

# EfficientBacillus: uma arquitetura profunda para detecção dos bacilos de Koch

Patrick Ryan Sales dos Santos<sup>1</sup>, Vitória de Carvalho Brito<sup>1</sup>,  
Antonio Oseas de Carvalho Filho<sup>1</sup>, Alcilene Dalília de Sousa<sup>1</sup>, João  
Otávio Bandeira Diniz<sup>2</sup>, Daniel de Sousa Luz<sup>3</sup>

<sup>1</sup>Universidade Federal do Piauí - UFPI  
Programa de Pós-Graduação em Engenharia Elétrica - PPGEE  
64607-670 – Picos – PI – Brazil

<sup>2</sup>Instituto Federal do Maranhão - IFMA  
65940-000 – Grajaú – MA – Brazil

<sup>3</sup>Instituto Federal do Piauí - IFPI  
64605-500 – Picos – PI – Brazil

{sales, vitoriacarvalho, antoniooseas}@ufpi.edu.br, alcilene luzsousa@gmail.com

joao.bandeira@ifma.edu.br, daniel.luz@ifpi.edu.br

**Abstract.** *Tuberculosis is a bacterial infection caused by Koch's bacillus and is transmitted through the air. The disease mainly affects the lungs and is considered the second leading cause of death from infection in the world. Despite this, tuberculosis is curable and early diagnosis is of paramount importance for successful treatment and to prevent the spread of the disease. Traditionally, sputum smear microscopy has been the main method for diagnosing and monitoring tuberculosis treatment. In this context, several computational approaches have been developed to aid in the diagnosis of tuberculosis, through the analysis of bacilloscopy images. In this work, we propose the use of EfficientDet, exploring each of its backbones in the bacilli identification task. We also evaluated 4 different color representations and applied a cross-validation with  $k$ -fold = 5. The results were promising, with IoU of 0.523, recall of 0.925, precision of 0.694 and  $f1$ -score of 0.774. The results obtained evidenced the potential of the method in the detection of bacilli, which may help in the diagnosis of tuberculosis.*

**Resumo.** *A tuberculose é uma infecção bacteriana causada pelo bacilo de Koch e é transmitida pelo ar. A doença afeta principalmente os pulmões e é considerada a segunda principal causa de morte por infecção no mundo. Apesar disso, a tuberculose tem cura e o diagnóstico precoce é de suma importância para o sucesso do tratamento e para evitar a disseminação da doença. Tradicionalmente, a baciloscopia de esfregaço de escarro tem sido o principal método para o diagnóstico e monitoramento do tratamento da tuberculose. Nesse contexto, diversas abordagens computacionais têm sido desenvolvidas para auxiliar no diagnóstico da tuberculose, através da análise das imagens de baciloscopia. Neste trabalho, propomos a utilização da EfficientDet, explorando cada um dos seus backbones na tarefa de identificação dos bacilos. Também avaliamos 4 representações de cor diferentes e aplicamos uma validação cruzada com  $k$ -fold = 5. Os resultados mostraram-se promissores, com IoU de 0,523, recall*

de 0,925, precisão de 0,694 e f1-score de 0,774. Os resultados alcançados evidenciaram o potencial do método na detecção dos bacilos, podendo auxiliar no diagnóstico da tuberculose.

## 1. Introdução

A tuberculose (TB) é uma infecção bacteriana causada pelo *Mycobacterium tuberculosis* (Mtb ou bacilo de Koch) que afeta principalmente os pulmões, mas também pode acometer outros órgãos. A transmissão da TB ocorre pelo ar, através da inalação de gotículas e aerossóis produzidos pela fala, espirro ou tosse de uma pessoa infectada [Reist et al. 2021]. O paciente com TB pode apresentar sintomas como tosse, febre, sudorese noturna ou perda de peso. [Díaz-Huerta et al. 2022].

De acordo com a Organização Mundial de Saúde, 10,6 milhões de pessoas foram acometidas pela TB no mundo em 2021. Neste mesmo ano, 1,6 milhão de pessoas morreram em decorrência da doença, apontada como a segunda principal causa de morte por infecção no mundo, atrás apenas da COVID-19 [WHO 2021]. Apesar dos dados preocupantes, a TB tem cura. O diagnóstico precoce seguido de um tratamento eficaz possibilita maiores chances de cura e recuperação.[Zachariou et al. 2022].

O diagnóstico da TB pode ser realizado através de exames laboratoriais, como radiografia de tórax, testes de cultura, testes moleculares rápidos e baciloscopia de escarro (SSM). Tradicionalmente, a SSM tem sido o principal método para o diagnóstico e controle do tratamento da TB. As amostras de escarro (muco pulmonar, saliva) são fixadas em lâminas e coloradas através de técnicas que rotulam os bacilos ácido-resistentes (AFB) como células Mtb [Zachariou et al. 2022].

