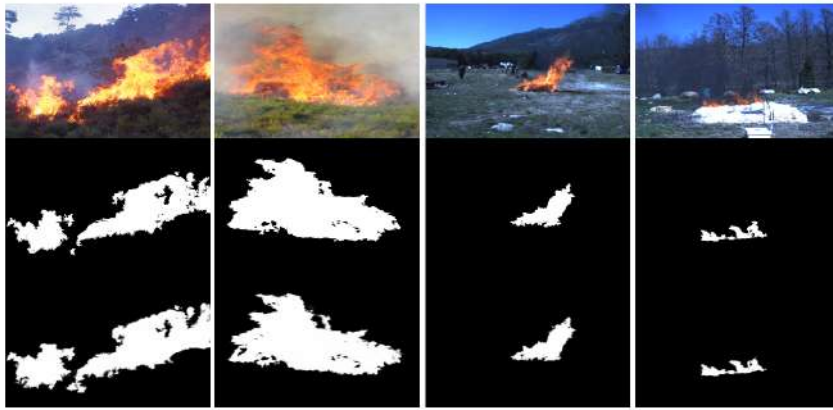


FIGURA 3.2: Resultados da segmentação de U2-Net. Retirado de Ghali et al. (2021).

Ghali et al. (2022) apresentam uma análise comparativa entre métodos de detecção de incêndios florestais baseados em UAVs, recorrendo a redes convolucionais profundas. Os autores destacam a distinção entre detetores de uma etapa, como YOLOv2, YOLOv3 e *Single Shot Multibox Detector* (SSD), que realizam detecção e localização numa única operação, e os detetores de duas etapas, como *Faster Region-based Convolutional Neural Network* (Faster R-CNN), que primeiro propõem regiões de interesse e posteriormente classificam e refinam as predições. Noutros estudos, Jiao et al. (2019) e Jiao et al. (2020) demonstraram a eficácia do YOLOv3 na detecção de fumo e chamas em imagens aéreas, com um F1-score de 81% e desempenho em tempo real. Por outro lado, Alexandrov et al. (2019) compararam YOLOv2, SSD e Faster R-CNN, tendo o YOLOv2 alcançado a melhor precisão (98,3%) na detecção precoce de fumo. Adicionalmente, Tang et al. (2020) propuseram uma abordagem

em imagens 4K com uma estratégia de sub-regiões combinada com YOLOv3, obtendo uma precisão média (*mean Average Precision*, mAP) de 67%. A Tabela 3.1 resume os principais resultados destes estudos, adaptada de Ghali et al. (2022).

TABELA 3.1: Detecção de incêndios com métodos de deep learning baseados em UAVs (adaptado de Ghali et al. (2022)).

Ref.	Método	Tipo	Dataset	Resultados
Jiao et al. (2019)	YOLOv3	Chamas	3.840.000 imagens	F1-score = 81%
Jiao et al. (2020)	YOLOv3	Fumo/Chamas	3.684.000 imagens	F1-score = 81%
Alexandrov et al. (2019)	YOLOv2	Fumo	12.000 imagens	Precisão = 98,3%
	Faster R-CNN	Fumo	12.000 imagens	Precisão = 95,9%
	SSD	Fumo	12.000 imagens	Precisão = 81,1%
Tang et al. (2020)	YOLOv3 + ARSB	Chamas	1.400 imagens 4K	mAP = 67%

Wang et al. (2023) desenvolvem um sistema modular, composto por dois blocos. O primeiro realiza a classificação de imagens com recurso ao modelo Reduce-VGGNet, uma versão compacta da CNN *Visual Geometry Group Network* (VGG-Net). O segundo módulo executa a deteção de regiões afetadas pelo fogo através de uma CNN otimizada que incorpora informação espacial e temporal. Os modelos obtiveram valores de *accuracy* de 91.20% e 97.35%, respetivamente. A estrutura dos módulos encontra-se esquematizada na Figura 3.3.

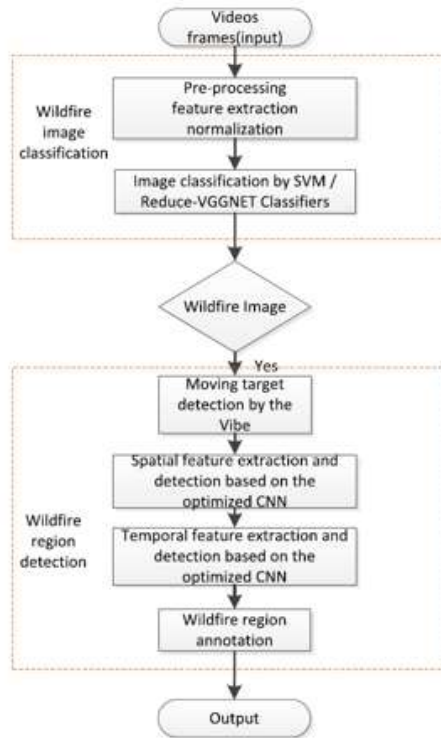


FIGURA 3.3: Framework of forest wildfire region detection. Retirado de Wang et al. (2023).

Saydirasulov et al. (2023) propõem uma versão modificada do YOLOv8, com vista a melhorar a detecção de fumo em imagens de UAV. O modelo introduz o mecanismo *BiFormer Transformer* (BiFormer), um tipo de *transformer* leve que melhora a seleção de regiões relevantes, bem como camadas convolucionais *Ghost Shuffle*, projetadas para reduzir redundâncias na extração de características. Utiliza ainda a função de perda *Wise Intersection over Union* (WIoU) v3, que ajusta dinamicamente os pesos em função da incerteza. O modelo final obteve uma *Average Precision* (AP) de 79.4%, com valores de AP *for small objects* (APS) de 71.3% e AP *for large objects* (APL) de 92.6%.

Yang et al. (2024) introduzem um modelo YOLOv8 melhorado com a inclusão dos módulos *Self-Calibrated Convolution* (SCConv) e *Coordinate Attention* (CoordAtt), além de convoluções deformáveis e da função de perda WIoU v3. O SCConv permite captar padrões espaciais complexos, enquanto o CoordAtt reforça a atenção em zonas de interesse com base em coordenadas espaciais. A mAP atingida foi de 90.02%, representando um ganho de 5.9% relativamente ao modelo base. A arquitetura do modelo encontra-se esquematizada na Figura 3.4.

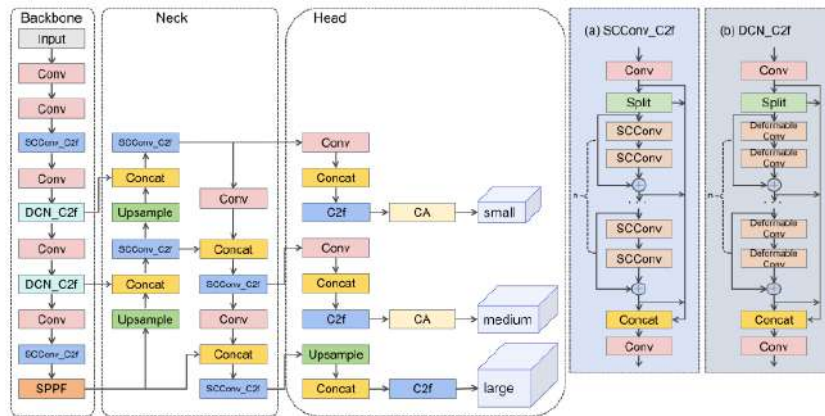


FIGURA 3.4: Arquitetura do modelo YOLOv8 melhorado. Retirado de Saydirasulov et al. (2023).

Casas et al. (2024) comparam diretamente os desempenhos dos modelos YOLOv5 e YOLOv8 nas suas versões small, medium e large, num cenário de detecção de incêndios e fumo. Os testes revelaram que o YOLOv5s apresentou melhor desempenho global em termos de mAP at IoU = 0.50 (mAP@50), *precision* e *recall*. Apesar disso, o YOLOv8 evidenciou vantagens visuais na detecção de fumo escuro e mais denso.

Chen et al. (2024) apresentam uma revisão sistemática sobre os métodos de detecção de incêndios, englobando abordagens clássicas (como *Support Vector Machine* (SVM), análise de cor e movimento) e modernas (como CNNs, YOLO, *Mask Region-based Convolutional Neural Network* (Mask R-CNN) e *transformers*). O artigo destaca como principais desafios: a elevada taxa de falsos positivos em ambientes com neblina, fumo industrial ou luz solar intensa; a escassez de conjuntos de dados públicos e devidamente anotados; e as dificuldades na detecção de fumo difuso ou de pequena dimensão, especialmente em condições de visibilidade reduzida. Embora o YOLOv8 não tenha sido objeto de comparação direta nos estudos analisados, é referido como uma evolução promissora da família YOLO, com melhorias estruturais que poderão beneficiar tarefas de detecção de fumo em tempo real.

3.2 Conclusão

A revisão realizada evidenciou a escassez de literatura científica dedicada à detecção automática de pontos de impacto, o que constitui a principal motivação para a presente investigação. Apesar da abundância de trabalhos no domínio da detecção de fogo e fumo com UAVs, estes continuam a enfrentar problemas significativos, como elevadas taxas de falsos positivos, dificuldade na detecção de fumo difuso e limitações de datasets. Embora existam semelhanças visuais entre fumo de incêndios e fumo de impactos, as especificidades destes últimos, (curta duração, a rápida dispersão e a elevada variabilidade) tornam evidente que se trata de um problema distinto, que justifica investigação própria.

Capítulo 4

Metodologia

Com o objetivo de desenvolver um sistema de detecção automática de impactos de artilharia com recurso a UAVs, foram exploradas duas abordagens distintas: uma baseada em visão computacional clássica, através de técnicas de subtração de fundo com GMM, e outra baseada em redes neuronais profundas, utilizando o algoritmo YOLOv8, com uma arquitetura de detecção *single-stage*. Cada abordagem implicou um processo de preparação e tratamento de dados específico, adaptado às suas exigências metodológicas.

Após a detecção dos impactos por cada uma das abordagens, foi aplicado o algoritmo MOT, com o intuito de associar e seguir as detecções ao longo do tempo, de forma a eliminar redundâncias. Esta metodologia permitiu comparar dois paradigmas distintos de detecção automática, visão computacional clássica e aprendizagem profunda, avaliando os seus desempenhos. A Figura 4.1 apresenta o esquema geral da metodologia aplicada.

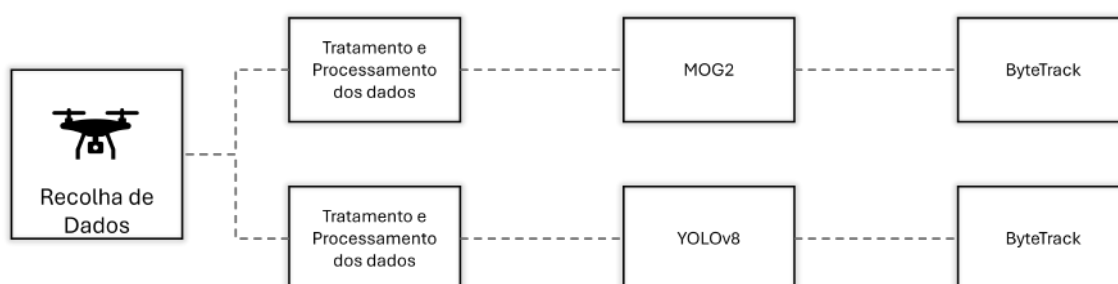


FIGURA 4.1: Esquema geral da metodologia aplicada.

No presente capítulo é descrita a metodologia adotada para a detecção automática de impactos de artilharia com recurso a UAVs. O capítulo encontra-se estruturado da seguinte forma: na Secção 4.1 é descrito o processo de aquisição e caracterização dos dados utilizados; na Secção 4.2 é detalhada a abordagem baseada

em visão computacional clássica, com recurso ao algoritmo MOG2; na Secção 4.3 apresenta-se a abordagem baseada em *deep learning*, utilizando o modelo YOLOv8; na Secção 4.4 é introduzido o algoritmo de *tracking* (ByteTrack) e a sua adaptação aos dois métodos de deteção; por fim, a Secção 4.5 apresenta-se uma conclusão ao capítulo.

4.1 Aquisição de Dados

A fase de recolha de dados reveste-se de particular importância no contexto da deteção automática de impactos de artilharia, visto que a qualidade e robustez dos algoritmos de deteção dependem diretamente da quantidade, qualidade e diversidade das imagens recolhidas. Esta recolha foi realizada em dois locais e momentos distintos, em contexto real de instrução operacional e de formação, com diferentes calibres e tipos de granadas, bem como diferentes condições meteorológicas. Nesta secção será abordado o equipamento utilizado, o ambiente de recolha dos *datasets* e a influência das condições meteorológicas e operacionais no conjunto de dados recolhido.

4.1.1 Características do Equipamento Utilizado

A recolha de dados foi realizada com um UAV *Dji Air 3*, equipado com duas câmaras: uma grande angular de 24 mm e uma teleobjetiva de 70 mm, ambas com sensores *Complementary Metal-Oxide-Semiconductor* (CMOS) de 1/1.3". As imagens foram gravadas em 4K a 60 *fps* e em 1080p a 100 *fps*, com estabilização por *gimbal* mecânico de três eixos, (DJI (2023)).

A autonomia teórica de voo por bateria é de aproximadamente 46 minutos, enquanto a autonomia prática observada em campo rondou os 35 minutos, dependendo das condições de vento, temperatura e movimentos da aeronave. Foram utilizadas três baterias para permitir voos consecutivos. No segundo exercício foi incluído um gerador portátil para carregamento no local, o que aumentou a autonomia operacional disponível.

O controlo foi realizado através da aplicação *DJI Fly* instalada num dispositivo móvel ligado ao comando remoto. As gravações foram armazenadas num cartão microSD com velocidade adequada para gravação contínua em alta definição.

TABELA 4.1: Especificações principais do UAV *DJI Air 3*. Adaptado de DJI (2023).

