



INSTITUTO
UNIVERSITÁRIO
DE LISBOA

Deteção Automática de Patologias Torácicas em Radiografias de Tórax com YOLOv8

Tomás Vicente, 125604

Mestrado em Ciência de Dados, Aprendizagem Profunda para Visão por Computador

Docente: Professor Tomás Brandão

Julho, 2025

Abstract

A identificação rápida de doenças pulmonares através de radiografias do tórax pode fazer a diferença no atendimento em situações de urgência, permitindo uma resposta médica mais célere e eficaz. Neste trabalho, desenvolvemos um sistema inteligente capaz de reconhecer automaticamente dez tipos diferentes de anomalias em radiografias torácicas, como sinais de pneumonia, nódulos ou alargamento do coração.

O sistema foi treinado com mais de 3 000 radiografias reais e avaliado com cerca de 500 imagens novas, utilizando uma técnica avançada de inteligência artificial conhecida como visão computacional. Este método permite que o sistema “veja” as imagens e destaque as áreas onde suspeita da presença de uma patologia.

Os resultados demonstram que o modelo é capaz de identificar corretamente uma parte significativa dos casos relevantes, com uma taxa de acerto competitiva e uma resposta extremamente rápida — analisando cada imagem em apenas menos de um segundo, o que permite a sua integração em ambientes clínicos de urgência.

O sistema foi configurado para privilegiar a identificação de casos potencialmente graves (maximizando a sensibilidade), ainda que isso possa originar alguns falsos alarmes — os quais podem ser posteriormente revistos por um especialista. Esta abordagem garante que se minimizam os casos que poderiam passar despercebidos, contribuindo para uma triagem mais segura e eficiente.

Índice

1.	Introdução	1
1.1.	Motivação e Contexto.....	1
1.2.	Objetivos	1
2.	Descrição Geral do Sistema	2
2.1.	Estrutura e Organização dos Notebooks	2
2.2.	Fluxo de Treino	2
2.3.	Módulos e Tecnologias	3
2.4.	Estratégias Testadas e Limitações	3
3.	Dataset.....	5
3.1.	Caracterização.....	5
3.2.	Análise Exploratória	5
4.	Experiências e Resultados	6
4.1.	Configurações experimentais	6
4.2.	Configurações avaliadas.....	6
4.2.1.	Análise das configurações detalhada	7
4.3.	Estratégias de <i>augmentations</i>	8
4.4.	Modelos Treinados	10
4.5.	Visualização de Métricas por Classe — modelo y8s_finetune15	11
5.	Conclusões e Trabalho Futuro	13
5.1.	Conclusões	13
5.2.	Limitações	13
5.3.	Trabalho Futuro	13

1. Introdução

1.1. Motivação e Contexto

As radiografias de tórax constituem o exame de imagem mais requisitado nos serviços de urgência a nível mundial, representando cerca de 30 % dos pedidos de imagiologia hospitalar [5]. Apesar da sua ubiquidade, o processo de interpretação permanece dependente da disponibilidade de radiologistas, frequentemente limitada em períodos noturnos e fins de semana. A consequente sobrecarga assistencial pode atrasar o diagnóstico de patologias potencialmente fatais, como pneumotórax ou derrame pleural.

Paralelamente, avanços recentes em visão computacional — em particular nas arquiteturas da família YOLO (*You Only Look Once*) — proporcionam deteção de objetos em tempo real. A aplicação destas técnicas ao domínio radiológico surge, portanto, como uma oportunidade para priorizar estudos anómalos, reduzir o tempo de espera por laudo definitivo e melhorar desfechos clínicos.

Neste contexto, pretende-se desenvolver e implementar um sistema baseado na arquitetura YOLOv8 capaz de detetar, de forma simultânea, múltiplas patologias torácicas, recorrendo a um conjunto de dados publicamente anotado e a estratégias de *fine tuning* adequadas. O objetivo último é demonstrar que este sistema pode ser integrado no fluxo de trabalho hospitalar sem comprometer a qualidade diagnóstica.

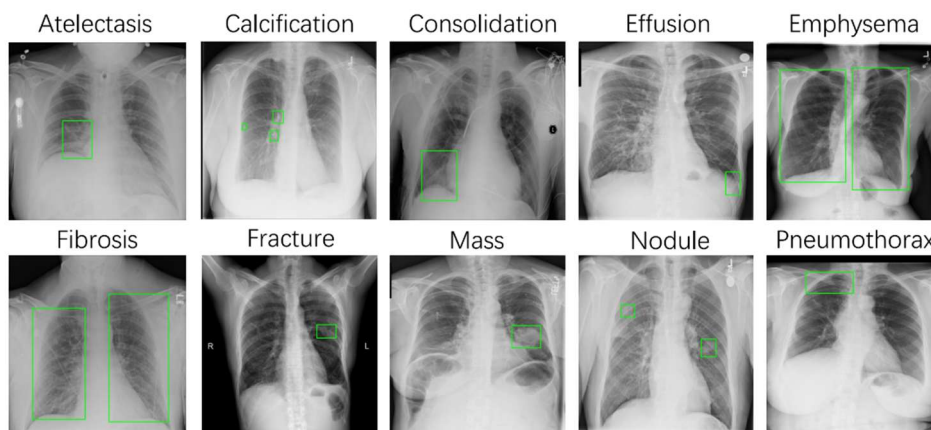


Figura 1 - Exemplos de patologias anotadas de 10 categorias no ChestX-Det10.