Este teste deve ser realizado por um bacteriologista e pode ser um processo demorado e exaustivo, tendo em vista a necessidade de se analisar pelo menos 100 campos da amostra [Rico-Garcia et al. 2015]. Alguns slides podem ser difíceis de se analisar porque alguns componentes não bacterianos se assemelham às células Mtb [Zachariou et al. 2022]. Dessa forma, a sensibilidade do resultado depende da experiência do especialista [Shah et al. 2016]. Diante disso, a automatização da análise das imagens de lâminas para a detecção e contagem dos bacilos pode auxiliar o bacteriologista no processo diagnóstico, tornando-o mais ágil, menos cansativo e menos propenso a erros.

Abordagens computacionais baseadas em análise de imagens têm sido desenvolvidas para apoiar o diagnóstico da TB, dentre as quais se destacam as baseadas em aprendizado profundo. Alguns desafios, como baixa qualidade das imagens microscópicas e indisponibilidade de conjuntos de dados, dificultam a detecção dos bacilos no processamento das imagens [Kotei and Thirunavukarasu 2022]. Neste trabalho, propomos a avaliação do modelo *EfficientDet* [Tan et al. 2020] e seus *backbones* na tarefa de detecção dos bacilos de Koch, avaliando diferentes esquemas de cores como pré-processamento das imagens, de forma a estimar o impacto da representação dos canais no realce dos bacilos.

Diferente das abordagens tradicionais, a *EfficientDet* realiza a fusão de características considerando que elas apresentam contribuições diferentes para a saída dependendo da diferença de resoluções. Com essa fusão multiescala eficiente, o modelo acaba ajustando os pesos para aprender a importância das diferentes características de entrada [Wu et al. 2022]. Assim, este trabalho propõe aplicar os benefícios da *EfficientDet*

na detecção dos bacilos da TB, sob diferentes representações de cores e aplicando uma avaliação robusta do método.

## 2. Trabalhos Relacionados

A literatura aponta uma série de trabalhos direcionados à detecção da TB, com abordagens voltadas ao melhoramento das imagens das lâminas coradas, segmentação dos bacilos, classificação das lâminas quanto à presença de bacilos da TB e até mesmo detecção dos bacilos. Nesta seção, apresentamos trabalhos recentes e relevantes relacionados aos métodos desenvolvidos para esse fim, e comparamos com o método proposto neste trabalho.

Em [Yousefi et al. 2020], um novo modelo estatístico é proposto para a identificação dos bacilos, considerando aspectos de cor e forma e um *dataset* com 43 imagens de SSM. Os autores aplicaram técnicas de pré-processamento para aprimorar as imagens antes da etapa de segmentação. Após a segmentação, características de forma e cor das imagens foram extraídas para servir de insumo para o classificador *k-Nearest Neighbors* (K-NN). Os resultados mostraram uma acurácia média de 82,7% para a identificação de bacilos únicos e sobrepostos, e 99,1% de acurácia na identificação de bacilos individuais de bacilos sobrepostos e outros objetos.

O método de [Shwetha et al. 2021] também inclui técnicas de pré-processamento, segmentação e classificação dos bacilos. A segmentação é feita com base no agrupamento *K-means* e, em seguida, são avaliadas 3 CNNs pré-treinadas (VGG16, ResNet50 e SqueezeNet) para classificar as imagens segmentadas em bacilos ou células. O *dataset* utilizado foi o *ZN sputum smear microscopy image database* (ZNSM-iDB), que contempla cerca de 2000 conjuntos de imagens em diferentes colorações. O método alcançou uma acurácia geral de 97% usando a SqueezeNet.

Diferente dos trabalhos anteriores, o método de [Reist et al. 2021] explora a detecção dos bacilos da TB utilizando um modelo de detecção de objetos, a RetinaNet. O modelo foi treinado e testado com um *dataset* público composto por 928 imagens de SSM, o mesmo utilizado neste trabalho. Os autores também aplicaram uma etapa de pré-processamento a fim de realçar as imagens. Os resultados mostraram mAP de 0,794, *recall* de 0,8656, precisão de 0,671 e f1-score de 0,7561.

A importância do pré-processamento nas imagens de SSM também é destacada em [Díaz-Huerta et al. 2022], onde os autores aplicaram técnicas para ajuste de brilho e contraste nas imagens antes da segmentação dos bacilos. Os métodos analisados na etapa de segmentação foram a *Artificial Neural Network* (ANN) e o classificador Bayesiano com *Gaussian mixture model* (GMM), com os quais o trabalho alcançou acurácia de 99,4% *Jaccard index* de 99,7% AUC de 0,9999. A aquisição de imagens também foi realizada pelos autores.

O trabalho de [Zachariou et al. 2022] propõe um *pipeline* composto por quatro estágios para a detecção dos bacilos. Inicialmente, os autores empregam uma etapa baseada em processamento de imagens para aprimorar as imagens de lâminas. Em seguida, *Generative Adversarial Networks* (GANs) são aplicadas para a segmentação semântica das lâminas. A partir das imagens segmentadas, *patches* das lâminas são extraídos e classificados. Por fim, os autores utilizaram regressão para contar a quantidade de bactérias

presentes em um *patch*. O *dataset* utilizado no trabalho contempla dados de 46 pacientes ao longo de 6 meses. Os autores alcançaram acurácia de 99,74%, *recall* de 0,967, precisão de 0,990 e f1-score de 0,960 na classificação dos *patches* e *Mean Squared Error* (MSE) de 0,0054 na contagem das bactérias.