Parâmetro	Valor
Peso (com bateria)	720 g
Dimensões (dobrado, sem hélices)	207 × 100.5 × 91.1 mm
Autonomia teórica por bateria	46 minutos
Autonomia prática estimada	~35 minutos
Sistema de transmissão	OcuSync 4.0, até 20 km
Sensores de imagem	1/1.3" CMOS (duplos)
Lentes	24 mm f/1.7 e 70 mm f/2.8
Resolução de vídeo	4K 60 <i>fps</i> , 1080p 100 <i>fps</i>
Estabilização	<i>Gimbal</i> mecânico de 3 eixos
Armazenamento	Cartão microSD (até 512 GB)

4.1.2 Descrição dos Exercícios de Tiro

Foram realizados dois momentos distintos de recolha de dados, em contextos de instrução operacional com granadas de morteiros de diferentes calibres e condições ambientais distintas. Ambos os exercícios visaram captar impactos reais de artilharia com UAV, permitindo a construção de um conjunto de dados representativo e variado. Segue-se a descrição de cada exercício.

Regimento de Artilharia N^o 5

O primeiro momento de aquisição de dados decorreu a 17 de Dezembro de 2024, no Regimento de Artilharia n^o 5 do Exército, em Vendas Novas, no âmbito de uma Instrução Operacional (IOP) conduzida pelos Fuzileiros, mais concretamente pela Unidade de Apoio de Combate (UAC). Esta IOP enquadrou-se no aprontamento de força para missão na Lituânia, tendo como objetivos principais o treino de Táticas, Técnicas e Procedimentos (TTP) do tiro de morteiros e qualificação dos OAs.

Durante o exercício foi efetuado fogo indireto com morteiro de 81 mm, utilizando granadas Training Practice (TP). Foram efetivamente disparadas 20 granadas, tendo sido captados 10 impactos. Três dos disparos foram considerados como perdidos, não tendo sido possível identificar os respetivos pontos de impacto. Os restantes sete impactos não foram registados devido à limitação imposta pela autonomia reduzida do UAV e ao escasso número de baterias suplementares. Este fator veio a ser colmatado no segundo momento de recolha, com a introdução de um gerador portátil para carregamento no local de observação.

As condições meteorológicas mantiveram-se relativamente estáveis ao longo do dia, com céu parcialmente nublado, temperaturas entre os 5 °C e os 13 °C, e vento moderado de Noroeste. O vento influenciou ligeiramente a estabilidade do UAV e a dinâmica do fumo, que foi deslocado na direção Sudeste, com conseqüente dispersão acelerada. O terreno na zona de impacto era predominantemente arenoso e de cor clara. A utilização de granadas de exercício, cuja carga gera fumo claro e de baixa densidade, resultou em impactos com reduzido contraste relativamente ao plano de fundo, dificultando a sua deteção em perspetiva aérea.

Campo Militar de Santa Margarida

O segundo momento de recolha ocorreu a 17 de Fevereiro de 2025, na Carreira de Tiro A7 do Campo Militar de Santa Margarida, durante um exercício de instrução básica ministrado a cadetes da Academia Militar. A atividade decorreu ao longo de todo o dia, dividida em dois períodos distintos.



FIGURA 4.2: Torre de Observação na Carreira de Tiro A7, CMSM.

Durante a manhã, foi realizado tiro indireto com morteiros de 81 mm, utilizando granadas de High Explosive (HE). Na parte da tarde, procedeu-se ao tiro com dois morteiros pesados: um de 102 mm e outro de 120 mm, também com granadas HE. No final do exercício foi ainda realizada uma salva com granadas de fumo. A observação e correção de tiro estiveram a cargo dos cadetes, sob supervisão de OAs experientes.

No total, foram registados 51 impactos: 11 de morteiro de 81 mm e 40 de morteiros de calibres 102 mm e 120 mm. As condições meteorológicas foram favoráveis à captação: céu maioritariamente nublado, temperaturas entre os 11 °C e os 17 °C e vento fraco. Estas condições proporcionaram boa estabilização do UAV e uma excelente definição visual dos impactos.

As granadas HE produziram explosões de maior magnitude e fumo escuro, com elevado contraste face ao solo, o que facilitou a deteção. Os cones de fumo apresentavam-se concentrados e com orientação vertical, características potenciadas pela ausência de vento significativo. Adicionalmente, a variação da cobertura de nuvens permitiu captar imagens com diferentes condições de iluminação e exposição, contribuindo para a diversidade do dataset recolhido.

4.1.3 Fatores Operacionais e Ambientais

A eficácia da captação dos impactos foi condicionada por múltiplos fatores operacionais e ambientais. No primeiro momento, a limitação da autonomia do UAV, associada à ausência de meios para carregamento em campo, impediu a recolha contínua e completa dos impactos. Este fator logístico foi devidamente identificado e corrigido, permitindo no segundo exercício uma cobertura substancialmente superior.

Do ponto de vista ambiental, o vento foi um elemento com impacto considerável, influenciando tanto a estabilidade da plataforma aérea como a dispersão dos fumos resultantes dos impactos. A natureza do terreno, arenoso e claro, associada à utilização de granadas TP, dificultou a perceção visual dos impactos no primeiro exercício. Em contraste, o uso de munições HE e a presença de fumo escuro e explosões mais evidentes permitiram, no segundo exercício, captar dados de maior qualidade visual.

Escolha da Altitude de Voo

Em contexto tático, é essencial que o UAV opere de forma a minimizar a sua deteção e a altitude de voo do UAV é um fator crítico para a sua furtividade. Altitudes mais elevadas dificultam a identificação a olho nu da aeronave não tripulada e a distância ao solo contribui para a atenuação do ruído emitido pelos motores ou rotores.

Mas a altitude de voo do UAV é também um fator determinante na detecção visual dos impactos de artilharia. Este parâmetro influencia a qualidade da detecção e a dimensão da área vigiada.

A uma menor altitude, o UAV capta imagens com maior resolução do solo, permitindo uma identificação mais precisa e imediata dos sinais característicos de um impacto, tais como o fumo, a dispersão de detritos ou alterações súbitas na textura do terreno. Esta proximidade ao plano de impacto potencia a eficácia dos algoritmos de detecção, ao facilitar a discriminação entre o impacto e elementos naturais ou ruído visual. Contudo, este modo apresenta uma desvantagem significativa: a área de vigilância é substancialmente reduzida, aumentando a probabilidade de um impacto ocorrer fora do campo de visão da câmara, sobretudo em contexto de tiro de baixa precisão característica dos equipamentos de tiro curvo.

Uma maior altitude de voo proporciona uma cobertura significativamente superior, reduzindo a probabilidade de perda dos impactos. Ainda que tal implique uma menor perceção das características dos impactos, tornando mais desafiante a sua detecção, esta abordagem permite uma maior capacidade de vigilância.

Tendo estes fatores em consideração, bem como as características e segurança do equipamento, a altitude de voo do UAV foi adaptada garantindo o compromisso entre a qualidade da perceção e minimização do risco de perda do impacto, optando por menores altitudes durante a utilização de granadas de menor calibre e do tipo TP, (entre os 80 e os 90 metros de altitude relativa) e maiores altitudes na utilização de granadas de maior calibre do tipo HE, (entre os 100 e os 115 metros de altitude relativa).

4.1.4 Caracterização do *dataset*

O *dataset* utilizado resulta de dois momentos distintos de recolha (Regimento de Artilharia nº 5 e Campo Militar de Santa Margarida) e integra ainda imagens adicionais de fontes abertas para aumentar a diversidade no treino do modelo de *deep learning*. A Tabela 4.2 sintetiza os totais por tipo de dado e as informações relevantes para cada etapa do *pipeline*.

TABELA 4.2: Sumário global do *dataset* utilizado neste trabalho

Item	Valor
Vídeos captados (total)	9
Resoluções originais	3840×2160 (4K) @ 60 (fps)
Pré-processamento para BS (MOG2)	1280×720 @ 30 fps
Imagens extraídas para YOLO (exercícios)	179
Imagens adicionais (fontes abertas)	24
Total de imagens para YOLO	203
Impactos captados no Regimento de Artilharia nº 5 (RA5)	10
Impactos captados no Campo Militar de Santa Margarida (CMSM)	51
Impactos totais captados	61

Regimento de Artilharia nº 5 (17-12-2024)

Sessão de instrução com morteiro de 81 mm (granada TP), em Vendas Novas. Terreno claro e fumo de baixo contraste. Autonomia limitada condicionou a captação contínua. A Tabela 4.3 apresenta informações relevantes de cada vídeo, após pré-processamento.

TABELA 4.3: Detalhe dos vídeos do RA5 (identificação e uso no estudo)

ID	Res./FPS	Duração	nº impactos	Calibre/munição	Uso
RA5-01	1280×720 @ 30 fps	00:00:35	1	81 mm / TP	Treino
RA5-02	1280×720 @ 30 fps	00:00:33	1	81 mm / TP	Treino
RA5-03	1280×720 @ 30 fps	00:00:29	1	81 mm / TP	Treino
RA5-04	1280×720 @ 30 fps	00:00:37	1	81 mm / TP	Teste
RA5-05	1280×720 @ 30 fps	00:00:21	1	81 mm / TP	Treino

Campo Militar de Santa Margarida (17-02-2025)

Exercício com 81 mm, 102 mm e 120 mm (granadas HE, e salva de fumo), câmara estável durante a manhã e maior variedade de condições ao longo do dia. A Tabela 4.4 apresenta informações relevantes de cada vídeo, após pré-processamento.

TABELA 4.4: Detalhe dos vídeos do CMSM (identificação e uso no estudo)

ID	Res./FPS	Duração	nº impactos	Calibre/munição	Uso
CMSM-01	1280×720 @ 30 fps	00:01:22	5	81 mm / HE	Treino
CMSM-02	1280×720 @ 30 fps	00:01:41	7	81 mm / HE	Teste
CMSM-03	1280×720 @ 30 fps	00:03:43	22	102 e 120 mm / HE	Treino
CMSM-04	1280×720 @ 30 fps	00:03:14	17	102 e 120 mm / HE	Treino

4.2 Visão Computacional Clássica

A primeira abordagem explorada para a deteção de impactos consistiu na utilização de técnicas clássicas de visão computacional, baseadas em *Background Subtraction*. Concretamente, recorreu-se ao algoritmo *Mixture of Gaussians v2* (MOG2), disponibilizado pela biblioteca OpenCV, o qual modela cada píxel do fundo da imagem como uma mistura adaptativa de distribuições Gaussianas. Esta abordagem mostrou-se particularmente adequada à deteção de alterações repentinas, como é característico dos impactos de artilharia.

4.2.1 Justificação do método escolhido

A deteção de impactos de morteiros a partir de UAVs exige identificar alterações súbitas e localizadas no cenário, como nuvens de fumo, poeira e projeções de material, fenómenos que contrastam de forma breve mas distinta em relação ao fundo. Neste contexto, os métodos de *background subtraction* revelam-se adequados, uma vez que permitem isolar regiões em movimento ou alterações repentinas face ao fundo modelado, sem necessidade de treino prévio com grandes quantidades de dados.

Para o método de *Background subtraction*, foi utilizado o algoritmo MOG2, uma versão melhorada do algoritmo original *Gaussian Mixture-based Background/-Foreground Segmentation Algorithm* proposto por Zivkovic (2004)). A sua base assenta na modelação estatística de cada píxel da imagem como uma mistura de distribuições gaussianas, permitindo identificar quais os valores que pertencem ao *background* e quais os que representam alterações (potencialmente objetos em movimento).

A versão MOG2, introduz um mecanismo adaptativo para o número de componentes gaussianos utilizados na modelação de cada pixel. Enquanto o algoritmo original (MOG1) utilizava um número fixo de distribuições gaussianas por pixel (normalmente entre 3 a 5), o MOG2 ajusta esse número com base na complexidade estatística do histórico recente dos valores observados em cada ponto da imagem.

Este mecanismo permite que, em regiões do vídeo com maior variabilidade (por exemplo, zonas com vegetação a oscilar ao vento ou superfícies com reflexos de luz), o modelo utilize mais gaussianas para representar adequadamente o comportamento do fundo. Por outro lado, em áreas mais estáveis, o número de gaussianas é menor, tornando o processamento mais leve.

Este ajuste automático torna o MOG2 mais robusto a variações ambientais, como flutuações de iluminação, sombras e pequenas movimentações no fundo. Além disso, incorpora a funcionalidade de deteção de sombras, que pode ser útil para distinguir entre alterações reais e artefactos de iluminação.

A deteção de impactos a partir de UAVs recorre à observação de sinais visuais discretos e de curta duração como nuvens de fumo, poeiras e projeções de material. Estes elementos tendem a contrastar de forma subtil com o terreno envolvente mas surgem de forma súbita, o que corresponde bem ao tipo de alterações que o MOG2 consegue identificar.

Outros métodos de subtração de fundo, como o KNN (*K-Nearest Neighbors*) ou técnicas baseadas em mediana temporal, foram considerados. No entanto, o KNN apresenta menor sensibilidade a alterações mais subtis e os métodos baseados em mediana assumem um cenário demasiado estático e são menos adaptativos a alterações ambientais, como variações de luz ou movimento do fundo. O MOG2 oferece um compromisso equilibrado entre adaptabilidade, desempenho e custo computacional, sendo amplamente utilizado em aplicações de vigilância e deteção de eventos em vídeo.

Embora o MOG2 consiga segmentar eficazmente regiões em movimento, as máscaras obtidas incluem frequentemente ruído visual. Para mitigar este problema, foram aplicadas operações morfológicas de *Opening* (erosão seguida de dilatação), já abordadas no enquadramento teórico. Esta etapa adicional permitiu eliminar regiões isoladas não representativas e suavizar os contornos das áreas detetadas, resultando em máscaras mais limpas e adequadas ao processo de deteção.

4.2.2 Pipeline

O *pipeline* de detecção inicia-se com o vídeo original, que é sujeito a pré-processamento (redução de resolução e *framerate*), seguido da subtração de fundo pelo algoritmo MOG2. As máscaras resultantes são posteriormente refinadas com operações morfológicas de *opening*, sendo finalmente sobrepostas ao vídeo original para evidenciar as regiões associadas aos impactos.