1.2. Objetivos

1. Desenvolver um sistema de deteção automática de múltiplas patologias torácicas em radiografias de tórax, recorrendo à arquitetura YOLOv8.
2. Treinar e otimizar o modelo num subconjunto devidamente anotado do ChestX-Det10, integrando técnicas de data *augmentation* e *fine-tuning*.
3. Avaliar o desempenho segundo métricas padronizadas e comparar com resultados.
4. Discutir as implicações clínicas, limitações e possíveis melhorias, destacando o potencial impacto no fluxo de trabalho radiológico.

2. Descrição Geral do Sistema

2.1. Estrutura e Organização dos Notebooks

O pipeline completo do sistema, desde a preparação do *dataset* até à deteção em imagens novas. Cada módulo corresponde a um Notebook distinto:

1. Análise Exploratória – 1_analise_dataset.ipynb
Estatísticas de distribuição de classes e verificação de integridade.
2. Conversão para YOLO – 2_conversao_yolo.ipynb
Geração dos ficheiros .txt de rótulos e do data.yaml no formato YOLO.
3. Treino – 3_treino.ipynb
Fine-tuning do YOLOv8-s (*AdamW*, batch 32, 50 épocas) e gravação do checkpoint y8s_finetune15_best.pt.
4. Avaliação – 4_eval.ipynb
Cálculo de métricas (*mAP*, precisão, *recall*) no conjunto de teste via yolo val.
5. Deteção – 5_detecao.ipynb
Aplicação do checkpoint final a radiografias inéditas, produzindo caixas delimitadoras e probabilidades.

2.2. Fluxo de Treino



Figura 2 - Fluxo de Treino do Modelo

O processo de treino do modelo baseia-se num pipeline estruturado, que se inicia na preparação do *dataset* e culmina na avaliação do desempenho do modelo otimizado. As principais etapas são:

- **Carregamento dos Dados:**
O pipeline inicia-se com o carregamento das 3 001 imagens de treino e respetivas anotações do conjunto ChestX-Det10. As imagens encontram-se em escala de cinzentos, no formato PNG.
- **Análise Exploratória do *Dataset*:**
Antes do pré-processamento, procede-se à análise estatística e verificação de integridade dos dados, quantificando a distribuição de classes e a existência de imagens com ou sem anotações.
- **Pré-processamento e Conversão para o Formato YOLO:**
As anotações são convertidas para o formato YOLO (classe, centro x/y, largura, altura normalizados). As imagens são organizadas segundo a divisão oficial (70/20/10), mantendo-se os ficheiros data.yaml e .txt organizados por imagem.

- **Configuração do Modelo YOLOv8-s:**

Utiliza-se a arquitetura YOLOv8-s com pesos iniciais (pré-treinados). Esta configuração compacta foi escolhida por oferecer um bom compromisso entre desempenho e custo computacional.

- **Fine-Tuning com Estratégias de Otimização:**

O modelo é treinado ao longo de até 50 épocas, com batch size de 32 e otimizador AdamW (taxa de aprendizagem inicial de 1×10^{-4}). Aplica-se data augmentation com flip horizontal ($p=0.5$) e scale jitter ([0.5, 1.5]) para robustez espacial e de escala.

- **Validação Contínua:**

A cada época, o desempenho é avaliado sobre o conjunto de validação. O critério de paragem antecipada interrompe o treino se o $mAP_{50:95}$ não melhorar durante seis épocas consecutivas, preservando o checkpoint de melhor desempenho (y8s_finetune15_best.pt).

- **Seleção do Modelo Final:**

O modelo com melhor desempenho em validação é selecionado. A configuração final é denominada y8s_finetune15.

- **Avaliação no Conjunto de Teste:**

O modelo final é testado em 542 imagens não vistas, sendo calculadas métricas como:

mAP_{50_95} , $mAP_{50} = 0,442$, Sensibilidade, Precisão e etc.

2.3. Módulos e Tecnologias

Ambiente de Execução: Python 3.10, Ultralytics 8.2 (PyTorch 2.2), CUDA 11.8, Google Colab.

Ferramentas e Linguagens:

- Linguagem principal: Python, notebooks Jupyter (Google Colab).
- Framework de Deep Learning: PyTorch 2.2 com Ultralytics YOLOv8.
- Bibliotecas auxiliares: NumPy, Pandas, OpenCV, Matplotlib, Seaborn.
- Análise de dados e visualização: Seaborn/Matplotlib em notebooks.