Em [An et al. 2022], a detecção dos bacilos é explorada através de um aprimoramento da YOLOv5, na qual os autores chamaram de DA-YOLO. Esta abordagem combina dois mecanismos de atenção com o intuito de obter características mais robustas por meio do *backbone* da rede. O *dataset* empregado para a avaliação do método é composto por 1265 imagens de SSM, com 9969 marcações de bacilos. O método alcançou mAP de 0,876, *recall* de 0,80 e precisão de 0,81.

Em geral, os trabalhos variam entre segmentação e detecção, sendo principalmente de segmentação. Os trabalhos que aplicam detecção dos bacilos exploram modelos amplamente utilizados para detecção de objetos e utilizam imagens semelhantes às utilizadas neste trabalho. Contudo, propomos a utilização de um modelo de detecção de objetos que combina eficiência e precisão, avaliando diferentes *backbones* a fim de encontrar a melhor representação das características. Além disso, buscou-se encontrar a melhor representação dos canais da imagem para o modelo.

### 3. Metodologia

Esta seção descreve os passos envolvidos neste trabalho para a detecção dos bacilos da TB em imagens de SSM. Inicialmente, as imagens foram adquiridas através de um *dataset* de imagens de SSM publicamente disponível. Em seguida, elas passaram por uma etapa de pré-processamento, onde 4 esquemas de cores foram avaliados para encontrar a melhor representação de cor para o modelo, ou seja, aquela que fornece o melhor realce das características dos bacilos para o modelo. Após a normalização destas imagens, foram realizados os experimentos com a EfficientDet e seus *backbones* numa etapa de treinamento e teste. O método foi avaliado através da validação cruzada com 5 *folds* e métricas da validação amplamente aplicadas na literatura. A Figura 1 ilustra o fluxo da metodologia apresentada e, a seguir, cada uma das etapas é descrita.

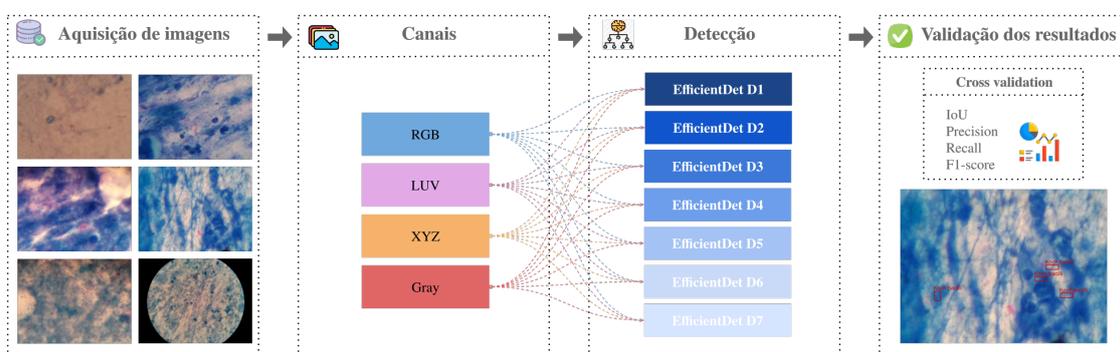


Figura 1. Representação da metodologia aplicada.

#### 3.1. Aquisição de Imagens

As imagens de SSM utilizadas para a avaliação do método proposto neste trabalho foram obtidas a partir de um *dataset* público disponibilizado pelo *AI Research and Automated*

*Laboratory Diagnostics* no Kaggle [Kaggle 2020]. Existem 928 imagens no *dataset*, todas com resolução de 1224x1623 e com um arquivo XML contendo as marcações dos bacilos em um formato de *bounding box*. A Figura 2 mostra alguns exemplos de imagens do *dataset*, com suas respectivas anotações dos bacilos. É possível notar a variação no aspecto visual das amostras, sendo que em alguma delas os bacilos se destacam e em outras não.

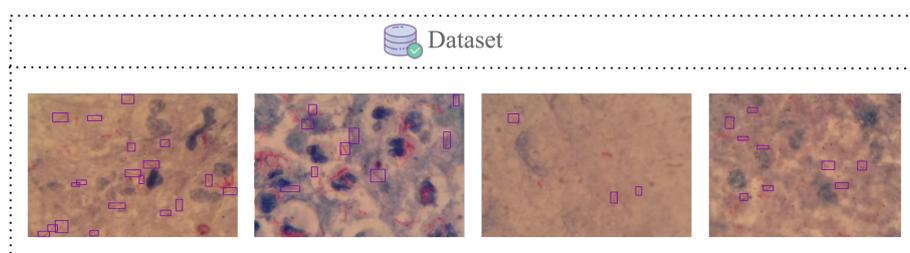


Figura 2. Exemplos de imagens do *dataset* com as marcações dos bacilos.