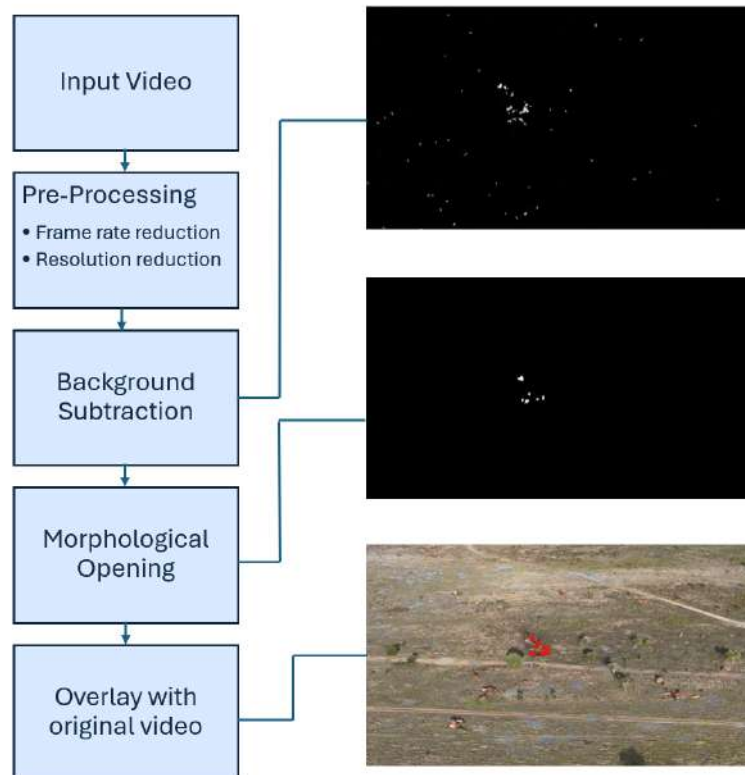


FIGURA 4.3: Esquema representativo do *Pipeline* utilizado

Pré-processamento

Com o objetivo de reduzir a complexidade computacional do processamento subsequente, foram aplicadas duas etapas de pré-processamento aos vídeos captados pelo UAV:

- **Redução do *framerate*:** o *framerate* inicial de 60 *frames* por segundo (fps) foi reduzido para 30 fps, de modo a reduzir o número de frames a processar.
- **Redução da resolução:** os vídeos foram redimensionados de 3840×2160 para 1280×720 píxeis, preservando uma qualidade visual suficiente para a detecção dos eventos relevantes, enquanto se minimiza a carga computacional.

Divisão do Dataset

Após a recolha das imagens, o *dataset* de nove vídeos foi dividido em dois grupos distintos. Foram utilizados sete vídeos (78%) para treino e dois (22%) para teste. Esta divisão do conjunto de dados alinha-se com as recomendações clássicas da literatura de visão computacional. Por exemplo, Tan et al. (2006, p.165) recomendam utilizar aproximadamente dois terços dos dados para treinar o modelo e reservar o terço restante para testá-lo, de modo a avaliar o desempenho de forma independente em dados não utilizados no treino, abordagem conhecida como método *holdout*. Outros autores, Bishop (2006), Gonzalez e Woods (2008) e Szeliski (2010), corroboram a importância de se manter uma parcela significativa dos dados apenas para teste, garantindo imparcialidade na avaliação do algoritmo.

Processamento com MOG2

A subtração de fundo foi realizada utilizando o algoritmo MOG2, com o objetivo de isolar as regiões em movimento (*foreground*) potencialmente associadas aos impactos, como o surgimento súbito de fumo e detritos.



FIGURA 4.4: Máscara binária realizada com Background Subtraction

As máscaras binárias obtidas pelo algoritmo foram posteriormente filtradas por meio de uma operação morfológica de *Opening* (erosão seguida de dilatação). Esta etapa teve como objetivo eliminar o ruído, pequenas regiões isoladas não representativas e suavizar os contornos das regiões detetadas.

FIGURA 4.5: Máscara binária após *Opening*

Treino

Foram realizados testes experimentais sobre o conjunto de treino (composto por 7 vídeos), a fim de comparar distintos valores dos parâmetros da função MOG2 (*history*, *varThreshold*) e dimensões do *kernel* para as operações morfológicas. A avaliação foi efetuada por inspeção visual. As máscaras geradas foram sobrepostas aos vídeos originais, permitindo aferir a qualidade da detecção e ajustar os parâmetros empiricamente.



FIGURA 4.6: Máscara binária em sobreposição.

O parâmetro *history*, que define o número de *frames* utilizados para a modelação do fundo, foi fixado em 150 (cerca de cinco segundos de vídeo a 30 fps), considerado suficiente para captar a variabilidade do ambiente. O limiar *varThreshold*, responsável por decidir se um píxel pertence ao fundo, foi testado com

diferentes valores, permitindo observar o equilíbrio entre sensibilidade e ruído. Adicionalmente, a detecção de sombras foi desativada, uma vez que se verificou que esta funcionalidade originava um número elevado de falsos positivos. Por fim, o tamanho do *kernel* utilizado nas operações morfológicas foi variado de modo a avaliar a sua influência na eliminação de ruído e suavização das regiões detetadas.

4.3 Detecção com *Deep Learning*

A segunda abordagem para detecção de impactos recorreu a um modelo de DL, baseado no modelo YOLOv8. Este método permite identificar e localizar objetos diretamente a partir de imagens, sem necessidade de modelação explícita do fundo. A implementação foi realizada com recurso à biblioteca *Ultralytics*, utilizando a versão YOLOv8n, treinada do zero com imagens anotadas manualmente.

4.3.1 Justificação do Método Escolhido

A arquitetura YOLOv8 divide-se em três componentes principais: *backbone*, *neck* e *head*, ((Yaseen, 2024)).

O *backbone* da arquitetura YOLOv8 é a componente responsável pela extração hierárquica de características visuais a partir da imagem de entrada. A sua função principal é transformar a informação bruta dos píxeis em representações progressivamente mais abstratas e informativas, que servirão de base para a detecção final. O processamento inicia-se com um bloco *Conv*, que realiza a primeira convolução sobre a imagem original, seguido por uma sequência de blocos *C2f*, intercalados com novas convoluções e operações de redução da resolução espacial (*stride* 2). Esta estrutura permite à rede manter a eficiência computacional, aumentando simultaneamente a profundidade semântica dos *feature maps*.

Cada bloco *C2f* inclui subestruturas do tipo *Bottleneck*, que utilizam caminhos residuais inspirados nas *ResNet*, facilitando a propagação do gradiente e a aprendizagem em profundidade. À medida que os dados avançam no *backbone*, a resolução espacial diminui (de 640×640 para 320×320 , 160×160 , assim por diante até 20×20), garantindo detecção multiescala, enquanto o número de canais aumenta, permitindo a codificação de padrões cada vez mais complexos. O *backbone* termina com o módulo *SPPF* (*Spatial Pyramid Pooling Fast*), que combina informações contextuais de diferentes escalas através de múltiplas operações de *max pooling*, reforçando a robustez da rede na detecção de objetos com dimensões e posições variadas.

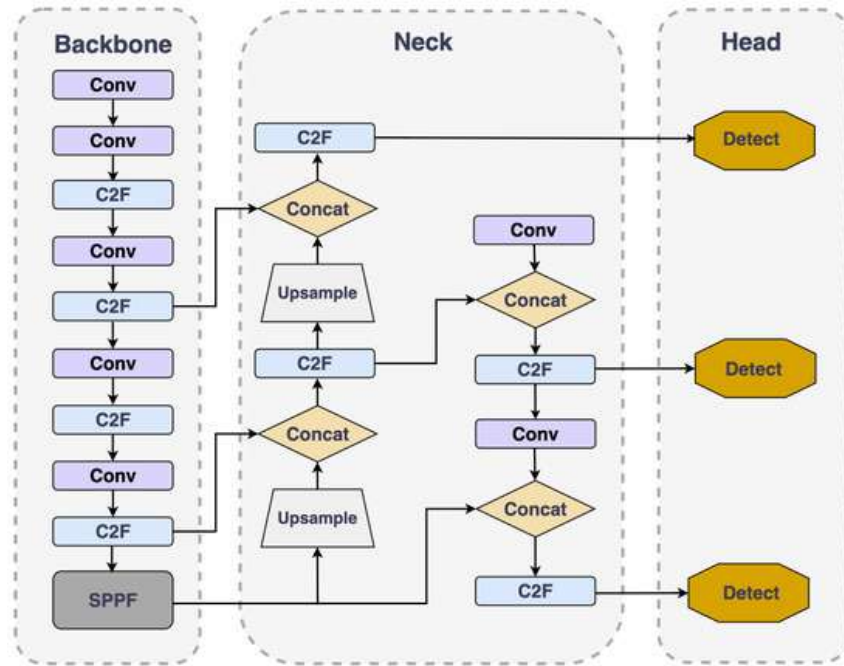


FIGURA 4.7: Estrutura do YOLOv8. Adaptado de Yao et al. (2024)

O *neck* é responsável por fundir as características extraídas em diferentes escalas e preparar os dados para a etapa de previsão. Embora a arquitetura YOLOv8 não represente explicitamente o *neck* em diagramas modulares, as suas funções estão presentes numa sequência de operações que incluem *upsampling*, *concatenação* e blocos adicionais do tipo *C2f*, aplicados sobre os *feature maps* de diferentes resoluções. Esta fase permite que a rede combine informação semântica de níveis profundos com detalhes espaciais preservados em resoluções mais altas, essencial para uma detecção multiescala eficaz.

A *head* da arquitetura é responsável pelas previsões finais da rede, nomeadamente a localização (caixas delimitadoras), a confiança e a classificação dos objetos detetados. No YOLOv8, a *head* adota uma abordagem desacoplada (*decoupled head*), com caminhos distintos para regressão (posição e dimensões da caixa) e classificação (categoria do objeto). Cada ramificação inclui um bloco *Detect*, que aplica convoluções sobre as diferentes resoluções espaciais para gerar as previsões finais em cada célula da grelha.

Importa destacar que o YOLOv8 segue uma abordagem *anchor-free*, ou seja, não utiliza caixas de referência pré-definidas (anchor boxes). Em vez disso, prevê diretamente os parâmetros das caixas delimitadoras, o que simplifica o processo de treino, reduz os hiperparâmetros e melhora a generalização.

O modelo também foi projetado com a aplicação prática em mente. A Ultralytics disponibiliza o YOLOv8 em diversos tamanhos pré-treinados (p. ex., YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l, YOLOv8x) que permitem ao utilizador escolher o melhor equilíbrio entre velocidade e eficácia de acordo com os seus requisitos de aplicação, ((Ultralytics, 2023b)). As versões menores (nano, small) têm menos camadas e parâmetros, conferindo maior velocidade e adequando-se a dispositivos com recursos limitados, enquanto versões maiores (large, xlarge) alcançam maior precisão em benchmarks ao custo de mais peso computacional.

Abaixo, a Tabela 4.5 apresenta uma comparação entre as diferentes variantes do modelo YOLOv8, evidenciando a sua evolução em termos de precisão e custo computacional. As métricas de precisão foram obtidas através da validação dos modelos no conjunto de dados COCO (*Common Objects in Context*), utilizando imagens de entrada com resolução de 640×640 píxeis. Os dados foram adaptados com base na documentação oficial disponibilizada pela Ultralytics, (Ultralytics, 2023a).

Modelo	Input (px)	mAP _{50:95} (%)	CPU (ms)	GPU A100 (ms)	Parâmetros (M)	FLOPs (B)
YOLOv8n	640	37.3	80.4	0.99	3.2	8.7
YOLOv8s	640	44.9	128.4	1.20	11.2	28.6
YOLOv8m	640	50.2	234.7	1.83	25.9	78.9
YOLOv8l	640	52.9	375.2	2.39	43.7	165.2
YOLOv8x	640	53.9	479.1	3.53	68.2	257.8

TABELA 4.5: Comparação entre variantes do YOLOv8. Dados adaptados da documentação oficial da Ultralytics.

Optou-se pela utilização do modelo YOLOv8 por ser, à data, a versão mais estável, documentada e amplamente suportada pela comunidade, com implementação oficial da Ultralytics. Embora existam versões mais recentes, como o YOLOv11 e YOLOv12, estas ainda se encontram em fase de avaliação e com menos documentação e suporte consolidado. Assim, o YOLOv8 representa uma escolha equilibrada entre desempenho, facilidade de utilização e confiança nos resultados obtidos.

Tendo em conta os objetivos deste trabalho, foi utilizada a versão YOLOv8n (nano), a mais leve da família YOLOv8. Devido à sua baixa exigência computacional, esta versão revela-se a mais adequada para integração em plataformas embarcadas, como UAVs.

4.3.2 Pipeline

O *pipeline* utilizado integra as fases de pré-processamento, anotação, aumento de dados, divisão do *dataset* e treino.

Pré-processamento e Anotação

Dos exercícios realizados no regimento de artilharia nº 5 e no campo militar de Santa Margarida, extraíram-se 179 imagens de impactos dos vídeos captados. A este conjunto de dados, acrescentou-se ainda 24 imagens de impactos capturadas com UAV, recolhidas de fontes abertas. Em seguida, todas as 203 imagens foram redimensionadas para 640×640 píxeis, em conformidade com os requisitos da arquitetura YOLOv8.

Todas as imagens foram posteriormente anotadas manualmente com a ferramenta *LabelImg*, utilizando uma única classe, impacto. As *bounding boxes* foram traçadas manualmente em cada imagem, e os dados exportados no formato YOLO (ficheiros `.txt` com a respetiva anotação). A Figura 4.8 ilustra um exemplo do processo de anotação.