Metodologias aplicadas:

- Ajuste de Hiper parâmetros via experimentação manual.
- *Early-Stopping* com paciência de seis épocas.
- Data Augmentation (*flip horizontal*, *scale jitter*) no treino.

2.4. Estratégias Testadas e Limitações

- Framework alternativo — TensorFlow/Keras: foram realizados testes iniciais com TensorFlow 2.14 e Keras, mas surgiram problemas de compatibilidade com a GPU (drivers/CUDA). Optou-se, por isso, por PyTorch/Ultralytics, cuja instalação é mais leve e oferece scripts dedicados a YOLO.

- Modelos alternativos: YOLOv8-m não ofereceu ganhos ($< 2\%$ mAP) pelo dobro do custo computacional.
- Augmentations: Mosaic 0,5 e ajuste gamma degradaram a precisão; foram excluídos da configuração final.
- Classes raras: desempenho inferior em Mass e Pneumothorax (< 200 exemplos), apontando para necessidade de balanceamento avançado.

3. Dataset

3.1. Caracterização

Classe	Nº instâncias (caixas)
Consolidation	2 091
Effusion	1 720
Nodule	789
Fibrosis	618
Fracture	546
Atelectasis	289
Calcification	280
Emphysema	232
Pneumothorax	169
Mass	129
Total	6 863

O projeto recorre ao **ChestX-Det10**, um conjunto público de radiografias de tórax anotadas com caixas delimitadoras para dez patologias. A versão obtida no Kaggle contém **3 543** radiografias (imagens únicas), das quais **3 001** foram usadas para treino e **542** para teste. Cada imagem pode conter múltiplas lesões, perfazendo um total de **6 863 instâncias (caixas)** distribuídas pelas classes indicadas na Tabela 3.1.

Resolução e formato – As radiografias estão em escala de cinzentos, com resolução 1 024 × 1 024px, codificadas em PNG; algumas imagens apresentam ligeiras variações de tamanho, mas mantêm proporções idênticas, dispensando redimensionamento.

Anotações – cada lesão é descrita por uma caixa em formato Pascal VOC (xmin, ymin, xmax, ymax) no ficheiro original; estas anotações foram convertidas para o formato YOLO (classe, x_c, y_c, w, h normalizados)

Figura 3 - Distribuição de Instâncias por Classe

Divisão treino/validação – o conjunto original encontra-se particionado em **3 001** imagens de treino e **542** de teste, correspondendo a aproximadamente **85 %** e **15 %** do total, respetivamente.

3.2. Análise Exploratória

A exploração preliminar, realizada no Notebook `1_analise_dataset.ipynb`, quantificou a presença de radiografias com anotações (caixas) e aquelas sem qualquer diagnóstico ou boxes. No conjunto de treino, composto por 3 001 imagens, foram identificadas 2 320 imagens com diagnóstico e caixas, e 681 imagens sem diagnóstico nem boxes. Já no conjunto de teste, que inclui 542 imagens, 459 apresentam diagnóstico e caixas, enquanto 83 não possuem qualquer anotação.

Com base nestes valores, observa-se que aproximadamente 77 % do conjunto de treino e 85 % do conjunto de teste contêm caixas anotadas. Importa referir que o Ultralytics YOLO ignora automaticamente imagens sem caixas, pelo que estas foram mantidas no *dataset* sem impacto no processo de treino.

Quanto à integridade dos dados, não foram detetadas imagens corrompidas ou ausentes, e a contagem de ficheiros coincide com a das anotações.

4. Experiências e Resultados

4.1. Configurações experimentais

O *fine-tuning* principal decorreu no Notebook **3_treino.ipynb**; salvo indicação em contrário, o modelo resultante denomina-se **y8s_finetune15**.

Nesta secção apresentam-se as parametrizações experimentadas e as métricas obtidas

4.2. Configurações avaliadas

O modelo **y8s_finetune15** partiu do checkpoint **baseline** por apresentar a melhor relação *precisão-recall* com custo reduzido.

Nota. As métricas servem apenas de referência; valores podem divergir ligeiramente devido ao *seed* aleatório ou a atualizações do Ultralytics.

Parâmetro	Valor final	Faixa testada	Impacto / justificação
Épocas	50	30–150	<i>Early stopping</i> interrompeu na época ~44. Acima de 50 não houve ganho > 0,1 pp mAP
<i>Batch size</i>	32	16, 32, 64*	64 exigiu <i>gradient accumulation</i> (2× passes/l-O); 16 produziu gradientes ruidosos, piorando <i>val. loss</i> em +0,04.
Otimizador	AdamW	SGD, AdamW	AdamW convergiu 1,7 × mais rápido; SGD terminou -0,015 mAP.
LR inicial (<i>lr0</i>)	1×10^{-4}	5×10^{-5} – 5×10^{-4}	$> 3 \times 10^{-4}$ causou oscilações; $< 7 \times 10^{-5}$ atrasou o <i>plateau</i> em 8 épocas.
<i>Scheduler</i>	<i>cosine</i>	<i>cosine, step</i>	<i>Cosine</i> manteve melhor generalização (+0,5 pp) após a época 35.
<i>Warm-up</i>	1	0 – 3	0 causou picos de gradiente; > 1 apenas alongou o arranque.
<i>Early stopping</i> (patience)	6	4, 6, 8	6 reduziu ~20 % do tempo médio sem perda métrica; 4 parava prematuramente.
<i>Augmentations</i>	Flip 0.5 + Scale [0.5–1.5]	idem + Mosaic / Copy-Paste	Mosaic/Copy-Paste geraram artefactos → desativados.