### 3.2. Pré-processamento

Neste trabalho, avaliamos o impacto da representação de cor através de 4 esquemas de cores tradicionais: RGB, HSV, LUV e XYZ. O intuito é aplicar a representação de cor que proporciona os melhores resultados ao modelo, ou seja, que melhor realça as estruturas dos bacilos. Outras operações, como ajuste de brilho e equalização de histograma, também são aplicadas na metodologia proposta, mas apenas dentro do aumento de dados da rede, a fim de dificultar o treinamento do modelo, influenciando-o a aprender diferentes padrões das imagens, com o intuito de avaliar essas diferentes representações de cores, executamos uma bateria de experimento variando os canais com os 7 *backbones* da *EfficientDet*, resultando assim em 28 experimentos, como ilustrado pelas linhas pontilhadas na figura 1.

### 3.3. EfficientDet

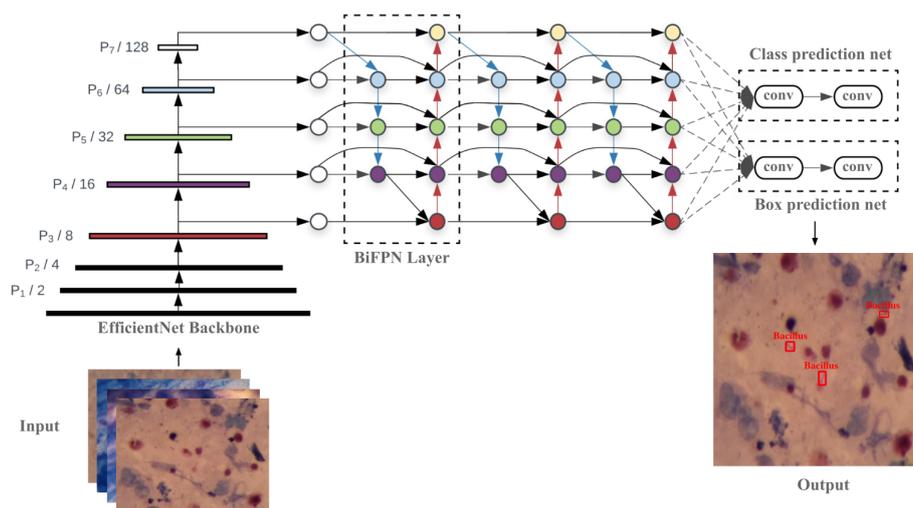
A *EfficientDet* é um modelo de detecção de objetos proposto pelo *Google Brain Team* [Tan et al. 2020], que caracteriza-se, principalmente, pelo uso de *backbones* e da *BiFPN*, além de ajustes de otimização. Com a fusão eficiente dos diferentes mapas de característica multiescala, a *EfficientDet* superou modelos do estado da arte em termos de precisão e eficiência [Tan et al. 2020].

A *EfficientDet* apresenta-se como um modelo de detecção leve e escalável, contendo um total de oito *backbones*, do D0 ao D7. A medida que o tamanho do modelo aumenta, também tem-se um aumento do poder de generalização e complexidade de tempo. O modelo utilizado como *backbone* da *EfficientDet* é a *EfficientNet* [Tan and Le 2019], que aplica convoluções separáveis para deixar o modelo mais leve [Song et al. 2021]. Em geral, a *EfficientNet*, também proposta pelo *Google Brain Team*, alcança melhor precisão e eficiência quando comparado às CNNs existentes [Marques et al. 2020].

A arquitetura da *EfficientDet* propõe o uso da *BiFPN*, onde os pesos são aprendidos de forma a penalizar as características com base em sua importância para a saída, ou seja, na fusão das características multiescala com fluxo de baixo para cima e de cima para baixo [Mekhali et al. 2021]. A *EfficientDet* também contempla um novo método

de escalonamento composto, que aumenta de forma conjunta resolução, profundidade e largura. A ideia é que cada componente da arquitetura (*backbone*, rede de características e rede de classificação e localização dos objetos) tenha um único fator de escala composto, que através de regras baseadas em heurística, controla todas as dimensões de escala [Tan et al. 2020].

Representamos na Figura 3 a arquitetura da *EfficientDet* utilizada neste trabalho.



**Figura 3. Arquitetura da EfficientDet.** Adaptada de [Tan et al. 2020, Mekhafi et al. 2021]. Backbone, responsável por extrair características e BiFPN Layer camada adicional que é adicionada à saída do backbone. BiFPN Layer, composta por um conjunto de camadas de convolução bidirecional, responsável por processar as características das camadas superiores e inferiores da FPN (Feature Pyramid Network) em paralelo.

### 3.4. Detalhes de Implementação

A primeira etapa após a aquisição de imagens foi o pré-processamento, onde 4 esquemas de cores diferentes foram avaliados: RGB, HSV, LUV e XYZ. Os *backbones* avaliados foram B1 a B7 (nomenclatura da EfficientNet, equivalente aos *backbones* D1 a D7 da *EfficientDet*), que diferem em termos de profundidade, dimensão de entrada e, consequentemente, poder de generalização e eficiência. Desconsideramos a avaliação com o *backbone* B0, por apresentar uma similaridade grande com B1. Os pesos dos experimentos foram iniciados de forma aleatória, sem a utilização de *transfer learning* e, com o intuito de deixar cada experimento equiparado em relação a quantidade de épocas, optou-se pela não utilização de *early stopping* durante os treinamentos, para assim, facilitar uma comparação época por época.