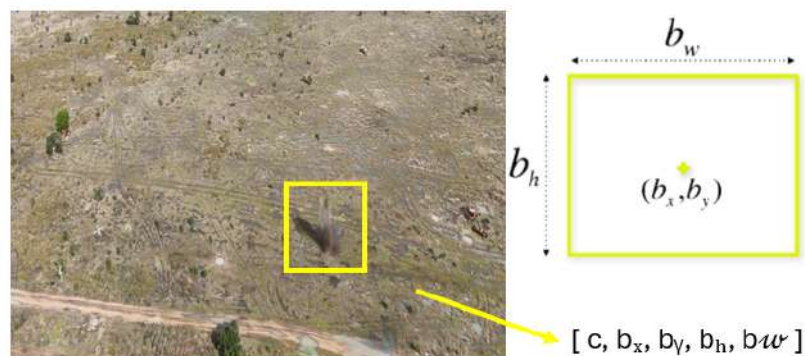


FIGURA 4.8: Exemplo de imagem e formato de anotação

Data Augmentation

Face o número bastante reduzido de imagens, para aumentar a diversidade do conjunto de treino e melhorar a generalização do modelo, foi aplicada a técnica de *data augmentation* com recurso à biblioteca *Albumentations*. O *data augmentation* consiste na aplicação de transformações às imagens originais com o objetivo de aumentar artificialmente o tamanho e a diversidade do conjunto de dados, contribuindo para melhorar a capacidade de generalização do modelo e reduzir o risco de sobreajuste.

Foi aplicada inversão horizontal (*HorizontalFlip*), permitindo simular diferentes direções dos impactos. A variação de brilho e contraste (*RandomBrightnessContrast*) visou replicar alterações de iluminação, frequentes em observação aérea. A rotação aleatória limitada a 15 graus (*Rotate*) introduziu variações na orientação do impacto, sem comprometer a coerência visual. A escala aleatória até 20% (*RandomScale*) simulou aproximações e afastamento do UAV em relação aos impactos. Foram ainda aplicados efeitos de desfocagem por movimento (*MotionBlur*), ruído gaussiano (*GaussNoise*) e *Contrast Limited Adaptive Histogram Equalization* (CLAHE), que aumenta o contraste de forma local em pequenas regiões da imagem, realçando detalhes sem amplificar excessivamente o ruído.

Estas transformações permitiram aumentar o conjunto de dados a partir das imagens originais, preservando os padrões visuais dos impactos e enriquecendo o conjunto de treino com variações realistas.

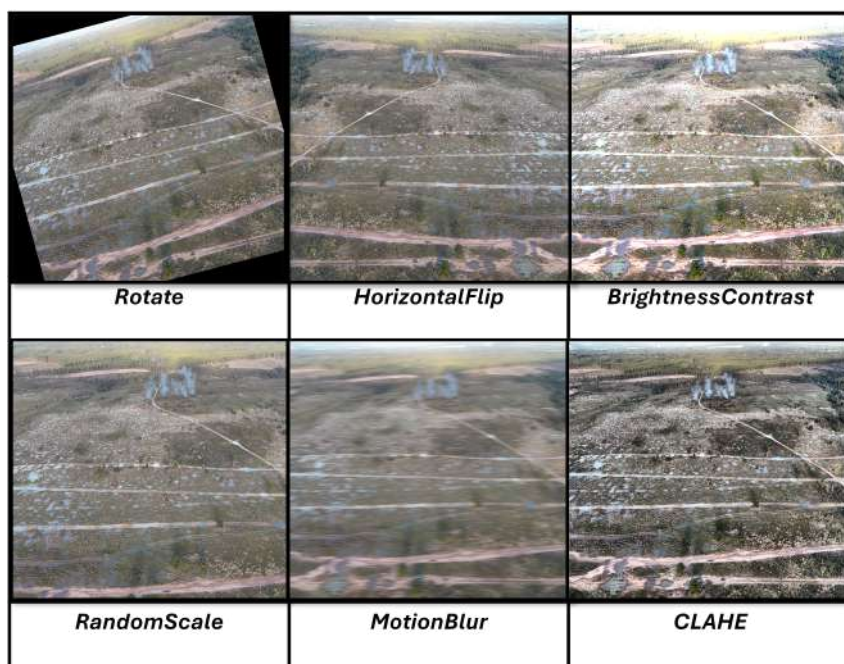


FIGURA 4.9: Transformações aplicadas no *Data Augmentation*

Após a fase de *data augmentation*, em que cada imagem original gerou entre 7 a 8 variações aumentadas (além da versão original), o conjunto expandiu-se para um total de 1670 imagens, assegurando maior diversidade e robustez no treino do modelo.

Divisão do *Dataset*

Para o treino de um modelo de *deep learning*, a divisão do dataset deve respeitar a separação entre treino, validação e teste, de forma a garantir a aprendizagem eficaz, a monitorização do desempenho durante o treino e a avaliação da capacidade de generalização. Segundo Goodfellow et al. (2016b), é comum alocar aproximadamente 60–80% dos dados para treino, 10–20% para validação e 10–20% para teste, dependendo da dimensão do dataset e dos requisitos específicos da tarefa.

Seguindo as boas práticas de separação dos conjuntos de dados e redução do risco de *overfitting*, após o pré-processamento e o aumento de dados, o conjunto final foi dividido em três subconjuntos: 1169 imagens para treino (73.3%), 334 para validação (20.9%) e 167 para teste (5.8%). Esta divisão visou assegurar um treino eficaz do modelo, com monitorização do desempenho em dados não vistos durante o treino (validação), bem como uma avaliação final em dados totalmente independentes (teste).

TABELA 4.6: Caracterização do conjunto de dados anotado e dividido para treino, validação e teste.

Origem	Nº de imagens	Com <i>augmentation</i>	Treino	Validação	Teste
Regimento de Artilharia nº 5 (RA5)	93	765	535	153	77
Campo Militar de Santa Margarida (CMSM)	86	708	496	142	70
Imagens da Internet	24	197	138	39	20
Total	203	1670	1169	334	167

Treino

O treino decorreu em duas fases distintas. Na primeira, o modelo foi treinado durante 100 épocas, com um *batch size* de 128, ou seja, os dados foram processados em blocos de 128 imagens por iteração de treino. Utilizou-se exclusivamente CPU, e os parâmetros mantiveram-se nos valores predefinidos pela biblioteca Ultralytics: *learning rate* inicial de 0.01, otimizador *Stochastic Gradient Descent* (SGD), que atualiza os pesos com base em gradientes de erro estimados em *mini-batches* aleatórios, *momentum* de 0.937, que acelera o treino ao suavizar flutuações, e *weight decay* de 0.0005, utilizado para prevenir sobreajuste ao penalizar pesos excessivos.

Nesta fase, observou-se uma evolução progressiva das métricas de desempenho: a *precision* (percentagem de deteções corretas entre todas as deteções) aumentou de 0.36 para 0.68, enquanto o mAP@50 (*mean Average Precision* para $\text{IoU} \geq 0.5$), subiu de 0.07 para 0.55.

Na segunda fase, o treino foi retomado a partir dos pesos finais da fase anterior e prolongado por mais 400 épocas. Esta etapa permitiu refinar os pesos aprendidos e melhorar a generalização do modelo. Nas primeiras épocas desta fase observou-se flutuação nos valores das métricas, um fenómeno comum quando o treino é retomado após interrupção, associado a reajustes nos gradientes. Rapidamente, porém, os valores estabilizaram com desempenho elevado: precisão de 0.986, *recall* de 0.963 (percentagem de impactos corretamente detetados entre todos os impactos reais), AP@50 de 0.987 e AP@50-95 de 0.753. Embora a interrupção introduza pequenas flutuações iniciais ao retomar o processo, como visível no gráfico da Figura 4.10, a literatura confirma que tal prática não compromete os resultados finais, sendo funcionalmente equivalente a um treino contínuo de 500 épocas (Eisenman et al., 2022).

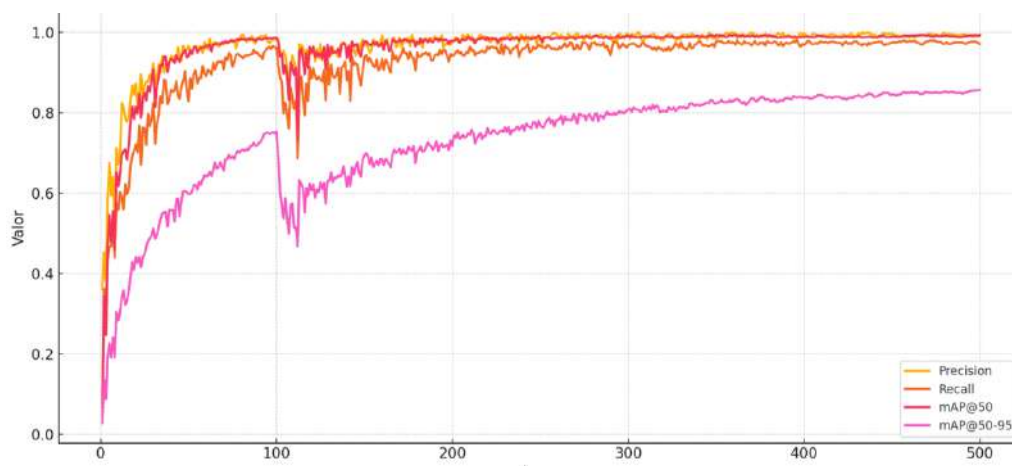


FIGURA 4.10: Evolução das métricas.

As perdas de validação, responsáveis por quantificar os erros cometidos pelo modelo em cada componente da predição, mantiveram-se baixas e estáveis: a *box loss* mede o erro na regressão das coordenadas das *bounding boxes*; a *cls loss* corresponde ao erro na classificação da classe dos objetos detetados (neste caso, apenas impacto); e a *dfl loss* (*Distribution Focal Loss*) refina a precisão da localização ao modelar a distribuição das bordas das caixas. Os respetivos valores finais foram 0.97, 0.48 e 1.04, sugerindo que o modelo atingiu um estado de convergência com bom equilíbrio entre sensibilidade e especificidade.



FIGURA 4.11: Evolução das *losses* da validação durante o treino do modelo

4.4 Seguimento de pontos de impacto (ByteTrack)

A detecção de impactos em vídeo gera múltiplas ocorrências da mesma explosão ao longo de vários frames consecutivos. Se cada detecção fosse contabilizada isoladamente, uma única ocorrência poderia resultar em vários pontos de impacto redundantes. Para evitar este problema, é necessário um mecanismo de seguimento que assegure a coerência temporal das detecções, agregando-as de forma consistente ao longo do tempo.

Neste trabalho, esse papel é desempenhado pelo algoritmo ByteTrack, uma solução de *multi-object tracking* (MOT) que permite associar as detecções em sequências sucessivas de frames, garantindo que cada impacto real é contabilizado apenas uma vez, enquanto se reduz a probabilidade de falsos positivos.

4.4.1 Justificação do Método

O ByteTrack é um algoritmo de Multi-Object Tracking concebido para alcançar alta precisão e desempenho em tempo real ao repensar o processo de associação de dados, (Zhang et al. (2021)). Contrariamente aos métodos tradicionais que descartam detecções de baixa confiança, o ByteTrack retém-nas e emprega uma estratégia de associação em duas etapas utilizando a métrica *Background-aware Yet Tiny and Efficient* (BYTE). Esta abordagem reduz significativamente as trocas de identidade e melhora a robustez do tracking, especialmente em ambientes complexos em que ocorrem oclusões de objetos, (Figura 4.12). Zhang et al. (2022) ampliaram este

trabalho ao integrar o ByteTrack com *frameworks* de detecção avançados, reforçando a sua aplicabilidade em sistemas em tempo real.

O ByteTrack utiliza uma abordagem de *tracking by detection*, onde a entrada para o tracker é um conjunto de detecções de *bounding boxes* $D_t = \{d_1, d_2, \dots, d_n\}$ em cada frame t , com cada detecção $d_i = (x_i, y_i, w_i, h_i, s_i)$ composta pelas coordenadas da caixa (centro x_i, y_i , largura w_i , altura h_i) e um *score* de confiança s_i . O tracker mantém um conjunto de trajetórias $T_t = \{T_1, T_2, \dots, T_m\}$, onde cada trajetória T_j representa o percurso de um objeto com uma identidade associada.

Deteções e Limiar de Score

O ByteTrack começa por categorizar as detecções em conjuntos de alta e baixa confiança com base num limiar τ :

- Deteções de alta confiança: $D_t^{high} = \{d_i \in D_t \mid s_i > \tau\}$
- Deteções de baixa confiança: $D_t^{low} = \{d_i \in D_t \mid s_i \leq \tau\}$

onde τ é um valor definido que pode ser ajustado conforme as características do detector, (Zhang et al. (2021)). Os trackers tradicionais descartam D_t^{low} , mas o ByteTrack retém estas detecções para uma segunda etapa de associação.

Associação de Dados em Duas Etapas

O ByteTrack utiliza um processo de associação em duas etapas, recorrendo a um filtro de *Kalman* para prever a posição futura dos objetos e a uma métrica de similaridade, por norma IoU, para associar as novas detecções aos objetos identificados nos frames anteriores.

Etapa 1: Associação de Alta Confiança

Para cada trajetória $T_j \in T_{t-1}$ do frame anterior, o Filtro de Kalman prevê o seu estado no tempo t . O vetor de estado de uma trajetória é geralmente definido como:

$$x = [x, y, w, h, \dot{x}, \dot{y}, \dot{w}, \dot{h}]^T$$

onde (x, y) é a posição central, (w, h) é o tamanho, e $(\dot{x}, \dot{y}, \dot{w}, \dot{h})$ são as velocidades. A etapa de predição atualiza o estado como:

$$\hat{x}_{t|t-1} = F \cdot x_{t-1} + w$$

onde F é a matriz de transição de estado (assumindo velocidade constante) e w é o ruído do processo.

A similaridade entre as caixas previstas das trajetórias $B_j^{\text{pred}'}$ e as detecções de alta confiança D_t^{high} é calculada usando a métrica IoU:

$$S_{i,j} = \text{IoU}(B_j^{\text{pred}}, d_i)$$

É construída uma matriz de custo C com $C_{i,j} = 1 - S_{i,j}$, e o algoritmo Húngaro resolve o problema de atribuição para associar trajetórias a detecções, garantindo a sua correspondência, (Zhang et al. (2021)).