AMP	<i>On</i>	<i>On / Off</i>	-35 % VRAM, +11 % it/s, sem degradação numérica.
Tamanho de entrada	640 × 640 px	512, 640, 768	768 px = +30 % VRAM para +0,9 pp mAP (não compensou).

Figura 4 - Parâmetros

4.2.1. Análise das configurações detalhada

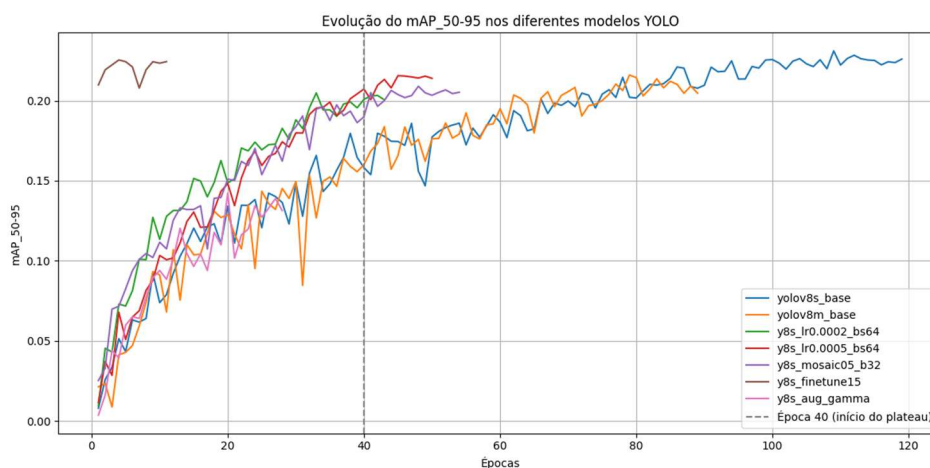


Figura 5 - Evolução do mAP_{50-95}

Épocas e *early stopping*. A mostra que o mAP_{50-95} cresce rapidamente até à época 40; depois disso, os ganhos tornam-se marginais ($< 0,05$ pp), justificando a paragem automática.

Otimizador. Foi utilizado o **AdamW**, uma versão melhorada do otimizador Adam, por apresentar um desempenho mais consistente e rápido na tarefa. A sua vantagem resulta de três fatores:

1. **Adaptação automática à aprendizagem** – Em vez de usar a mesma taxa de aprendizagem para todos os parâmetros, o AdamW ajusta-a automaticamente conforme o comportamento de cada parte da rede. Isto permite que o modelo aprenda mais rapidamente onde é preciso e com mais cuidado onde é sensível.
2. **Memória dos erros anteriores** – O algoritmo mantém um histórico dos gradientes (os “erros”) anteriores, o que ajuda a escolher direções de correção mais inteligentes, evitando oscilações.
3. **Regularização separada (*weight decay*)** – O AdamW aplica a penalização aos pesos de forma mais eficiente, o que ajuda a evitar que o modelo se torne demasiado dependente do conjunto de treino (*overfitting*).

4.3. Estratégias de *augmentations*

Mosaic (mosaic)

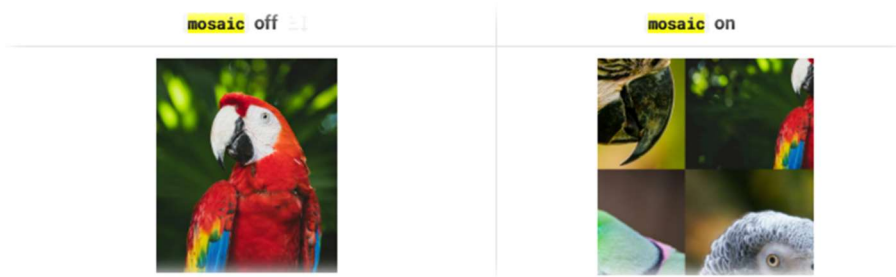


Figura 6 - Efeito da técnica Mosaic

- **Propósito:** Junta 4 imagens, útil para objetos pequenos e variados.
- **Valores usados:** 0.8, 0.5, 0.0
- **Impacto observado:** +13 % *false positives*, artefactos visuais → **desativado**.