Executou-se o algoritmo *Grid Search* levando em consideração o menor erro, utilizando 5 épocas de treinamento, com 10% da base. A fim de encontrar o melhor *input size* e otimizador para cada *backbone*. As dimensões para a busca foram 224, 299 e 512. Já os otimizadores, a saber: *Adadelta*, *Adam*, *AdamW*, *Adamax*, *ASGD*, *NAdam*, *RAdam*, *RMSprop*, *Rprop* and *SGD*.

Em cada experimento, foram utilizados alguns hiperparâmetros definidos empiricamente, a saber:  $\gamma = 0.1$ , *batch size* = 4 e número de épocas = 120. O tamanho da

entrada foi definido como  $512 \times 512$  e o otimizador foi o *RMSprop*, ambos escolhidos dentro de um espaço de busca pelo algoritmo *Grid Search*. Em todos os experimentos foi utilizada a estratégia de decaimento de *learning rate* (LR), iniciando em  $1e-3$ . Também empregamos o *focal loss*, amplamente utilizado com  $\alpha = 0,25$  e  $\gamma = 1,5$ .

Tendo em vista a variação nas imagens de lâminas mencionada na Seção 3.1, aplicamos além dos diferentes esquemas de cores um aumento de dados, a fim de proporcionar uma melhor robustez e generalização no aprendizado do modelo. O aumento de dados neste trabalho foi realizado dinamicamente, ou seja, em tempo de execução, sendo utilizado uma probabilidade de 50% para cada função de transformação. As transformações aplicadas foram: *horizontal flip*, *vertical flip*, *random brightness contrast*, *random rotate 90°*, *transpose*, *CLAHE*, *gauss noise* e *hue saturation value*.

As implementações de código realizadas neste trabalho foram escritas na linguagem *Python*, utilizando a *framework PyTorch*, e executadas em uma máquina com a seguinte configuração: processador Intel Xeon E5-2683 v4 de 2.10GHz, memória RAM de 128 GB, placa de vídeo Nvidia Tesla K80 24GB e sistema operacional Ubuntu 18.04 LTS.

### 3.5. Validação

Para avaliar a capacidade de generalização do modelo no *dataset* utilizado, optou-se por realizar uma validação cruzada com  $k = 5$ , dividindo aleatoriamente o *dataset* em 5 *folds* de tamanho aproximadamente igual. A validação cruzada foi executada para cada *backbone*, em cada esquema de cor testado, ou seja, todos os experimentos foram treinados e avaliados utilizando a estratégia de validação cruzada. As métricas utilizadas para validar o desempenho do modelo foram *Intersection Over Union* (IoU), *recall*, *precision* e *f1-score*. Cada uma dessas métricas é calculada com base na matriz de confusão.

## 4. Resultados

Nesta Seção, apresentamos os resultados alcançados com a execução da metodologia proposta. A Tabela 1 mostra os resultados médios dos experimentos em cada *fold*, após 120 épocas de treinamento. Os gráficos de desempenho dos melhores experimentos estão ilustrados na Figura 4.

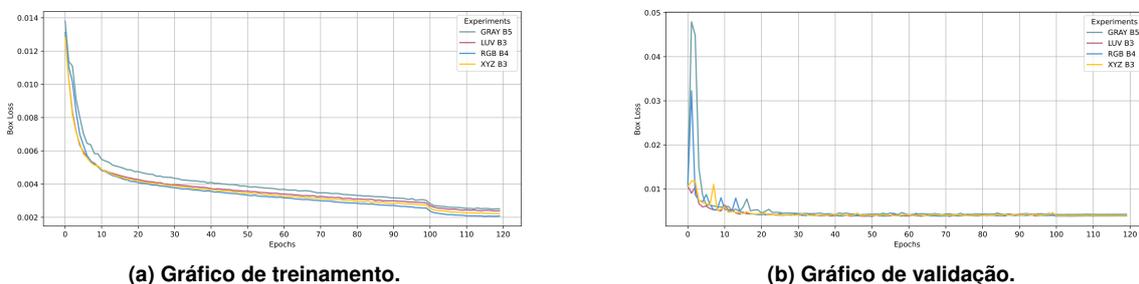


Figura 4. Gráficos de desempenho dos melhores resultados da Tabela 1.

Analisando a Tabela 1, é possível notar que os melhores resultados estão entre as colunas dos *backbones* B3 a B5, acredita-se que tais *backbones* forneceram os melhores resultados, pelo fato dos bacilos tratarem-se de pequenas estruturas quando comparado ao *patch* da lamina, sendo assim, *backbones* com profundidade média tende a melhor representar essas estruturas ao final da etapa de extração de característica. Para fazer esta