Etapa 2: Associação de Baixa Confiança

As trajetórias não associadas na Etapa 1 são então associadas a D_t^{low} usando a mesma métrica de similaridade baseada em IoU. Esta etapa recupera objetos temporariamente oclusos ou detetados com baixa confiança, reduzindo trocas de identidade. A associação BYTE é definida como:

$$\text{BYTE}(T_j, d_i) = \text{IoU}(B_j^{\text{pred}}, d_i)$$

se $\text{IoU} > \theta$, onde θ é um limiar inferior (por exemplo, 0.1), permitindo uma correspondência mais permissiva para detecções de baixa confiança Zhang et al. (2021).

Gestão de Trajetórias

Início: Detecções não associadas em D_t^{high} iniciam novas trajetórias.

Atualização: Trajetórias associadas atualizam os seus estados do Filtro de Kalman com a detecção atribuída usando a etapa de atualização de medição:

$$x_{t|t} = \hat{x}_{t|t-1} + K(z_t - H\hat{x}_{t|t-1})$$

onde z_t é a medição (caixa de detecção), H é a matriz de observação, e K é o ganho de Kalman.

Término: Trajetórias não associadas por um número definido de frames (por exemplo, 30) são descartadas.

Para este trabalho foi escolhido o algoritmo ByteTrack por se distinguir de outros modelos de MOT pelo uso eficaz de detecções de baixa confiança, resultando em menos trajetórias perdidas e trocas de identidade. Apresenta valores de desempenho elevados em *benchmarks* como MOT17 e MOT20, com um Multiple Object Tracking Accuracy (MOTA) reportado de 80.3 no MOT17 e uma velocidade de tracking superior a 30 FPS, (Zhang et al. (2021)). Além disso, a sua simplicidade,

sem dependência de características de aparência complexas ou modelos profundos de Re-ID, torna-o computacionalmente eficiente, ideal para a finalidade em causa.

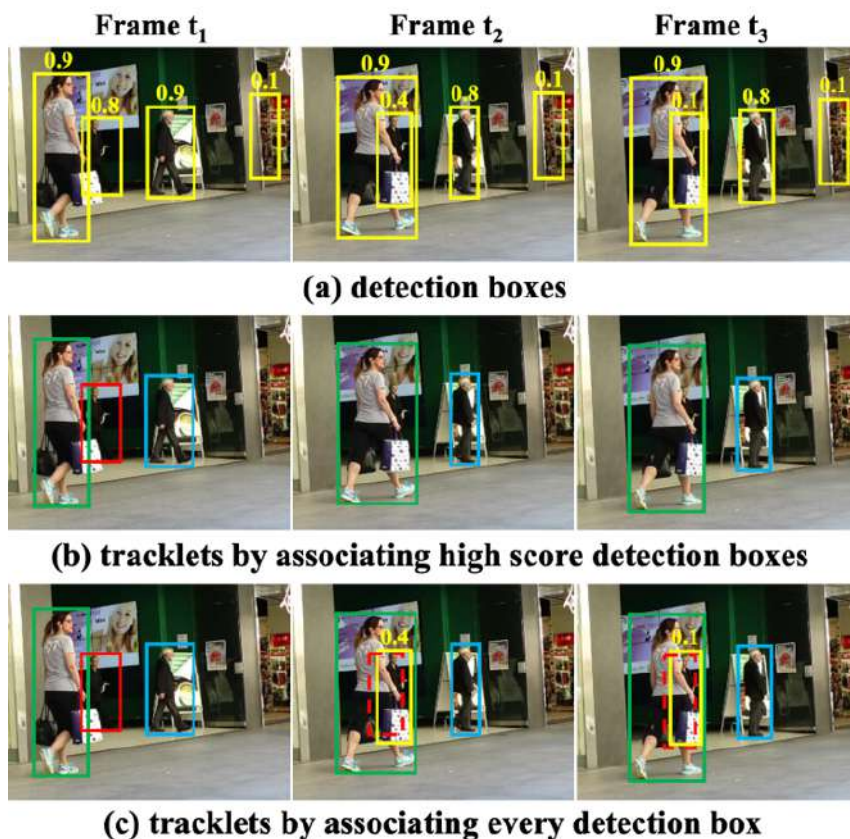


FIGURA 4.12: MOT com associações de alta e baixa confiança

4.4.2 Associação de deteções

O algoritmo *ByteTrack* foi aplicado nos dois cenários de deteção distintos: um baseado em *background subtraction* (*MOG2*) e outro recorrendo ao modelo *YOLOv8*. Em ambos os casos, o objetivo foi garantir a coerência temporal das deteções, agrupando-as em trajetórias (*tracks*) consistentes ao longo do tempo. No entanto, a sua aplicação apresentou diferenças, decorrentes da forma como as caixas de deteção foram geradas e processadas em cada método.

Associação de deteções com BS

Para associar as deteções dos impactos ao longo dos vídeos no método baseado em *background subtraction* (*MOG2*), foi desenvolvido um algoritmo de *tracking* inspirado nos princípios do *ByteTrack*, adaptado ao contexto específico

deste trabalho. A detecção inicial foi feita através da segmentação da máscara binária em sobreposição através da operação `cv2.findContours`. Os contornos obtidos através da segmentação foram convertidos em *bounding boxes* utilizando a função `cv2.boundingRect`. Esta função calcula o menor retângulo que contém o contorno detetado, resultando numa caixa definida por coordenadas (x , y , *largura*, *altura*). Cada caixa foi tratada como uma detecção individual de um possível impacto, sendo-lhe atribuída, por convenção, a classe *impacto*.

As *bounding boxes* extraídas em cada *frame* foram associadas ao longo do tempo com base em dois critérios: *IoU* e distância do cosseno. O *IoU* mede a sobreposição entre duas caixas e a distância do cosseno avalia a semelhança entre vetores formados pelas coordenadas das *bounding boxes*, isto é, no contexto do *tracking* cada caixa é representada por um vetor com quatro componentes: coordenadas do centro da caixa (posição) e respetiva largura e altura. Se duas caixas sucessivas tiverem vetores com direções semelhantes, significa que mantêm uma posição e dimensão coerentes ao longo do tempo. Um valor baixo de distância do cosseno indica forte semelhança entre caixas; um valor próximo de 1 indica orientações muito diferentes.

As detecções sucessivas de um mesmo impacto foram associadas ao longo do tempo, formando *tracks* que representam a trajetória temporal de cada evento. O rastreamento foi realizado em tempo real, *frame a frame*, com os seguintes parâmetros: Foi considerado um intervalo entre 5 píxeis² e 5 000 000 píxeis² como área válida dos contornos, de modo a filtrar ruído e ignorar objetos demasiado pequenos ou irrelevantes. Para que uma detecção fosse associada a uma anterior, exigiu-se uma *IoU* mínima de 20% entre as caixas, o que corresponde a uma correspondência moderadamente exigente, suficiente para lidar com ligeiras variações de posição e forma entre *frames* sucessivos. Adicionalmente, avaliou-se a consistência espacial entre caixas com base na distância do cosseno entre vetores definidos pelas coordenadas do centro e pelas dimensões de cada caixa. Foi aceite uma diferença angular até 0,8, permitindo uma associação flexível, justificada pela variabilidade e brevidade visual dos impactos no vídeo. Foi estabelecido 180 como o número máximo de *frames* consecutivos em que um *track* pode deixar de ser atualizado antes de ser eliminado, o que corresponde a cerca de 6 segundos de vídeo, assumindo uma taxa de 30 *fps*. Este limite permite tolerar interrupções temporárias na visibilidade do impacto. Mesmo que um *track* continue a ser atualizado, é removido se ultrapassar uma duração de 330 *frames*, o equivalente a 11 segundos, por se considerar que, após esse período, já não representa uma ocorrência relevante. Por fim, qualquer trajetória com pelo menos uma detecção válida foi considerada. Esta abordagem garante que impactos subtis não sejam ignorados. A Tabela 4.7 resume os parâmetros definidos.

TABELA 4.7: Resumo dos parâmetros definidos para o *tracking*

Parâmetro	Nome da variável	Valor
Área mínima	<i>min_area</i>	5 píxeis ²
Área máxima	<i>max_area</i>	5 000 000 píxeis ²
Sobreposição mínima	<i>iou_threshold</i>	0,2
Distância do cosseno máxima	<i>max_cosine_distance</i>	0,8
Máx. <i>frames</i> sem atualização	<i>max_disappeared</i>	180
Duração máxima do <i>track</i>	<i>max_age</i>	330 <i>frames</i>
N.º mínimo de atualizações	<i>min_hits</i>	1

Adicionalmente, para lidar com falhas temporárias de detecção, implementou-se uma fusão baseada na distância euclidiana entre novas caixas e posições iniciais de *tracks* recentes (com menos de 4 segundos). Se a distância fosse inferior a 100 píxeis, a nova caixa era associada ao *track* existente. Esta adaptação foi feita para evitar que, no caso de haver separação do material projetado, essa separação de partículas fosse considerada como uma nova detecção, como representado na Figura 4.13.

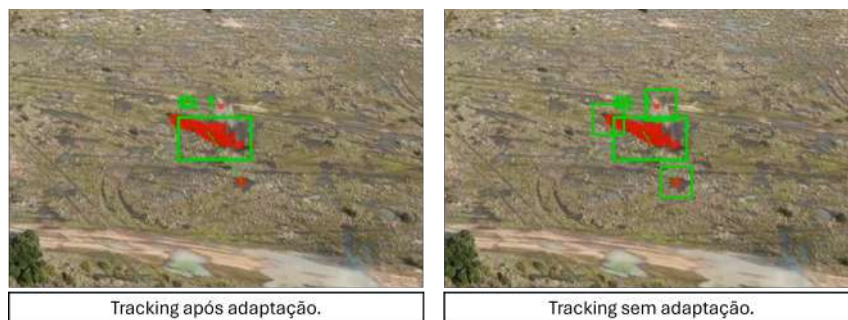


FIGURA 4.13: Exemplo do efeito da adaptação ao algoritmo *ByteTrack* na detecção de um impacto.

Associação de detecções com YOLOv8

No segundo método, as detecções foram obtidas diretamente através do modelo treinado YOLOv8. Neste caso, cada *frame* foi processado individualmente pelo modelo, que produziu *bounding boxes* e pontuações de confiança associadas à classe *impacto*. Estas caixas foram imediatamente utilizadas como entradas para o algoritmo de *tracking*, sem necessidade de segmentação ou processamento adicional.

A associação temporal das detecções foi realizada com o mesmo algoritmo de *tracking* descrito anteriormente, baseado nos princípios do *ByteTrack*. Foram

mantidos todos os parâmetros e critérios definidos para o caso anterior, incluindo os limites de área, *IoU* mínimo, distância do cosseno, duração máxima de um *track* e número máximo de *frames* sem atualização.

Ao contrário do que acontece com o método baseado em *MOG2*, não foi necessário aplicar a adaptação adicional para fusão de detecções próximas. A robustez do YOLOv8 permite manter uma única detecção coesa ao longo do tempo, mesmo quando ocorre dispersão momentânea do material no impacto. Isso simplificou a associação e reduziu o risco de falsos positivos decorrentes de projeção de partículas dispersas.

4.5 Conclusão

Neste capítulo foi apresentada a metodologia desenvolvida para a detecção automática de impactos de artilharia com UAVs. Foram exploradas duas abordagens distintas: uma baseada em visão computacional clássica, através do algoritmo *MOG2* de *background subtraction*, e outra baseada em *deep learning*, utilizando o modelo YOLOv8. Cada abordagem implicou um processo próprio de preparação e tratamento de dados, desde o pré-processamento até ao treino dos algoritmos.

Adicionalmente, foi integrado o algoritmo ByteTrack como solução de *multi-object tracking*, permitindo associar as detecções ao longo do tempo e eliminar redundâncias, assegurando a contagem consistente dos impactos. Esta etapa revelou-se essencial para transformar as detecções individuais em trajetórias temporais coerentes, aumentando a robustez dos resultados.

A metodologia aqui descrita estabeleceu as bases experimentais para a avaliação e comparação de desempenho entre os dois paradigmas de detecção: VC clássica e DL.

Capítulo 5

Resultados

Este capítulo apresenta e discute os resultados obtidos com os dois métodos de deteção de impactos considerados: a abordagem baseada em *background subtraction* (MOG2) e o modelo de *deep learning* YOLOv8.

5.1 Resultados obtidos com o método baseado em Background Subtraction

Nesta secção são apresentados os resultados obtidos com o método clássico de *background subtraction* (MOG2). O objetivo é identificar as configurações de parâmetros mais adequadas e avaliar a sua capacidade para detetar impactos em vídeo. Para tal, realizaram-se testes experimentais variando diferentes combinações de parâmetros, seguidos da avaliação final em vídeos reservados para teste.

5.1.1 Testes experimentais e seleção de parâmetros

Foram realizados testes experimentais para avaliar diferentes combinações de parâmetros do algoritmo MOG2. Manteve-se o valor de *history* fixo em 150 (aproximadamente cinco segundos de vídeo a 30 fps), considerado suficiente para modelar a variabilidade do fundo sem comprometer a capacidade de reação do modelo. A tabela da Figura 5.1 apresenta os resultados obtidos, utilizando as métricas de avaliação: *True Positives* (TP), correspondentes a impactos corretamente detetados; *False Positives* (FP), referentes a deteções incorretas de impactos inexistentes; e *False Negatives* (FN), que indicam impactos reais não detetados.

	Vídeo 1	Vídeo 2	Vídeo 3	Vídeo 4	Vídeo 5	Vídeo 6	Vídeo 7	Total
H150_V12_K3								
TP	5	22	17	1	0	1	0	46
FP	7	4	14	0	0	2	0	27
FN	0	0	0	0	1	0	1	2
H150_V12_K4								
TP	5	22	17	1	0	1	0	46
FP	2	0	0	0	0	0	0	2
FN	0	0	0	0	1	0	1	2
H150_V12_K5								
TP	5	22	15	1	0	1	0	44
FP	2	0	0	0	0	0	0	2
FN	0	0	2	0	1	0	1	4
H150_V16_K3								
TP	5	22	17	1	0	1	0	46
FP	6	4	6	0	0	4	0	20
FN	0	0	0	0	1	0	1	2
H150_V16_K4								
TP	5	22	17	1	0	1	0	46
FP	0	2	0	0	0	0	0	2
FN	0	0	0	0	1	0	1	2
H150_V16_K5								
TP	5	20	14	1	0	0	0	40
FP	1	0	0	0	0	0	0	1
FN	0	2	3	0	1	1	1	8
H150_V24_K3								
TP	5	22	16	1	0	1	0	45
FP	4	2	2	0	0	3	0	11
FN	0	0	1	0	1	0	1	3
H150_V24_K4								
TP	5	21	16	1	0	1	0	44
FP	1	0	0	0	0	0	0	1
FN	0	1	1	0	1	0	1	4

FIGURA 5.1: Resultados obtidos para diferentes configurações do algoritmo MOG2. TP – impactos corretamente detetados; FP – falsos positivos; FN – impactos não detetados. Note-se que o número total de impactos reais em cada vídeo corresponde sempre a $TP + FN$.