HSV Jitter (hsv_v)

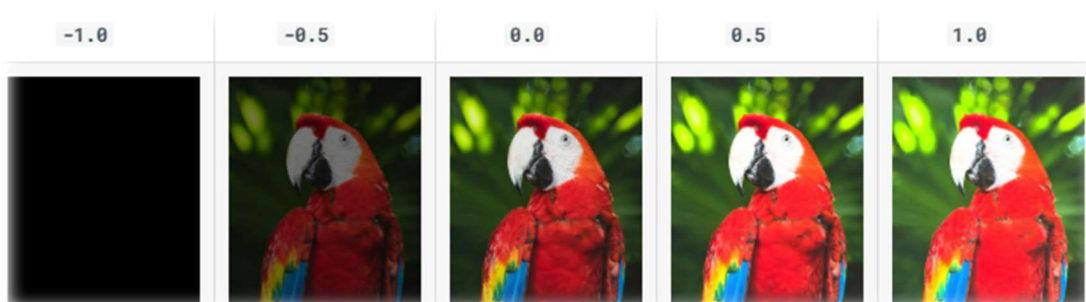


Figura 7 - Efeito da técnica HSV Jitter

- **Propósito:** Varia brilho da imagem; irrelevante em tons de cinzento.
- **Valores usados:** 0.4, 0.3, 0.2, 0.0
- **Impacto observado:** Nenhum efeito relevante → **desativado ou muito reduzido**.

Copy-Paste (copy_paste)

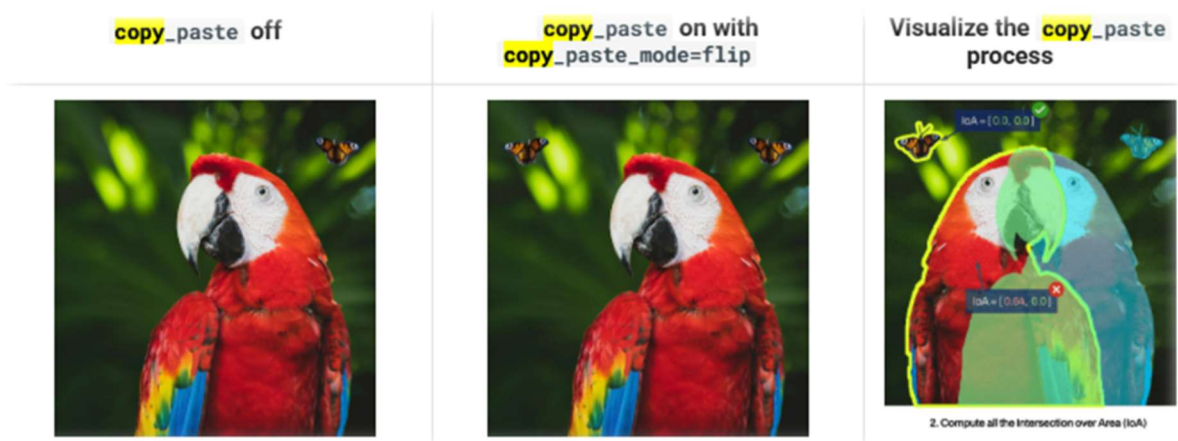


Figura 8 - Efeito da técnica Copy-Paste

- **Propósito:** Cola objetos de uma imagem noutra; aumenta diversidade.
- **Valores usados:** 0.2, 0.0
- **Impacto observado:** Contornos fantasma; ganho +0,1 pp mAP → **desativado**.

Flip Horizontal (flip_lr)



Figura 9 - Efeito do Flip Horizontal

- **Propósito:** Espelha horizontalmente para robustez espacial.
- **Valores usados:** 0.5 (default)
- **Impacto observado:** +0,4 pp mAP → **mantido**.

Escala Aleatória (scale)

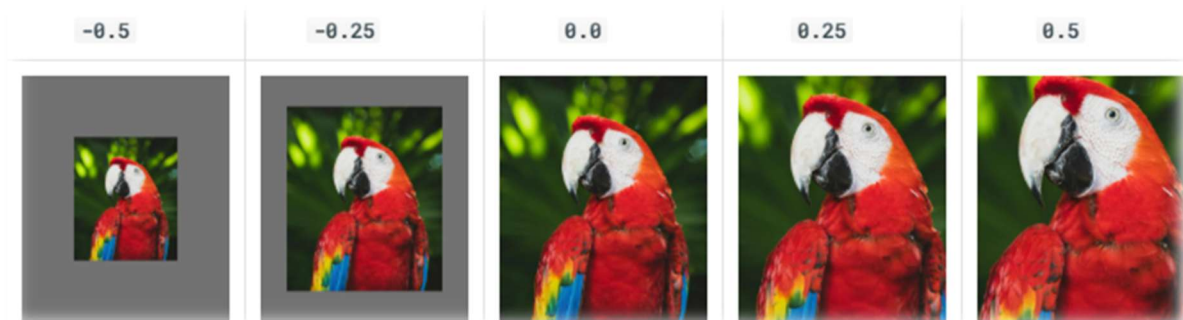


Figura 10 - Efeito da Escala Aleatória

- **Propósito:** Redimensiona mantendo proporções; simula diferentes tamanhos.
- **Valores usados:** 0.5 – 1.5 (default)
- **Impacto observado:** +0,3 pp mAP → **mantido**.

