

Detecção de Tuberculose em Radiografias de Tórax

Utilizando Deep Learning e Transfer Learning

Evandro Rocha

Universidade Federal do Rio de Janeiro
UFRJ/COPPE/PEE

10 de dezembro de 2025



Agenda

1. Introdução
2. Problema Abordado
3. Método Proposto
4. Resultados Obtidos
 - ▶ Comparação de Desempenho
 - ▶ Interpretabilidade (Grad-CAM)
5. Conclusões e Recomendação



Introdução



Contexto

Tuberculose: Um Problema Global

- ▶ 10 milhões de casos/ano (OMS, 2022) [9]
- ▶ 1,5 milhões de mortes/ano
- ▶ Principal causa de morte por doença infecciosa
- ▶ Diagnóstico precoce é crucial

Desafios do Diagnóstico

- ▶ Escassez de radiologistas
- ▶ Variabilidade inter-observador
- ▶ Tempo de análise
- ▶ Custo elevado



Motivação

Por que Deep Learning?

- ▶ Capacidade de aprender características complexas
- ▶ Alto desempenho em tarefas de visão computacional
- ▶ Potencial para auxiliar diagnóstico médico [1][3]
- ▶ Redução de tempo e custo

Objetivo do Trabalho

Desenvolver e avaliar modelos de Deep Learning para detecção automática de tuberculose em radiografias de tórax do dataset Shenzhen Hospital.



Problema Abordado



Definição do Problema

Problema de Classificação Binária

Dado uma radiografia de tórax (raio-X), classificar como:

- ▶ **Classe 0:** Normal (sem tuberculose)
- ▶ **Classe 1:** Tuberculose (presença da doença)

Desafios Técnicos:

- ▶ Variabilidade nas imagens
- ▶ Sutiliza dos padrões
- ▶ Dataset limitado (662 imagens)
- ▶ Desbalanceamento de classes

Requisitos:

- ▶ Alta sensibilidade (recall)
- ▶ Alta especificidade
- ▶ Interpretabilidade
- ▶ Eficiência computacional



Dataset Shenzhen Hospital

Características do Dataset:

- **Total:** 662 radiografias
- **Normal:** 326 imagens
- **Tuberculose:** 336 imagens
- **Formato:** PNG, grayscale
- **Resolução:** Variável
- **Fonte:** NIH/NLM

Divisão dos Dados:

Conjunto	%
Treino	70%
Validação	15%
Teste	15%

Balanceamento: 1.03
(quase perfeito)



Método Proposto



Pipeline de Processamento (CNNs Pré-treinadas)

1. Pré-processamento

- ▶ Redimensionamento: 224×224 pixels
- ▶ Normalização ImageNet: $\mu = [0.485, 0.456, 0.406]$,
 $\sigma = [0.229, 0.224, 0.225]$
- ▶ Conversão para RGB (3 canais)

2. Data Augmentation (apenas treino)

- ▶ Horizontal flip ($p=0.5$)
- ▶ Random brightness/contrast ($p=0.3$)
- ▶ Rotation/scale/shift ($p=0.5$, ± 15)

3. Transfer Learning [4]

- ▶ Backbone congelado (feature extraction)
- ▶ Apenas classificador treinado
- ▶ Reduz overfitting em datasets pequenos

Nota

CNN Baseline usa mesmo pré-processamento, mas **sem transfer learning** (treinada do zero)



Transfer Learning: Feature Extraction

Estratégia Adotada

- ▶ **Backbone:** Pesos pré-treinados no ImageNet (congelados)
- ▶ **Classificador:** Treinado do zero para TB vs Normal
- ▶ **Parâmetros treináveis:** 527K–1.05M (ResNet: 1.05M, DenseNet: 527K, EfficientNet: 658K)

Vantagens:

- ▶ Evita overfitting
- ▶ Treina mais rápido
- ▶ Usa menos memória GPU
- ▶ Aproveita features do ImageNet

Treinamento:

- ▶ Otimizador: Adam
- ▶ Learning rate: 10^{-4}
- ▶ Batch size: 16
- ▶ Early stopping: patience=10



Técnicas de Regularização

Prevenção de Overfitting

Múltiplas técnicas aplicadas para generalização com dataset pequeno (662 imagens)

CNNs Pré-treinadas:

- ▶ Dropout: 0.5
- ▶ Batch Normalization
- ▶ Data Augmentation
- ▶ Early Stopping ($p=10$)
- ▶ Weight Decay: 10^{-5}
- ▶ Transfer Learning

CNN Baseline:

- ▶ Dropout: 0.4
- ▶ Batch Normalization
- ▶ Data Augmentation
- ▶ Early Stopping ($p=10$)
- ▶ Weight Decay: 10^{-4}
- ▶ Init: Kaiming (He) [5]

Nota Metodológica

Hiperparâmetros otimizados individualmente para cada abordagem (prática padrão em transfer learning)



CNN Baseline: Arquitetura Tradicional

Arquitetura:

- ▶ CNN treinada do zero
- ▶ 4 blocos convolucionais
- ▶ 1.2M parâmetros
- ▶ Sem pré-treinamento

Objetivo:

- ▶ Servir como baseline
- ▶ Demonstrar valor do transfer learning
- ▶ Comparação justa

Estrutura:

1. Conv2D (32 filtros)
2. Conv2D (64 filtros)
3. Conv2D (128 filtros)
4. Conv2D (256 filtros)
5. Global Average Pooling
6. Classificador (2 classes)

Limitações

Dataset pequeno (662 imagens)
dificulta treinamento do zero



Por que CNN Baseline?

Justificativa Acadêmica

- ▶ **Comparação justa:** Mesma tarefa, mesmos dados
- ▶ **Demonstra valor:** Transfer learning vs treinar do zero
- ▶ **Baseline tradicional:** CNN sem conhecimento prévio
- ▶ **Evidência empírica:** Quantifica ganho do ImageNet

CNN Baseline (do zero):

- ▶ Aprende tudo do zero
- ▶ Precisa de mais épocas
- ▶ Risco de overfitting

Pré-treinadas (ImageNet):

- ▶ Já conhecem features básicas
- ▶ Convergem em menos épocas
- ▶ Menor risco de overfitting



Modelos Avaliados

Modelo	Parâmetros	Pretrained	Características
CNN Baseline	1.2M	—	4 blocos, treinada do zero
ResNet-50 [5]	25.6M	ImageNet	Skip connections, 50 camadas
DenseNet-121 [6]	8.0M	ImageNet	Dense connections
EfficientNet-B0 [7]	5.3M	ImageNet	Compound scaling

Figuras de Mérito

- **Sensibilidade:** Detectar casos de TB
- **Especificidade:** Identificar casos normais
- **AUC-ROC:** Capacidade discriminativa
- **F1-Score:** Balanço geral



Resultados Obtidos



Comparação Completa dos Modelos

Figura de Mérito	CNN Baseline	ResNet50	DenseNet121	EfficientNet-B0	Melhor
Sensibilidade	80.00%	90.00%	82.00%	80.00%	ResNet50
Especificidade	90.00%	92.00%	88.00%	94.00%	EfficientNet
AUC-ROC	90.64%	96.04%	86.32%	89.48%	ResNet50
Acurácia	85.00%	91.00%	85.00%	87.00%	ResNet50
Precisão	88.89%	91.84%	87.23%	93.02%	EfficientNet
F1-Score	84.21%	90.91%	84.54%	86.02%	ResNet50
Parâmetros	1.2M	25.6M	8.0M	5.3M	Baseline
Épocas	33	25	13	16	DenseNet
FN (Falsos Neg.)	10	5	9	10	ResNet50
FP (Falsos Pos.)	5	4	6	3	EfficientNet

Análise:

- **ResNet50**: Melhor performance
- **EfficientNet**: Melhor especificidade
- **DenseNet**: Convergiu em menos épocas
- **Baseline**: Menor complexidade

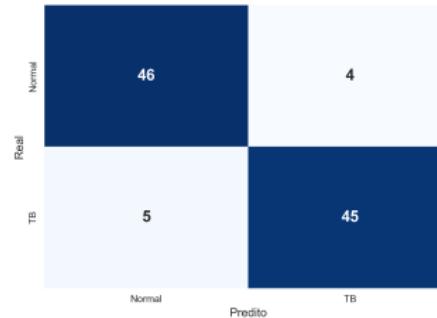
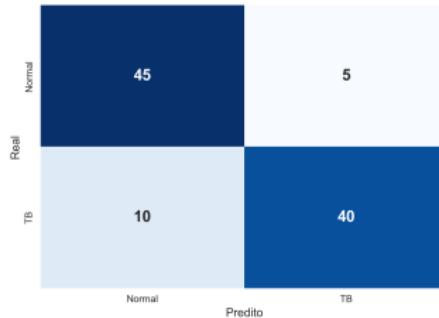
Trade-offs:

- Performance vs Eficiência
- Sensibilidade vs Especificidade

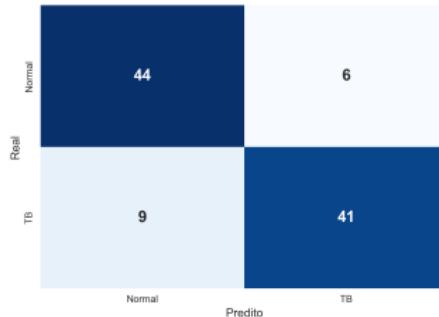


Matrizes de Confusão (Comparação Visual)

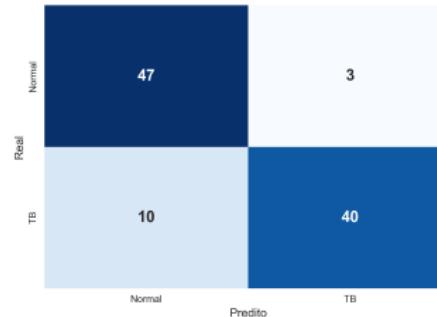
Comparação de Matrizes de Confusão: Classificação de Tuberculose
Simples CNN (Baseline) ResNet-50 (Melhor Performance)



DenseNet-121 (Convergência Rápida)



EfficientNet-B0 (Melhor Especificidade)





Análise Detalhada: Pontos Fortes e Fracos

Modelo	Pontos Fortes	Pontos Fracos
CNN Baseline	<ul style="list-style-type: none">• Simplicidade arquitetural• Serve como referência• Sem dependências externas	<ul style="list-style-type: none">• AUC-ROC: 90.64%• Sensibilidade: 80%• Sem transfer learning
ResNet-50	<ul style="list-style-type: none">✓ Melhor AUC-ROC: 96.04%✓ Melhor sensibilidade: 90%✓ Especificidade: 92%✓ Menos falsos negativos (5)	<ul style="list-style-type: none">• 25.6M parâmetros• 25 épocas• Custo computacional alto
DenseNet-121	<ul style="list-style-type: none">✓ Convergência rápida (13 épocas)✓ Eficiente (8M params)✓ Performance aceitável	<ul style="list-style-type: none">• AUC-ROC: 86.32% (pior)• Sensibilidade: 82%• Mais falsos negativos (9)
EfficientNet-B0	<ul style="list-style-type: none">✓ Mais leve (5.3M params)✓ Melhor especificidade: 94%✓ Alta precisão: 93.02%✓ Menos falsos positivos (3)	<ul style="list-style-type: none">• Sensibilidade: 80%• AUC-ROC: 89.48%• Mais falsos negativos (10)

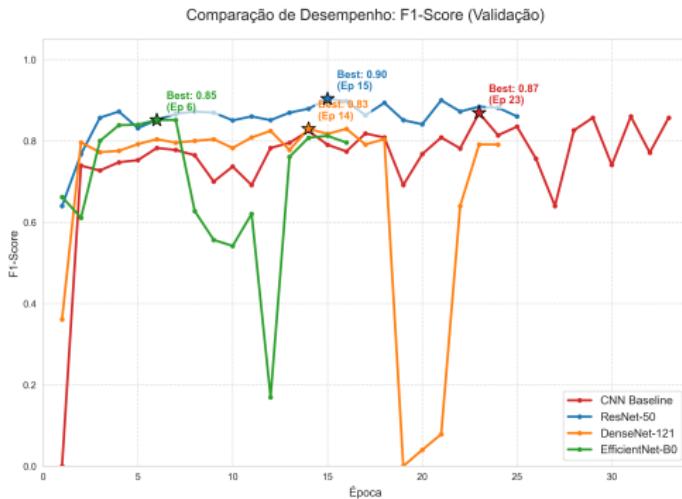
Conclusão

ResNet-50: melhor performance clínica. **EfficientNet-B0:** melhor custo-benefício. **DenseNet-121:** convergência rápida. **CNN Baseline:** Serve como referência.