Os testes experimentais indicaram que as combinações mais eficazes ocorreram para *varThreshold* entre 12 e 16, com *kernel* de 3×3 ou 4×4 píxeis nas operações morfológicas. A configuração H150_V12_K4 apresentou um equilíbrio particularmente favorável entre sensibilidade e precisão, com apenas dois falsos positivos ($FP = 2$) e dois falsos negativos ($FN = 2$), mantendo todos os impactos corretamente detetados ($TP = 46$). Configurações com *varThreshold* mais elevado (como V24) resultaram em menor sensibilidade, enquanto valores mais baixos originaram maior ruído.

Avaliação em vídeos de teste

Para avaliar o desempenho do algoritmo MOG2 na detecção de impactos, foram utilizados dois vídeos reservados exclusivamente para teste. O primeiro vídeo foi capturado durante um exercício de tiro no Campo Militar de Santa Margarida, contendo um total de sete impactos. O segundo vídeo, com um único impacto, foi obtido no Regimento de Artilharia nº 5, em Vendas Novas. Ambos os vídeos têm uma taxa de 30 fps.

Aos vídeos de teste foram aplicados os parâmetros que obtiveram melhor desempenho nos testes exploratórios, resumidos na Tabela 5.1.

TABELA 5.1: Parâmetros definidos para o algoritmo MOG2 nos vídeos de teste

Parâmetro	Valor
history	150
varThreshold	16
detectShadows	False

Após a segmentação inicial, aplicaram-se operações morfológicas de *opening* com *kernel* de 4×4 píxeis, visando reduzir ruído e unir regiões fragmentadas.

A avaliação foi realizada com recurso a *ground truths* construídas manualmente, consistindo em *bounding boxes* que identificam a ocorrência e localização dos impactos. De forma a evitar a análise exaustiva de todos os *frames* e tendo em conta o objetivo prático da detecção, adotou-se uma abordagem baseada em janelas temporais. Cada impacto foi associado a uma janela temporal de 15 frames (0,5 s). Uma detecção foi considerada TP se, dentro da janela, existisse sobreposição com a caixa da *ground truth* (Figura 5.8).

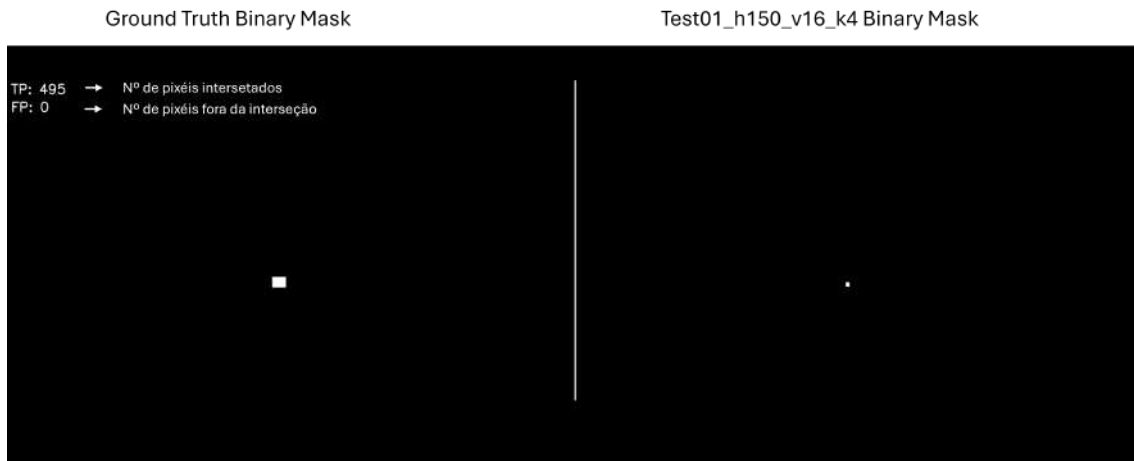


FIGURA 5.2: Comparação de um frame de uma deteção com a *ground truth*.

As deteções produzidas fora de qualquer janela de impacto foram consideradas FP, refletindo situações em que o algoritmo reagiu a ruído ou movimentos irrelevantes. Por fim, sempre que não foi registada qualquer deteção válida dentro da janela de um impacto, considerou-se um FN, indicando a falha do algoritmo em detetar um impacto real.

Com base nos valores obtidos, foram calculadas as métricas de *Precision*, que indica a proporção de deteções corretas em relação ao total de alertas emitidos pelo sistema, e *Recall* que mede a capacidade do sistema em detetar todos os impactos efetivamente ocorridos:

$$\text{Precision} = \frac{TP}{TP + FP}, \quad \text{Recall} = \frac{TP}{TP + FN}$$

TABELA 5.2: Desempenho do algoritmo MOG2 na deteção de impactos (vídeos de teste).

Vídeo	TP	FP	FN	Precision	Recall
Santa Margarida ¹ (7 impactos)	7	0	0	1,00	1,00
Vendas Novas ² (1 impacto)	1	0	0	1,00	1,00
Total (8 impactos)	8	0	0	1,00	1,00

¹<https://youtu.be/86RYKuUtXTM>

²https://youtu.be/tdrqivu_1HQ

Os resultados demonstram um desempenho excelente do algoritmo MOG2 nas condições de teste, com *Precision* e *Recall* de 100%. Todos os impactos foram corretamente detetados, sem ocorrência de falsos positivos ou falsos negativos.

5.2 Resultados com o Algoritmo YOLOv8

A avaliação do modelo YOLOv8 foi realizada sobre o conjunto de teste definido previamente no capítulo da metodologia. Importa referir que este conjunto de teste não corresponde diretamente aos dois vídeos usados na avaliação do algoritmo MOG2, embora partilhe algumas imagens extraídas dos mesmos.

O modelo foi avaliado com os pesos obtidos na melhor época (`best.pt`), após 500 épocas de treino. A avaliação foi conduzida com a ferramenta oficial da Ultralytics, que aplica o modelo ao conjunto de teste e calcula as métricas *Precision*, *Recall* e *mAP* (*mean Average Precision*) com diferentes limiares de IoU.

Os resultados obtidos no conjunto de teste foram os seguintes (Tabela 5.3):

TABELA 5.3: Resultados do YOLOv8 no conjunto de teste

Métrica	Valor
Precision	0,980
Recall	0,973
mAP@0.5	0,993
mAP@0.5:0.95	0,859

Estes resultados revelam um desempenho extremamente elevado, com elevada fiabilidade nas predições e excelente correspondência espacial entre as predições e os impactos reais.

A Figura 5.3 mostra que o desempenho global do modelo é maximizado com um limiar de confiança de aproximadamente 0,4, onde o F1-score atinge 0,98. O valor mantém-se elevado numa larga gama de confiança, indicando robustez e estabilidade nas predições.

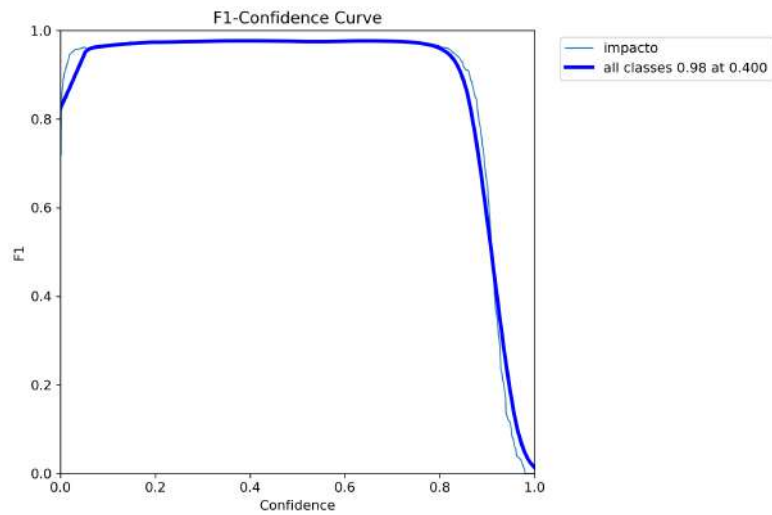


FIGURA 5.3: Curva F1-Confidence obtida na validação do modelo YOLOv8.

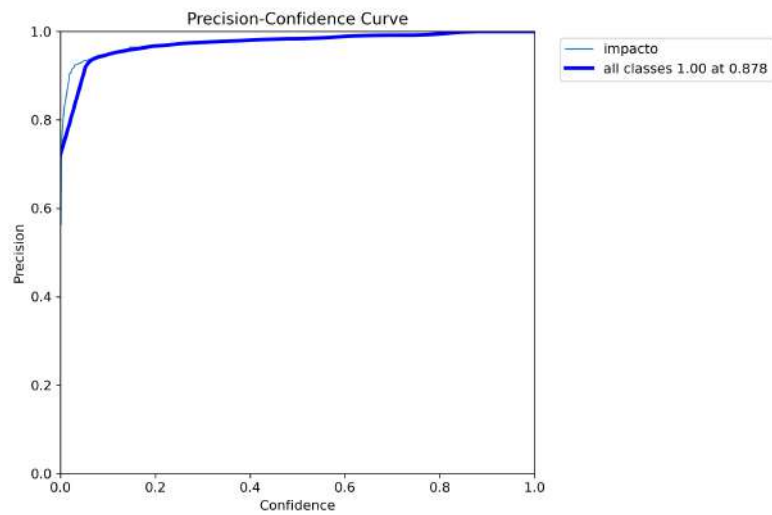


FIGURA 5.4: Curva Precision-Confidence obtida na validação do modelo YOLOv8.

A curva da Figura 5.4 mostra que a *Precision* atinge o valor máximo de 1,00 com uma confiança superior a 0,878. Isto revela que as predições mais confiantes são extremamente fiáveis, com praticamente ausência de falsos positivos.

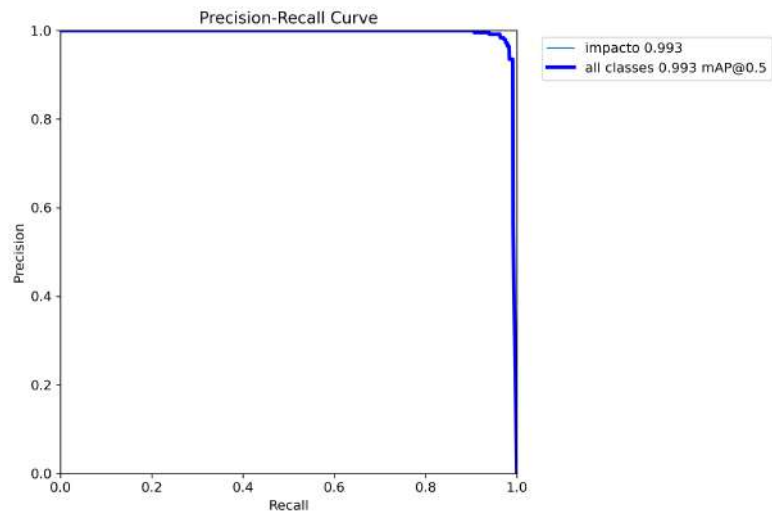


FIGURA 5.5: Curva Precision-Recall obtida na validação do modelo YOLOv8.

A Figura 5.5 apresenta a curva *Precision-Recall*, cuja área sob a curva atinge um valor de $mAP@0.5$ de 0,993. A proximidade do traçado ao canto superior direito evidencia o excelente compromisso entre sensibilidade e precisão.

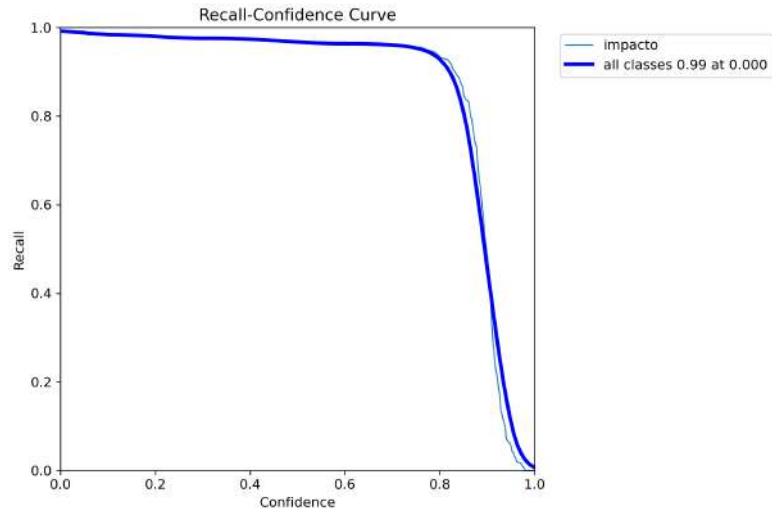


FIGURA 5.6: Curva Recall-Confidence obtida na validação do modelo YOLOv8.

Por fim, a curva da Figura 5.6 mostra que o *Recall* se mantém elevado (superior a 0,95) até cerca de 0,7 de confiança. A partir desse ponto, verifica-se uma

queda acentuada, o que indica que limiares de confiança demasiado elevados podem comprometer a capacidade de deteção.³

Os resultados apresentados demonstram que o modelo mantém um desempenho elevado e consistente para uma ampla gama de valores de confiança, permitindo adaptar o seu comportamento conforme a necessidade. Valores mais baixos de confiança favorecem uma maior taxa de deteção, enquanto valores mais elevados reduzem significativamente a probabilidade de falsos positivos. Importa, contudo, reconhecer que as métricas obtidas refletem também a natureza relativamente limitada e pouco diversificada do *dataset*, em que muitas imagens partilham características visuais semelhantes. Esse fator contribui para facilitar a tarefa do modelo e ajuda a explicar os valores excecionalmente elevados de desempenho observados.