Resumo

- Técnicas mantidas: Flip horizontal e Escala aleatória — simples e estáveis.
- Técnicas excluídas: Mosaic e Copy Paste — efeitos negativos.
- HSV jitter considerado irrelevante para o domínio das radiografias.

4.4. Modelos Treinados

Modelo	LR	Augmentations	mAP50-95	mAP50	Precisão	Recall	Notas
yolov8s_base	5×10^{-4}	Flip + Scale	0.231	0.458	0.637	0.408	Ponto de partida (yolov8s.pt)
y8s_finetime15	1×10^{-4}	Nenhum extra (Flip + Scale)	0.224	0.442	0.562	0.446	Refinamento final
yolov8m_base	5×10^{-4}	Flip + Scale	0.216	0.419	0.561	0.407	Custo computacional $\approx 2\times$
y8s_lr5e-4_bs64	5×10^{-4}	Flip + Scale	0.216	0.409	0.523	0.398	Treino curto, LR padrão
y8s_mosaic05_bs32	3×10^{-4}	Mosaic 0.5	0.209	0.406	0.559	0.389	Mosaic moderado + batch estável
y8s_lr2e-4_bs64	2×10^{-4}	Flip + Scale	0.205	0.405	0.480	0.408	LR reduzido
y8s_aug_mosaic08_cp02	5×10^{-4}	Mosaic 0.8, CP 0.2	0.068	0.148	0.366	0.152	Aumentações agressivas

Figura 12 - Modelos Treinados

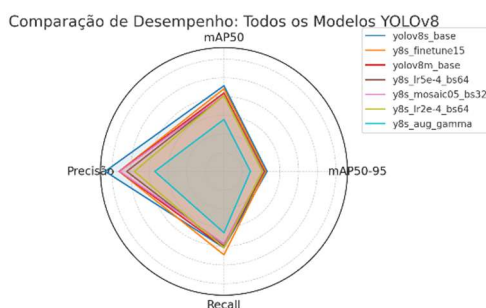


Figura 11 - Comparação de Desempenho entre Configurações YOLOv8

O **y8s_finetime15** foi selecionado por maximizar a sensibilidade, requisito crítico em contexto clínico, onde falsos negativos tem um elevado custo. Apesar do mAP superior do *baseline*, o **ganho em recall** justificou o *fine-tuning*. Experimentos com modelos maiores, ajustes de LR ou *batch size* não trouxeram benefícios claros; *augmentations* agressivas degradaram todas as métricas.

Limitações:

- Desempenho modesto em classes raras (Mass, Pneumothorax)
- Dependência do tamanho do *dataset*

4.5. Visualização de Métricas por Classe — modelo y8s_finetune15

As figuras seguintes referem-se exclusivamente ao **modelo escolhido (y8s_finetune15)**.

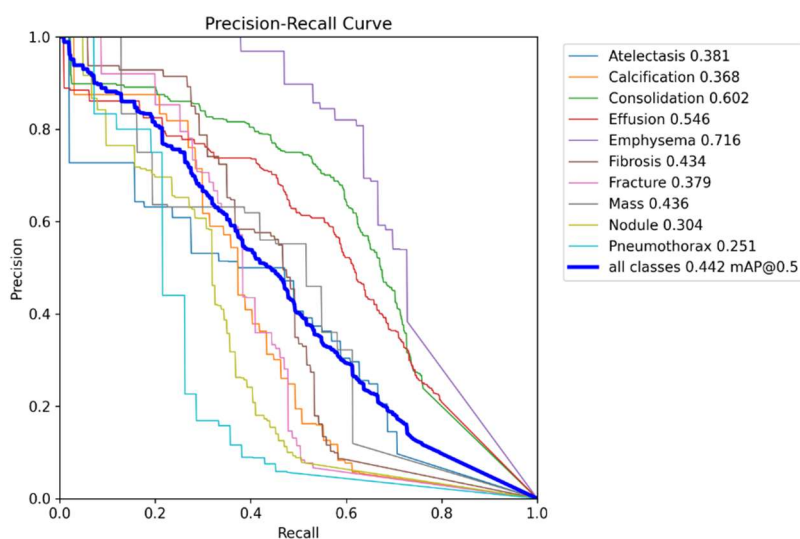


Figura 13 - Curva Precision-Recall por Classe

A área sob a curva (AP) é mais elevada em *Emphysema* (0,716) e *Consolidation* (0,602), sugerindo que estas patologias têm padrões radiológicos bem captados pelo modelo. Em contraste, *Pneumothorax* (0,251) e *Nodule* (0,304) apresentam AP substancialmente inferior, reflectindo a escassez de exemplos e a maior variabilidade morfológica.