Evolução do Treinamento (F1-Score)

Interpretação:



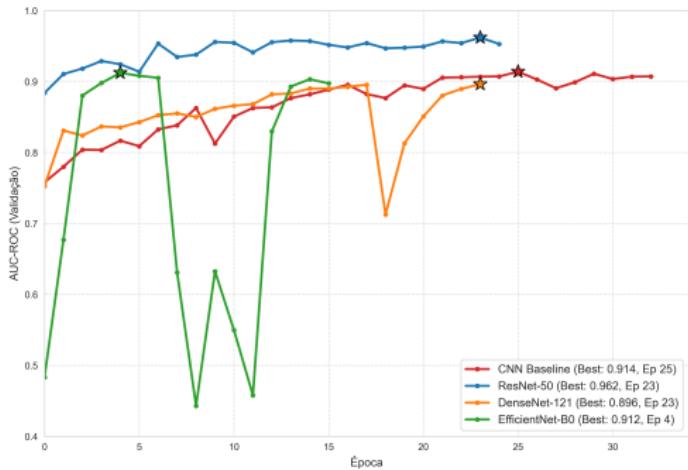
- **Eixo Y:** F1-Score (quanto maior, melhor). Harmonia entre Precisão e Sensibilidade.
- **Convergência:** A DenseNet (laranja) sobe muito rápido, aprendendo cedo.
- **Performance:** A ResNet (azul) atinge o pico mais alto.
- **Baseline:** A CNN Simples (vermelho) tem desempenho inferior e oscila mais.

Evolução do Treinamento (AUC-ROC)

Interpretação:

- **Eixo Y:** Área sob a Curva ROC (Capacidade discriminativa).
- **ResNet-50:** Liderança consistente (> 0.96).
- **DenseNet-121 (Nota):**
 - AUC continuou subindo até a Época 23.
 - O treinamento parou (Early Stop) porque o **F1-Score** (prioridade de equilíbrio) parou de melhorar na Época 13.

Comparação de Desempenho: AUC-ROC por Época





Metodologia de Comparação Visual

O que é Grad-CAM?

Técnica que destaca os pixels da imagem que mais influenciaram a decisão da rede neural (Gradient-weighted Class Activation Mapping) [8].

Cenário de Consenso

Para garantir uma comparação justa entre as arquiteturas, selecionamos uma amostra onde **todos os 4 modelos** classificaram corretamente a Tuberculose (Probabilidade > 50%).

Objetivo: Analisar se redes mais complexas (ex: ResNet, EfficientNet) focam em áreas mais precisas da lesão comparadas à Baseline.



Grad-CAM (CNN Baseline)

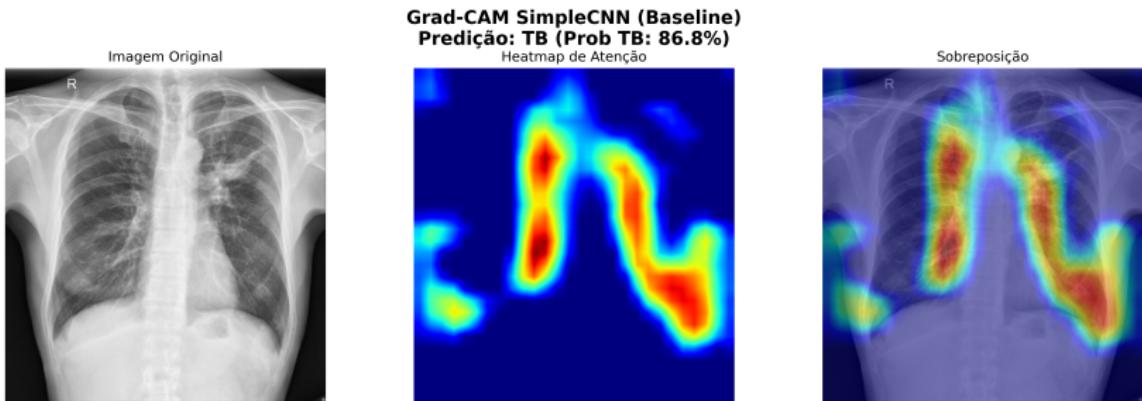


Figura: Grad-CAM na camada Conv4 da SimpleCNN. A rede identifica as opacidades, apresentando áreas de ativação dispersas sobre a região pulmonar.