5.3 Comparação dos Métodos

Após a implementação do algoritmo de associação inspirado no ByteTrack, procedeu-se à comparação entre os dois métodos completos. A avaliação foi feita com base no vídeo de teste gravado no Campo Militar de Santa Margarida, no qual foram registados sete impactos de granadas de alto explosivo (HE) de morteiros de 81mm.

No caso do método baseado no *YOLOv8*, consideraram-se apenas as deteções com confiança mínima de 0,35. Este método detetou corretamente 5 dos 7 impactos. No entanto, gerou uma deteção falsa, ao interpretar o reflexo de uma poça de água como um impacto, (figura 5.7). No total, foram produzidas 6 deteções, sendo uma incorreta (FP).

O método baseado em *Background Subtraction* com GMM apresentou melhor desempenho. Todos os 7 impactos foram corretamente detetados e não se registaram falsos positivos. O algoritmo revelou-se robusto face a ruído visual, como reflexos, e sensível à presença de impactos reais.

Em resumo, o método baseado no *YOLOv8* obteve uma taxa de deteção verdadeira (*Recall*) de aproximadamente 71%, com uma precisão (*Precision*) de cerca de 83%. Por sua vez, o método baseado em *Background Subtraction* (BS) com GMM atingiu uma taxa de deteção de 100% em ambos os indicadores. Estes

³Uma vez que a tarefa envolve apenas uma classe, a utilização de uma matriz de confusão não acrescentaria informação relevante para a análise.



FIGURA 5.7: Falso Positivo. Reflexo de uma poça de água considerada como uma detecção de impacto.

resultados indicam que, neste cenário, a abordagem baseada em GMM foi mais eficaz para a detecção autónoma de impactos de artilharia.

TABELA 5.4: Comparação dos dois métodos no vídeo de teste (7 impactos)

Método	Impactos Detetados	Falsos Positivos	Recall	Precision
YOLOv8	5 / 7	1	71 %	83 %
BS (GMM)	7 / 7	0	100 %	100 %

Apesar dos resultados muito positivos obtidos pelo método baseado em *Background Subtraction*, importa salientar que o cenário do vídeo de teste favorece particularmente este tipo de abordagem. O ambiente é visualmente estável, sem elementos dinâmicos como pessoas, veículos ou fumo em movimento, e o drone permanece praticamente imóvel, com a câmara fixa, sem rotações ou zooms. Estas condições reduzem significativamente a probabilidade de falsos positivos e permitem ao GMM construir um modelo de fundo bastante fiável.

No entanto, estas circunstâncias não refletem necessariamente os desafios da detecção num ambiente real de combate. Assim, para uma avaliação mais justa e abrangente, foi elaborada uma nova compilação de vídeos de impactos de artilharia, obtidos a partir de fontes *open source*, totalmente independentes dos *datasets* utilizados anteriormente, contendo maior diversidade de condições: movimento de câmara, presença de elementos móveis, diferentes resoluções e contextos visuais. A



FIGURA 5.8: Exemplo de TP. Método baseado em *Background Subtraction*

Tabela 5.5 apresenta a caracterização deste conjunto de vídeos adicionais, incluindo duração, número de impactos registados e resolução.

TABELA 5.5: Caracterização dos vídeos de fontes *open source* utilizados na segunda fase de testes.

Vídeo	Impactos	Resolução	Frame rate
Vídeo 1	1	1280×720	28 fps
Vídeo 2	2	1280×720	28 fps
Vídeo 3	1	1280×720	28 fps
Vídeo 4	1	1280×720	28 fps
Vídeo 5	2	1280×720	28 fps
Vídeo 6	1	1280×720	28 fps

Neste conjunto, o método baseado em *Background Subtraction* com GMM revelou-se extremamente sensível ao ruído visual. Em todos os sete impactos analisados, os conjuntos de *frames* próximos ao impacto apresentaram dezenas a centenas de falsas deteções. Nenhum dos impactos foi identificado com menos de 10 falsas deteções. Os movimentos do UAV, bem como o movimento de viaturas e outros elementos no cenário, foram interpretados como eventos de interesse, o que resultou numa quantidade excessiva de falsos positivos. A Figura 5.9 ilustra o padrão de deteção obtido com este método.



FIGURA 5.9: Exemplo de resultados com o método baseado em GMM: múltiplas falsas detecções e ruído em cenário com movimento.

Por outro lado, o método com base no *YOLOv8*, utilizando um limiar de confiança mínimo de 0,35, produziu duas detecções, das quais apenas uma correspondeu a um impacto real. A outra detecção foi um falso positivo. Assim, apenas um dos sete impactos foi corretamente detetado. A repetição dos testes com limiares de confiança inferiores (0,25 e 0,10) não resultou em melhorias: o número de detecções manteve-se idêntico. A Figura 5.10 mostra um exemplo das detecções produzidas por este método.



FIGURA 5.10: Exemplo de resultados com o método *YOLOv8*: detecção verdadeira e falso positivo no mesmo *frame*.

Este comportamento revela limitações em ambos os métodos quando aplicados a cenários com elevado dinamismo. O GMM apresenta uma elevada taxa de falsas detecções em presença de movimento no fundo ou da câmara, tornando-se

pouco fiável em operações reais. O *YOLOv8*, embora mais robusto a esse tipo de ruído, mostrou-se insuficientemente sensível para detetar os impactos em contextos distintos daqueles usados durante o treino, o que reflete uma fraca capacidade de generalização.

Resumindo, esta segunda fase de testes evidenciou as fragilidades de ambos os métodos quando expostos a condições visuais mais desafiantes. A deteção automática de impactos em ambientes reais requer soluções mais robustas.

5.4 Discussão dos Resultados

A análise dos resultados obtidos permite identificar com clareza as vantagens e limitações de cada abordagem testada no contexto da deteção automática de impactos de artilharia.

No teste realizado com o vídeo gravado no CMSM, que apresenta um cenário visualmente estável, sem movimento da câmara e sem elementos móveis, o método com base no modelo *YOLOv8* detetou corretamente cinco dos sete impactos, com apenas um falso positivo. Este desempenho foi razoável tendo em conta a natureza abrupta dos impactos e a ausência de análise temporal no modelo. Contudo, ao aplicar o mesmo método a uma compilação de vídeos de fontes abertas, que incluía sete impactos registados com movimento significativo da câmara, variações de iluminação, presença de viaturas e maior diversidade de cenários, o desempenho decaiu drasticamente: apenas um impacto foi detetado corretamente e foi registado um falso positivo.

Estes resultados evidenciam uma fraca capacidade de generalização do modelo *YOLOv8* quando exposto a condições diferentes daquelas presentes nos dados de treino. A principal razão para esta limitação está provavelmente relacionada com a reduzida dimensão e diversidade do conjunto de dados usado no treino. Apesar das técnicas de aumento de dados aplicadas, o modelo foi treinado com imagens predominantemente extraídas de vídeos dos exercícios, que foram captados em cenário controlado e com a câmara estática, o que limita a sua robustez em ambientes mais variados.

Além disso, os modelos da família YOLO analisam cada *frame* de forma isolada, sem memória temporal. Essa abordagem impede o modelo de explorar a continuidade entre imagens consecutivas.

Apesar das suas limitações, o método *YOLOv8* demonstrou ser mais robusto em ambientes com elevado ruído visual e movimento da plataforma. Mesmo nos vídeos mais exigentes, manteve o número de falsas deteções reduzido, o que é relevante para aplicações reais onde o UAV pode estar em movimento para evitar ser abatido pelo inimigo e o cenário contém atividade constante. Esta resistência ao ruído é uma vantagem clara face a métodos puramente baseados em variação de píxeis.

Por outro lado, o método baseado em *Background Subtraction* com GMM apresentou um desempenho notável no vídeo de teste no CMSM. Todos os sete impactos foram corretamente detetados, sem qualquer falso positivo. Isto confirma a eficácia deste tipo de abordagem em ambientes onde as alterações visuais estão diretamente relacionadas com a ocorrência de um impacto.

No entanto, ao aplicar o mesmo método à compilação de vídeos mais exigente, os resultados foram drasticamente diferentes. O movimento da câmara, a deslocação do UAV e a presença de viaturas introduziram variações constantes nos píxeis da imagem. Como resultado, todos os impactos foram acompanhados por dezenas a centenas de falsas deteções, comprometendo por completo a utilidade do algoritmo. Esta elevada sensibilidade ao movimento global da imagem torna o método impraticável em contextos operacionais reais.

5.5 Conclusão

Concluindo, o método de *Background Subtraction* revelou-se extremamente eficaz em vídeos estáticos, mas altamente vulnerável ao movimento da câmara e a alterações visuais não relacionadas com impactos. Já o método *YOLOv8*, apesar de menos sensível a eventos muito breves, demonstrou maior fiabilidade quando sujeito a variações típicas de operações com UAV. A escolha do método adequado dependerá, portanto, das condições de operação esperadas.

Capítulo 6

Conclusões

À luz dos resultados, consolida-se que o MOG2 é particularmente eficaz quando o fundo é estável e a câmara permanece fixa, enquanto o YOLOv8 se revela mais robusto face a ruído visual, movimento da plataforma e variação do cenário, embora a sua generalização dependa da dimensão e diversidade do dataset. Em ambos os casos, a associação temporal assegurada pelo ByteTrack é condição necessária para evitar duplicações e garantir coerência na contagem ao longo do tempo. Com base neste quadro, a secção seguinte apresenta as conclusões do estudo e, de seguida, discute limitações, implicações operacionais e prioridades para trabalho futuro.

6.1 Considerações Finais

Este trabalho teve como objetivo explorar soluções para a deteção automática de impactos de artilharia recorrendo a imagens captadas por UAV. Para tal, foram definidos três objetivos principais: o estudo e aplicação de técnicas de visão computacional clássica, o estudo e implementação de métodos baseados em aprendizagem profunda, e a aquisição de um conjunto de dados com imagens reais de impactos, anotado manualmente para treino e avaliação dos algoritmos propostos.

Através da implementação e avaliação de dois métodos distintos, um baseado em *Background Subtraction* com GMM e outro com recurso ao modelo *YOLOv8*, foi possível identificar as suas capacidades e limitações em diferentes condições operacionais. A subtração de fundo demonstrou grande eficácia em ambientes controlados com câmara estática, revelou-se extremamente vulnerável ao movimento da plataforma e ao ruído visual. Por outro lado, o modelo *YOLOv8*, apesar de menos sensível e preciso nas condições iniciais, apresentou uma maior robustez em cenários mais complexos e com movimento, aproximando-se das exigências reais de utilização em operações com UAV.

Além disso, o trabalho contribuiu com a criação de um *dataset* anotado especificamente para este fim, com base em vídeos reais e de fontes abertas, e com a implementação de algoritmos de *tracking* que complementam os detetores utilizados.

Em suma, os resultados obtidos reforçam o potencial das técnicas de visão computacional para apoiar a observação avançada e a correção de tiro de morteiros, ao mesmo tempo que sublinham a necessidade de melhorar a robustez e a fiabilidade dos sistemas desenvolvidos para que possam ser utilizados de forma eficaz e autónoma em contexto real.

6.2 Limitações e Trabalho Futuro

Com base nas limitações identificadas ao longo deste trabalho, delinearam-se várias linhas de ação que poderão contribuir para melhorar substancialmente a robustez, fiabilidade e aplicabilidade do sistema de deteção automática de impactos.

Uma das principais prioridades consiste no alargamento, em quantidade e diversidade, do conjunto de dados utilizado para treino. A fraca capacidade de generalização demonstrada pelo modelo *YOLOv8* nos vídeos provenientes de fontes abertas está, muito provavelmente, relacionada com a reduzida quantidade e diversidade dos dados utilizados. Para mitigar esta limitação, será essencial recolher e anotar novas imagens, tanto a partir de fontes *open source* associadas a conflitos armados atuais, como através da captação de vídeos em exercícios militares ou instruções operacionais conduzidas em território nacional. Esta diversificação permitirá ao modelo aprender a reconhecer impactos sob diferentes condições visuais, contextos ambientais e movimentos de câmara.

Importa notar que este trabalho operou unicamente no espectro visível, por limitações de *hardware*. Em trabalhos futuros, faz sentido alargar a recolha e criação de *datasets* com sensores de infravermelho e termal, dado que os impactos libertam energia térmica de curta duração e geram plumas quentes, o que pode facilitar a separação do fundo e a deteção.

Outro fator relevante para a evolução do sistema será a adaptação método baseado em *YOLOv8*, com integração de uma componente de análise temporal. Dado que este tipo de detetor opera *frame-a-frame*, a ausência de contexto temporal limita a sua capacidade de detetar eventos súbitos. A introdução de mecanismos que analisem sequências de *frames*, permitindo ao sistema detetar padrões de variação súbita ao longo do tempo, em vez de depender exclusivamente da aparência isolada de cada imagem, reduzindo simultaneamente o risco de falsos negativos.

Paralelamente, para o método baseado em técnicas de visão computacional clássicas, será vantajoso explorar técnicas de cálculo de fluxo ótico com dois objetivos complementares: estabilizar a imagem e distinguir alterações causadas pelo movimento da plataforma daquelas que resultam de mudanças reais no cenário. Esta distinção será particularmente útil para reduzir falsas detecções geradas por movimento do UAV. Ao quantificar o movimento global da câmara, o sistema poderá filtrar regiões cuja variação se explique apenas por deslocação da plataforma, concentrando a análise nas alterações relevantes.

Adicionalmente, propõe-se fundir ambas as abordagens, com a estabilização por cálculo do fluxo ótico a gerar propostas espaço-temporais dos impactos e o detetor *single-stage* a validá-las (com o ByteTrack a consolidar no tempo).