**Tabela 1. Resultados alcançados com o método proposto.**

		Backbones						
		B1	B2	B3	B4	B5	B6	B7
RGB	IOU	0,498 ± 0,028	0,527 ± 0,009	0,419 ± 0,026	0,535 ± 0,009	0,46 ± 0,016	0,426 ± 0,093	0,477 ± 0,017
	Rec	0,905 ± 0,012	0,908 ± 0,010	0,871 ± 0,025	0,908 ± 0,020	0,852 ± 0,012	0,856 ± 0,048	0,857 ± 0,026
	Prec	0,666 ± 0,012	0,691 ± 0,006	0,569 ± 0,041	0,701 ± 0,013	0,585 ± 0,027	0,582 ± 0,062	0,624 ± 0,019
	F1	0,749 ± 0,013	0,767 ± 0,009	0,662 ± 0,037	<b>0,772 ± 0,015</b>	0,673 ± 0,022	0,664 ± 0,057	0,697 ± 0,018
LUV	IOU	0,472 ± 0,088	0,524 ± 0,006	0,523 ± 0,016	0,439 ± 0,007	0,355 ± 0,180	0,308 ± 0,185	0,457 ± 0,025
	Rec	0,928 ± 0,036	0,920 ± 0,020	0,925 ± 0,009	0,896 ± 0,013	0,764 ± 0,291	0,703 ± 0,393	0,893 ± 0,011
	Prec	0,651 ± 0,064	0,685 ± 0,005	0,694 ± 0,016	0,574 ± 0,040	0,485 ± 0,210	0,423 ± 0,240	0,609 ± 0,011
	F1	0,742 ± 0,066	0,768 ± 0,010	<b>0,774 ± 0,015</b>	0,675 ± 0,032	0,571 ± 0,241	0,505 ± 0,284	0,702 ± 0,009
XYZ	IOU	0,524 ± 0,006	0,426 ± 0,006	0,524 ± 0,016	0,442 ± 0,014	0,426 ± 0,028	0,269 ± 0,246	0,36 ± 0,137
	Rec	0,919 ± 0,027	0,872 ± 0,029	0,911 ± 0,021	0,858 ± 0,025	0,861 ± 0,017	0,519 ± 0,474	0,844 ± 0,054
	Prec	0,687 ± 0,012	0,549 ± 0,019	0,688 ± 0,017	0,576 ± 0,027	0,561 ± 0,029	0,345 ± 0,315	0,541 ± 0,071
	F1	0,766 ± 0,019	0,649 ± 0,020	<b>0,766 ± 0,014</b>	0,666 ± 0,026	0,655 ± 0,028	0,398 ± 0,364	0,619 ± 0,079
GRAY	IOU	0,494 ± 0,011	0,508 ± 0,015	0,509 ± 0,018	0,406 ± 0,015	0,513 ± 0,024	0,306 ± 0,064	0,398 ± 0,041
	Rec	0,836 ± 0,024	0,820 ± 0,019	0,832 ± 0,022	0,741 ± 0,039	0,836 ± 0,034	0,628 ± 0,179	0,767 ± 0,031
	Prec	0,633 ± 0,008	0,649 ± 0,022	0,653 ± 0,026	0,502 ± 0,028	0,655 ± 0,032	0,390 ± 0,140	0,518 ± 0,056
	F1	0,698 ± 0,010	0,701 ± 0,020	0,708 ± 0,021	0,574 ± 0,032	<b>0,712 ± 0,030</b>	0,452 ± 0,155	0,583 ± 0,048

- Os resultados em negrito referem-se aos melhores resultados de cada canal utilizando como base a métrica *f1-score*.

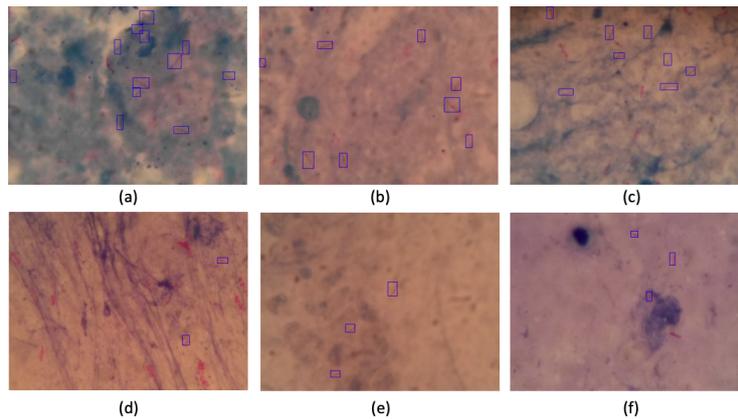
comparação, utilizamos a métrica *f1-score* em razão de apresentar uma média harmônica entre *precision* e *recall*, visando uma comparação a fim de trazer um número único que determine a qualidade geral do modelo analisado. Com relação aos canais, o canal LUV apresentou o melhor resultado, no entanto, com um empate técnico em relação aos melhores resultados nos demais canais.

Ao analisar os gráficos de desempenho dos experimentos que apresentaram os melhores resultados (Figura 4), é possível notar que o desempenho do treinamento condiz com os resultados de teste. O treinamento (Figura 4a) terminou com uma diferença de erro menor que 0,002 entre os experimentos e podemos notar uma tendência de estabilização do aprendizado por volta da época 110, sendo assim, 120 épocas mostrou-se suficiente para o treinamento. Já na validação (Figura 4b), o comportamento foi um pouco diferente, pois a partir da época 30, aproximadamente, a rede manteve uma *loss* estável. Contudo, os gráficos não mostraram tendência de *overfitting*, já que o *loss* de treino continuou caindo apesar da validação apresentar estabilidade.