Grad-CAM (ResNet-50) vs Baseline

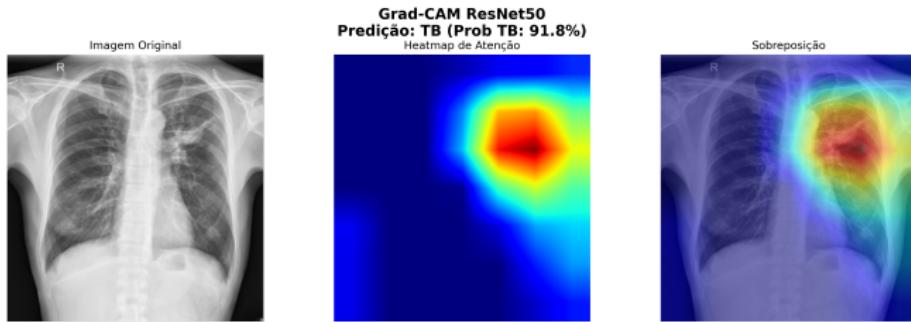
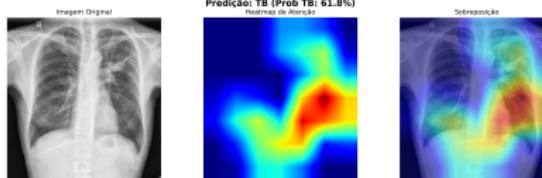


Figura: Mapa de atenção (Grad-CAM) da ResNet-50. Note como o foco é mais concentrado e menos ruidoso em comparação à Baseline.

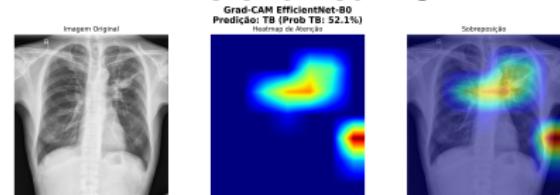


Grad-CAM (Outras Arquiteturas)

DenseNet121



EfficientNet-B0



Síntese da Comparação Visual:

- ▶ **ResNet-50 (Slide anterior):** Foco mais intenso e contínuo (correlaciona com maior Sensibilidade).
- ▶ **EfficientNet-B0:** Foco mais recortado e conservador (correlaciona com maior Especificidade).
- ▶ **DenseNet-121:** Foco mais granular e disperso (atenção a detalhes de textura).



Conclusões



Conclusões e Recomendação

Recomendação Principal: ResNet-50

- ▶ **Maior Sensibilidade (90%)**: Melhor modelo para **triagem**, minimizando Falsos Negativos (crítico em saúde).
- ▶ **Maior AUC-ROC (96%)**: Melhor capacidade geral de separação entre classes.

Alternativas (Trade-offs):

- ▶ **EfficientNet**: Máxima Especificidade (94%). Ideal para confirmação / reduzir falsos alarmes.
- ▶ **DenseNet**: Aprendizado mais rápido, mas instável no final.

Contribuição Técnica:

- ▶ **Transfer Learning**: Provou reduzir drasticamente o ruído visual (visto no Grad-CAM) comparado à CNN do zero.
- ▶ **Generalização**: Modelos pré-treinados evitaram o overfitting severo, comum



Obrigado!

Perguntas?



Referências

-  Lakhani, P., & Sundaram, B. (2017). Deep learning at chest radiography: automated classification of pulmonary tuberculosis by using convolutional neural networks. *Radiology*, 284(2), 574-582. [1]
-  Rajaraman, S., et al. (2018). Pre-trained convolutional neural networks as feature extractors toward improved malaria parasite detection in thin blood smear images. *PeerJ*, 6, e4568. [2]
-  Hwang, S., et al. (2016). A novel approach for tuberculosis screening based on deep convolutional neural networks. *Medical Imaging 2016: Computer-Aided Diagnosis*. [3]
-  Pan, S. J., & Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10), 1345-1359. [4]
-  He, K., et al. (2016). Deep residual learning for image recognition. *CvPR*. [5]
-  Huang, G., et al. (2017). Densely connected convolutional networks. *CVPR*. [6]
-  Tan, M., & Le, Q. (2019). EfficientNet: Rethinking model scaling for CNNs. *ICML*. [7]
-  Selvaraju, R. R., et al. (2017). Grad-CAM: Visual explanations from deep networks via gradient-based localization. *ICCV*. [8]
-  World Health Organization. (2022). *Global tuberculosis report 2022*. WHO. [9]