Neste trabalho, a filtragem espacial, de frequência e temporal não foi explorada de forma sistemática nem constituiu o foco da investigação. Como trabalho futuro, propõe-se avaliar o contributo destas técnicas para a capacidade de deteção e quantificar o impacto nas principais métricas de desempenho.

Por fim, uma próxima etapa essencial para aumentar a utilidade prática do sistema será a georreferenciação dos impactos detetados. Com base nas coordenadas e altitude do UAV, na orientação da câmara e nas características da deteção no plano da imagem, será possível estimar a localização do impacto em relação a um ponto de referência previamente conhecido, como o alvo. Esta etapa deverá incluir o cálculo da distância e do azimute do impacto ao alvo, permitindo apoiar diretamente operações de correção de tiro e avaliação da eficácia dos disparos. A automatização desta capacidade tem o potencial de reduzir significativamente o tempo entre a deteção visual do impacto e a transmissão da correção, encurtando o ciclo de decisão e aumentando a eficácia do apoio de fogo.

Bibliografia

- Alexandrov, D., Pertseva, E., Berman, I., Pantiukhin, I., & Kapitonov, A. (2019). Analysis of Machine Learning Methods for Wildfire Security Monitoring with an Unmanned Aerial Vehicles. *Proceedings of the 24th Conference of Open Innovations Association (FRUCT)*, 3–9.
- Army, U. S. (2021). *Dedicated Aerial Forward Observers: A Shift in Allocating UAS* (rel. téc.). <https://api.army.mil/e2/c/downloads/2023/01/31/4e594bd6/shift-in-allocating-uas-can-increase-the-sensor-to-shooter-lethality-oct-21-public.pdf>
- Army, U. S. (2024). The Future of Strategic Fires Target Acquisition. *Field Artillery E-Edition*. <https://www.lineofdeparture.army.mil/Journals/Field-Artillery/Field-Artillery-2024-E-Edition/Future-of-Fires>
- Bakheet, S., Al-Hamadi, A., Michaelis, B., & Sayed, U. (2012). Human action recognition via affine moment invariants. *Proceedings - International Conference on Pattern Recognition*, 218–221.
- Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.
- Brownlee, J. (2019). *Deep Learning for Computer Vision*. Machine Learning Mastery.
- Casas, E., Ramos, L., Bendek, E., & Rivas-Echeverria, F. (2024). YOLOv5 vs. YOLOv8: Performance Benchmarking in Wildfire and Smoke Detection Scenarios. *Journal of Image and Graphics*, 12(2), 127–136.
- Centre, A. A. R. (2023). *Drones in Modern Warfare* (rel. téc.). <https://researchcentre.army.gov.au/library/occasional-papers/drones-modern-warfare>
- Chen, Y., Zhang, T., & Wang, H. (2024). Computer vision for wildfire detection: a critical brief review. *Multimedia Tools and Applications*, 83, 83427–83470.
- Cong, X., Chen, F., Liu, C., & Meng, Y. (2023). A Review of YOLO Object Detection Algorithms based on Deep Learning. *Frontiers in Computing and Intelligent Systems*, 4, 17–20. <https://doi.org/10.54097/fcis.v4i2.9730>
- Crifasi, J. R. (2025, julho 17). *Targeting at Machine Speed: The Capabilities—and Limits—of Artificial Intelligence*. Modern War Institute at West Point. Obtido agosto 26, 2025, de <https://mwi.westpoint.edu/targeting-at-machine-speed-the-capabilities-and-limits-of-artificial-intelligence/>

- Dalal, N., & Triggs, B. (2005). Histograms of Oriented Gradients for Human Detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 886–893.
- Department of the Army. (1991). *Field Manual 6-30: Tactics, Techniques, and Procedures for Observed Fire*. Washington, DC.
- DJI. (2023). DJI Air 3 Specifications [Acedido em junho de 2025]. <https://www.dji.com/air-3/specs>
- Eisenman, A., Devanur, N., Fu, Z., Horvath, A., Li, X., Liu, S., Ma, Y., Netravali, R., Pirk, H., Suresh, A., Xie, Y., Zhang, L., & Zhu, Y. (2022). Checkpointing for Deep Learning Training. *19th USENIX Symposium on Networked Systems Design and Implementation (NSDI 22)*, 323–339. <https://www.usenix.org/system/files/nsdi22-paper-eisenman.pdf>
- Escola de Fuzileiros. (s.d.). *Manual de Pelotão de Morteiros*. Marinha Portuguesa. Vale de Zebro, Portugal.
- Exército Português. (2004). *MC 20-100: Manual de Tática de Artilharia de Campanha* [Capítulo 1, Secção 2: O Apoio de Fogos e a Manobra]. Estado-Maior do Exército. Lisboa.
- for European Policy Analysis, C. (2024). *Adaptation Under Fire: Russia’s Kill Chain In Ukraine* (rel. téc.). <https://cepa.org/comprehensive-reports/adaptation-under-fire-mass-speed-and-accuracy-transform-russias-kill-chain-in-ukraine>
- Ghali, R., Akhloufi, M., Jmal, M., Mseddi, W., & Attia, R. (2021). Forest Fires Segmentation using Deep Convolutional Neural Networks. *Unpublished manuscript*.
- Ghali, R., Akhloufi, M., & Mseddi, W. (2022). Deep Learning and Transformer Approaches for UAV-Based Wildfire Detection and Segmentation. *Sensors*, 22(5), 1977.
- Gil, C. M. F. (1998, novembro). *Folheto de Tática de Morteiros* [21090 2 Ten FZ]. Corpo de Fuzileiros, Companhia de Apoio de Fogos.
- Gonzalez, R. C., & Woods, R. E. (2008). *Digital Image Processing* (3rd). Prentice Hall.
- Goodfellow, I., Bengio, Y., & Courville, A. (2016a). *Deep Learning* [<http://www.deeplearningbook.org>]. MIT Press.
- Goodfellow, I., Bengio, Y., & Courville, A. (2016b). *Deep Learning*. MIT Press. <https://www.deeplearningbook.org/>
- Guest, T. (2023). Forward Observers – Highly Skilled, Highly Trained [Disponível em: <https://euro-sd.com/2023/10/articles/34775/forward-observers->

- highly-skilled-highly-trained/ (acesso em 30 Out 2023)]. *European Security & Defence*.
- Hassan, S., Mujtaba, G., Rajput, A., & Fatima, N. (2024). Multi-object tracking: a systematic literature review. *Multimedia Tools and Applications*, 83(14), 43439–43492.
- Jiao, Z., Zhang, Y., Mu, L., Xin, J., Jiao, S., Liu, H., & Liu, D. (2020). A YOLOv3-based Learning Strategy for Real-time UAV-based Forest Fire Detection. *Proceedings of the Chinese Control and Decision Conference (CCDC)*, 4963–4967.
- Jiao, Z., Zhang, Y., Xin, J., Mu, L., Yi, Y., Liu, H., & Liu, D. (2019). A Deep Learning Based Forest Fire Detection Approach Using UAV and YOLOv3. *Proceedings of the 1st International Conference on Industrial Artificial Intelligence (IAI)*, 1–5.
- Khan, A., Rahmani, A., Shah, M., & Bennamoun, M. (2018). A Guide to Convolutional Neural Networks for Computer Vision. *Synthesis Lectures on Computer Vision*, 8(1), 1–207.
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. *Advances in Neural Information Processing Systems (NeurIPS)*, 1097–1105.
- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep Learning. *Nature*, 521(7553), 436–444.
- Lian, X., Chen, D., & Yan, X. (2024). Burst Point of Projectile Instance Segmentation Method Based on Improved SOLOv2 [20.35 FPS; +3.21 precision improvement (abstract)]. *Proc. SPIE 13105, 13th International Conference on Computer Graphics, Imaging and Visualization*, 131053M. <https://doi.org/10.1117/12.3026726>
- Lowe, D. G. (2004). Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2), 91–110.
- Marinha Portuguesa. (2022, março 22). Diretiva Estratégica da Marinha 2022 [Iniciativa Estratégica 40: conceito de emprego das forças de fuzileiros].
- McLachlan, G. J., & Peel, D. (2000). *Finite Mixture Models*. Wiley.
- Mohsan, S. A. H., Khan, M. A., Noor, F., Ullah, I., & Alsharif, M. H. (2022). Towards the Unmanned Aerial Vehicles (UAVs): A Comprehensive Review. *Drones*, 6(6), 147. <https://doi.org/10.3390/drones6060147>
- NATO. (2020). *AArtyP-01 EDD V1 RD1 E – Allied Procedures for Land-Based Fire Support* [Standardization Agreement (STANAG)]. NATO Standardization Office.

- NATO Standardization Office. (2015). *A ArtyP-5 NATO Fire Support Doctrine (STANAG 2484)* [Edition B, Version 1].
- North Atlantic Treaty Organization. (2023). *AAP-6: NATO glossary of terms and definitions* [Accessed August 7, 2025]. NATO Standardization Office. <https://nso.nato.int/nso/zPublic/ap/AAP-6.pdf>
- OpenCV Development Team. (2025). Morphological Transformations [Consultado em 2025-06-21].
- Pereira, R. G. (2024). *Evolução histórica da observação avançada no âmbito dos procedimentos de tiro de Artilharia de Campanha* [Dissertação de Mestrado Integrado]. Academia Militar.
- Pinto, A. C. A. B. (2022). *Deteção de alvos terrestres a partir de um UAV, em apoio a Operações Anfíbias* [tese de mestrado, Escola Naval] [Dissertação de Mestrado em Ciências Militares Navais (Especialidade de Fuzileiro)].
- Prince, S. J. (2012). *Computer Vision: Models, Learning, and Inference*. Cambridge University Press.
- Qin, X., & Yuan, G. (2013). Detection of Artillery Blast Point Based on Principal Component Pursuit and Analysis [Chinese; method: PCP for foreground recovery + PCA on trajectories]. *Acta Armamentarii* (), 34(10), 1266–1272.
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 779–788.
- Reuters. (2024). How drone combat in Ukraine is changing warfare [“In Ukraine, drones ... guiding artillery and destroying enemy targets.”]. *Reuters Graphics*. <https://www.reuters.com/graphics/UKRAINE-CRISIS/DRONES/dwpkeyjwkpmp/>
- Sarudalf3. (s.d.). YOLO - You Only Look Once [Acedido em 4 de junho de 2025].
- Saydirasulov, N., Mukhiddinov, M., Djuraev, O., Abdusalomov, A., & Cho, Y.-I. (2023). An Improved Wildfire Smoke Detection Based on YOLOv8 and UAV Images. *Sensors*, 23(20), 8374.
- Sims, C. S., Davis, M. A., & Hauser, C. E. (2021). Dedicated Aerial Forward Observers: How a fundamental shift in allocating UAS can increase the sensor-to-shooter lethality. *Fires Bulletin*. <https://api.army.mil/e2/c/downloads/2023/01/31/4e594bd6/shift-in-allocating-uas-can-increase-the-sensor-to-shooter-lethality-oct-21-public.pdf>
- Srinivas, S., & Babu, R. V. (2016). Learning Image Representations Using Convolutional Neural Networks for Classification and Detection Tasks. *Frontiers in Robotics and AI*, 3, 28.
- Szeliski, R. (2010). *Computer Vision: Algorithms and Applications*. Springer.

- Tan, P.-N., Steinbach, M., & Kumar, V. (2006). *Introduction to Data Mining*. Addison-Wesley.
- Tang, Z., Liu, X., Chen, H., Hupy, J., & Yang, B. (2020). Deep Learning Based Wildfire Event Object Detection from 4K Aerial Images Acquired by UAS. *AI*, 1(1), 10. <https://doi.org/10.3390/ai1010010>
- Ultralytics. (2023a). YOLOv8 Models [Acedido em abril de 2025].
- Ultralytics. (2023b). YOLOv8: The latest in the YOLO series (Docs) [Acessado em 28/04/2025].
- U.S. Department of the Army. (2024). *Field Manual 3-09: Fire Support and Field Artillery Operations*. https://armypubs.army.mil/epubs/DR_pubs/DR_a/ARN41659-FM_3-09-000-WEB-2.pdf
- Voulodimos, A., Doulamis, N., Doulamis, A., & Protopapadakis, E. (2018). Deep Learning for Computer Vision: A Brief Review. *Computational Intelligence and Neuroscience*, 2018, 1–13.
- Walker, J. (2009). *Bracketing the Enemy: Forward Observers and Combined Arms Effectiveness during the Second World War* [tese de doutoramento, Kent State University]. https://etd.ohiolink.edu/!etd.send_file?accession=kent1248041184&disposition=inline
- Wang, L., Zhang, H., Zhang, Y., Hu, K., & An, K. (2023). A Deep Learning-Based Experiment on Forest Wildfire Detection in Machine Vision Course. *IEEE Access*, 11, 3262701.
- Yang, Z., Shao, Y., Wei, Y., & Li, J. (2024). Precision-Boosted Forest Fire Target Detection via Enhanced YOLOv8 Model. *Applied Sciences*, 14(6), 2413.
- Yao, G., Zhu, S., Zhang, L., & Qi, M. (2024). HP-YOLOv8: High-Precision Small Object Detection Algorithm for Remote Sensing Images. *Sensors*, 24(15). <https://doi.org/10.3390/s24154858>
- Yaseen, M. (2024). What is YOLOv8: An In-Depth Exploration of the Internal Features of the Next-Generation Object Detector. *arXiv preprint arXiv:2408.15857*.
- Zhang, Y., Sun, P., Jiang, Y., Yu, D., Weng, F., Yuan, Z., Luo, P., Liu, W., & Wang, X. (2022). Bytetrack: Multi-object tracking by associating every detection box. *European conference on computer vision*, 1–21.
- Zhang, Y., Sun, P., Jiang, Y., Yu, D., Yuan, Z., Luo, P., Liu, W., & Wang, X. (2021). ByteTrack: Multi-Object Tracking by Associating Every Detection Box. *CoRR*, abs/2110.06864. <https://arxiv.org/abs/2110.06864>

Zivkovic, Z. (2004). Improved adaptive Gaussian mixture model for background subtraction. *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, 2, 28–31. <https://doi.org/10.1109/ICPR.2004.1333992>