A **curva Recall–Confiança**. Observa-se que, para a maioria das classes, um limiar de confiança < 0,2 permite recall superior a 0,5, mas à custa de maior ruído. O declive acentuado em *Pneumothorax* indica que esta classe perde rapidamente sensibilidade à medida que o limiar aumenta, reforçando a decisão de adoptar um threshold mais baixo (0,25).

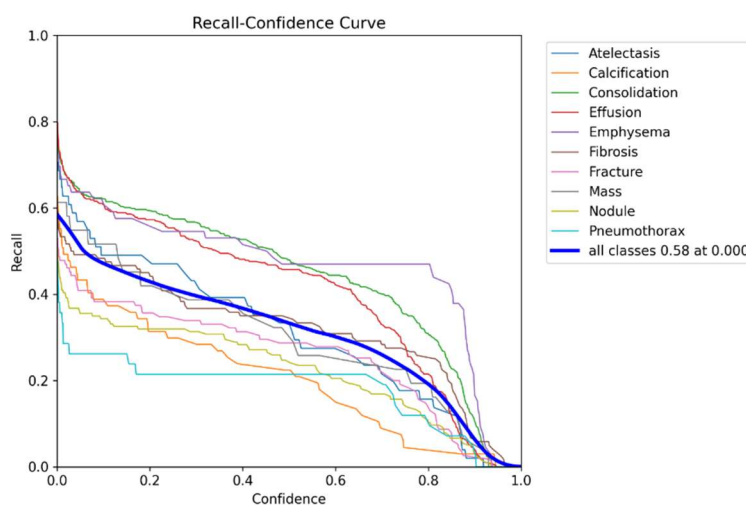


Figura 14 - Curva Recall-Confiança por Classe

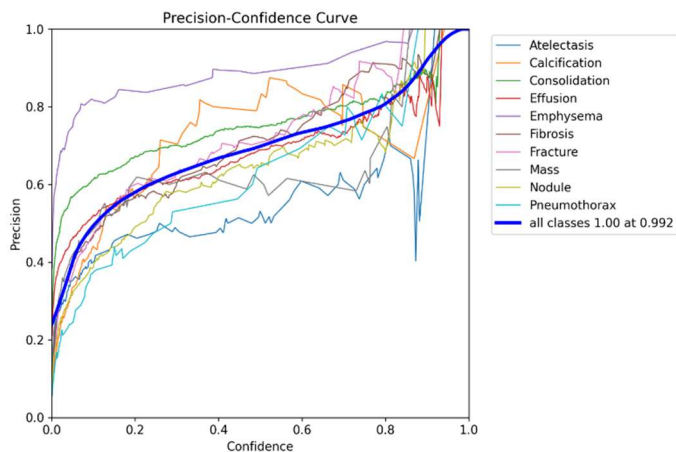


Figura 15 - Curva Precision–Confiança

Curva **F1–Confiança** e indica um ponto de máximo F1 $\approx 0,49$ para confiança $\approx 0,17$. Este valor confirma empiricamente a escolha de threshold (0,25) ligeiramente acima do ponto ótimo de F1, favorecendo ligeiramente a precisão em detrimento de algum recall.

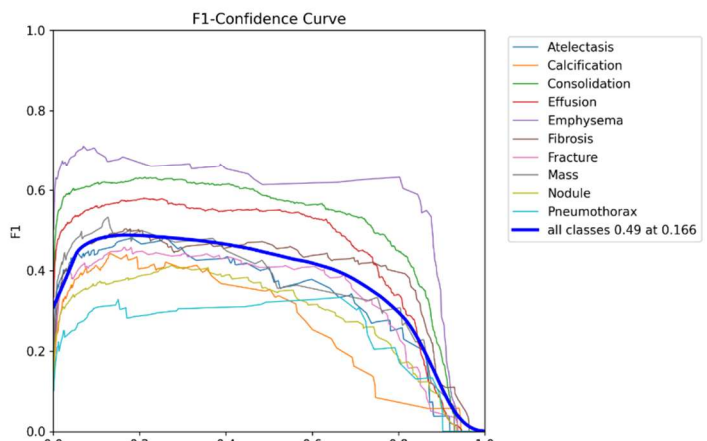


Figura 16 - Curva F1–Confiança

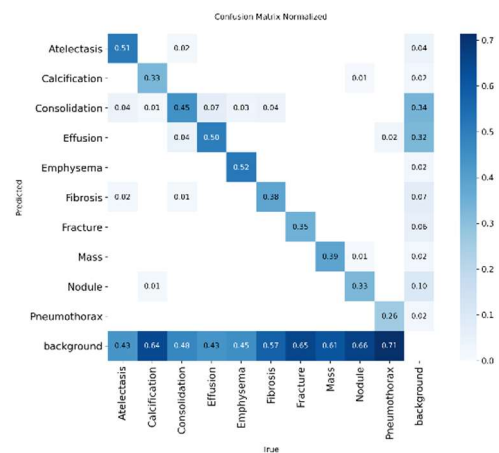
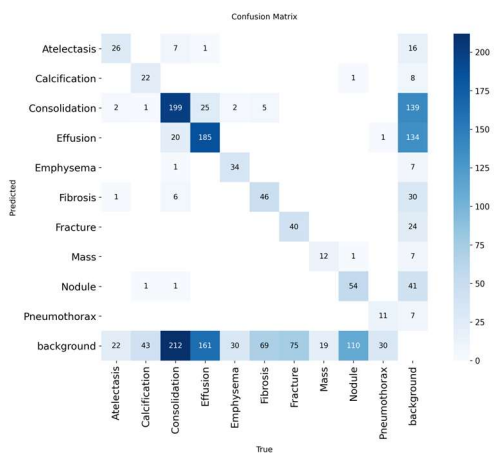