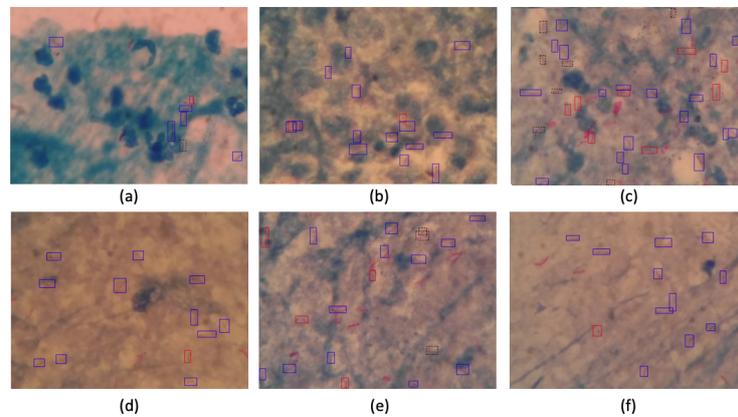
Para melhor ilustrar os resultados, é apresentado nas Figuras 5 e 6, casos de sucesso e falhas na detecção dos bacilos pelo modelo apresentado neste trabalho. Para gerar as Figuras 5 e 6, utilizou-se os modelo de cores RGB com o *backbone* 4. Em azul, foi marcado os bacilos corretamente identificados pelo modelo (*True Positives (TP)*); em vermelho os bacilos delimitados erroneamente pelo modelo (*False Positives (FP)*); e nos retângulos com linhas pontilhadas, os casos em que o modelo não conseguiu identificar o bacilo (*False Negatives (FN)*).

Como é possível observar na Figura 5, o modelo proposto apresenta resultados promissores na detecção dos bacilos em tipos de imagens com diferentes contrastes. Isso mostra que o modelo é robusto e consegue performar bem em diferentes cenários.

Na Figura 6, é possível observar que o método marca regiões como FP que apresentam grande semelhança aos bacilos, enquanto outras regiões, embora semelhantes a um bacilo, o modelo não consegue identificar como tal (FN). No caso dos FPs, acredita-se que isso esteja ocorrendo devido a fatores como textura, cor e forma serem muito semelhantes a dos bacilos. Isso pode atrapalhar no aprendizado da rede, pois, no momento do



**Figura 5. Casos de sucesso na detecção dos bacilos.**



**Figura 6. Casos de falha na detecção dos bacilos.**

treinamento, o modelo tende a ser penalizado por prever regiões similares, mesmo que essas regiões estejam compartilhando características semelhantes.

Com relação aos FNs na Figura 6, acredita-se que, por conta da grande semelhança entre os bacilos, o modelo sofre penalização quando detecta um bacilo que na verdade não existe marcação pra ele, então a rede acaba tentando aprender alguma outra característica (como posição, orientação, etc) para compensar o aprendizado com características que não são relevantes a se analisar no problema. Dessa forma, alguns bacilos mais isolados ou um pouco diferentes da maioria, acabam não sendo identificados com uma confiança satisfatória.

## 5. Discussões

Nesta Seção, é apresentado uma breve discussão sobre os resultados alcançados, destacando alguns pontos sobre o método proposto e comparando com os estudos relacionados. Na Tabela 2, um resumo dos trabalhos relacionados e seus respectivos resultados.

Dos trabalhos apresentados na Tabela 2, apenas o de [Reist et al. 2021] utilizou o mesmo *dataset* do trabalho proposto, entretanto o método proposto por este trabalho apresentou melhores resultados, considerando as métricas comuns *recall*, *precision* e *f1-score*. Em relação ao tamanho da amostra, alguns trabalhos utilizaram uma quantidade menor de imagens em comparação aos demais trabalhos, mas alcançaram resultados pro-

**Tabela 2. Comparação do método proposto com estudos relacionados.**

Trabalho	Objetivo	Amostra	Conjunto de dados público	Métricas de validação
[Yousefi et al. 2020]	Segmentação e classificação dos bacilos usando um modelo estatístico de cor e forma	43	Não	<i>Average accuracy</i> = 0.827
[Shwetha et al. 2021]	Pré-processamento, segmentação e classificação dos bacilos usando <i>K-means</i> e 3 CNNs pré-treinadas	2000	Sim	<i>Overall accuracy</i> = 0.97
[Reist et al. 2021]	Pré-processamento + detecção dos bacilos da TB usando RetinaNet	928	Sim	mAP = 0,794 Recall = 0,8656 Precision = 0,671 F1-score = 0,7561
[Díaz-Huerta et al. 2022]	Pré-processamento + segmentação dos bacilos usando ANN e classificador Bayesiano com GMM	160	Não	<i>Accuracy</i> = 0,994 <i>Jaccard index</i> = 0,997 <i>AUC</i> = 0,9999
[Zachariou et al. 2022]	Pré-processamento + segmentação de lâminas usando GAN + extração e classificação dos <i>patches</i> + regressão para contar a quantidade de bactérias em uma <i>patch</i>	500	Não informado	<i>Accuracy</i> = 0,9974 <i>Recall</i> = 0,967 <i>Precision</i> = 0,990 <i>F1-score</i> = 0,960 MSE = 0,0054
[An et al. 2022]	Detecção de bacilos usando DA-YOLO	1265	Não informado	mAP = 0,876 Recall = 0,80 Precision = 0,81
Método proposto	Pré-processamento + detecção de bacilos usando EfficientDet e seus <i>backbones</i>	928	Sim	IoU = 0,523 Recall = 0,925 Precision = 0,694 F1-score = 0,774