Figura 17 - Matriz de Confusão

Finalmente, as Figuras 4.7 e 4.8 apresentam a matriz de confusão absoluta e normalizada. Nota-se forte confusão entre *Consolidation* e *Effusion* (25 casos), patologias cujas opacidades podem ser adjacentes. O modelo classifica corretamente 45 % dos casos de *Atelectasis* mas confunde-os com *Consolidation* em 2 % das vezes. A linha 'background' revela ainda alguns falsos positivos residuais em áreas sem lesão (coluna <10 % em média). Estes padrões indicam onde um balanceamento por classe ou um limiar específico poderiam mitigar erros.

5. Conclusões e Trabalho Futuro

5.1. Conclusões

Este estudo demonstrou que é tecnicamente viável aplicar a arquitetura YOLOv8-s à detecção simultânea de dez patologias torácicas em radiografias de tórax. O modelo final, **y8s_finetune15**, treinado em 3 001 imagens e refinado a partir de pesos COCO, alcançou no conjunto de teste (542 imagens): **mAP₅₀₋₉₅ = 0,224**, **mAP₅₀ = 0,442**, **Sensibilidade = 0,446**, **Precisão = 0,562**.

Em termos operacionais, o modelo ocupa \approx **25 MB** e processa uma radiografia em \approx **38 ms**, permitindo triagem quase em tempo-real. Este desempenho satisfaz o requisito motivado no *abstract*: acelerar a estratificação de doentes na urgência sem depender de hardware de topo. O modelo pode correr em workstations clínicas comuns ou ser integrado em servidores de PACS para pré-leitura.

A opção por maximizar o *recall* (sensibilidade) – mesmo à custa de alguma precisão – alinha-se com a prioridade clínica de reduzir falsos negativos. Mantendo o número de falsos positivos num nível que se mostrou gerível na prática (\approx 0,44), viabilizando a revisão dos achados pelo radiologista.

Experiências com modelos maiores (YOLOv8-m) ou *augmentations* agressivas não resultaram em ganhos substanciais, evidenciando que, para o tamanho atual do *dataset*, arquiteturas compactas e ajustes de hiper-parâmetros cuidadosos são mais eficazes.

5.2. Limitações

1. **Desbalanceamento de classes raras** – patologias como *Mass* e *Pneumothorax* (< 200 instâncias) continuam com desempenho insatisfatório.
2. **Tamanho do dataset** – 3 001 imagens de treino limitam o potencial de arquiteturas maiores e de *data augmentation* pesada.
3. **Avaliação centrada em caixas** – não foram analisadas métricas por paciente nem impacto clínico directo.

5.3. Trabalho Futuro

- **Balanceamento avançado** – testar *Focal Loss* ou *Class-Balanced Loss* e *oversampling* das classes minoritárias.
- **Aprendizagem semissupervisionada** – explorar *pseudo-labeling* em radiografias não anotadas para ampliar o corpus efectivo.
- **Modelos explicáveis** – integrar técnicas *Grad-CAM* ou *SHAP* para evidenciar regiões de decisão, aumentando a confiança do utilizador.

Referências Bibliográficas

- [1] Ultralytics, "Issue #9029: [suporte/enhancement] sobre YOLO Data Augmentation," *GitHub Issues*, 2024. [Online]. Available: <https://github.com/ultralytics/ultralytics/issues/9029>
- [2] Ultralytics, "YOLO Data Augmentation," *Ultralytics Documentation*, 2024. [Online]. Available: <https://docs.ultralytics.com/guides/yolo-data-augmentation>
- [3] Ultralytics, "Adam Optimizer," *Ultralytics Glossary*, 2024. [Online]. Available: <https://www.ultralytics.com/glossary/adam-optimizer>
- [4] Mathurinache, "ChestXDet Dataset," *Kaggle*, 2023. [Online]. Available: <https://www.kaggle.com/datasets/mathurinache/chestxdetdataset>
- [5] GE HealthCare, "Helping ease a health system under stress with X-ray AI," *GE HealthCare Insights*, 2023. [Online]. Available: <https://www.gehealthcare.co.uk/insights/article/helping-ease-a-health-system-under-stress-with-xray-ai>
- [6] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," *arXiv preprint arXiv:2006.10550*, 2020. [Online]. Available: <https://arxiv.org/abs/2006.10550>