missores. O método está entre as 3 maiores amostras de imagem da tabela. Sobre as abordagens, algumas aplicaram métodos clássicos para a identificação dos bacilos. Esses métodos, embora apresentem um bom desempenho em termos de eficiência e resultados, podem acabar se limitando às características do *dataset*, tendo em vista a variação na coloração das lâminas e dos bacilos. Por fim, em comparação com o método proposto por [An et al. 2022], que também utiliza uma rede de detecção em sua abordagem, o método proposto por este trabalho alcançou resultados próximos, inclusive com um melhor *recall*.

Em geral, é possível perceber que o método alcançou resultados promissores. A seguir, destaca-se alguns pontos sobre o método proposto:

- É explorado no problema da detecção de bacilos um modelo recente de detecção de objetos que realiza uma fusão multiescala das características e apresenta diferentes *backbones* que variam em tamanho e complexidade;
- Apesar da complexidade do problema e da variação nos aspectos visuais e estruturais das lâminas e dos bacilos, o método alcançou resultados próximos à literatura relacionada;
- A validação cruzada proporcionou uma validação robusta, mostrando um resultado que independe da divisão do *dataset*;
- Por meio da variação dos esquemas de cores, tornou-se possível avaliar a melhor representação de imagens para o modelo. Além disso, utilizou-se um aumento de dados, que consistiu em diversas operações, com o objetivo de permitir que o modelo aprendesse diferentes padrões das imagens;
- A realização de experimentos com os *backbones* da EfficientDet (D1 a D7) permitiu uma avaliação mais precisa do método em relação ao problema de detecção de bacilos, destacando que certos modelos podem ser mais adequados para a identificação desses objetos pequenos nas imagens;
- Foi utilizado um *dataset* público de imagens, facilitando a reprodução e comparação dos resultados com outros métodos.

## 6. Conclusão

Neste trabalho, é apresentado uma metodologia para a detecção de bacilos da TB utilizando a *EfficientDet*, através de experimentos com cada um dos seus *backbones*. é utilizado um *dataset* público com imagens de SSM e explorou-se 4 modelos de cores diferentes durante os treinamentos, que foram avaliados a partir da validação cruzada com 5 *folds*. Apesar da complexidade do problema, tendo em vista a variação das imagens e dos bacilos em termos de características visuais e estruturais, o método obteve um desempenho satisfatório, superando alguns resultados da literatura relacionada. Dessa forma, pode-se constatar o potencial da *EfficientDet* para a detecção dos bacilos da TB, bem como a efetividade dos diferentes *backbones* avaliados, que apresentaram diferenças significativas em relação à precisão na identificação de objetos pequenos.

Como trabalhos futuros, pretende-se:

- Utilizar outros *datasets* com bacilos da TB, públicos ou privados;
- Estender a classificação para outros tipos de bacilos que se assemelham ao bacilo de Koch, como por exemplo o bacilo da hanseníase;
- Explorar outras estratégias de pré-processamento para aprimorar a identificação dos bacilos nas imagens;
- Aplicar técnicas de explicabilidade do modelo para compreender a influência dos hiperparâmetros e, assim, implementar melhorias que possam impactar positivamente no desempenho do modelo.

## Agradecimentos

O método proposto foi apoiado pelas seguintes instituições: FAPEPI - [www.fapepi.pi.gov.br](http://www.fapepi.pi.gov.br) (013/2021 - 004/2020); CAPES - [www.capes.gov.br](http://www.capes.gov.br); and the CNPq - [www.cnpq.br](http://www.cnpq.br) (305025/2021-0).

## Referências

- An, L., Peng, K., Yang, X., Feng, P., and Huang, P. (2022). Automated detection of tuberculosis bacilli using deep neural networks with sputum smear images. In *2022 5th International Conference on Pattern Recognition and Artificial Intelligence (PRAI)*, pages 1040–1045. IEEE.
- Díaz-Huerta, J. L., Téllez-Anguiano, A. C., Gutiérrez-Gnecchi, J. A., Arellano-Calderón, S., and Olivares-Rojas, J. C. (2022). Analysis and comparison of image processing and artificial intelligence algorithms to detect afb in pulmonary tuberculosis images. *Tuberculosis*, 134:102196.
- Kaggle (2020). Tuberculosis image dataset. Retrieved from <https://www.kaggle.com/saife245/tuberculosis-image-datasets>. Accessed November 13, 2022.
- Kotei, E. and Thirunavukarasu, R. (2022). Computational techniques for the automated detection of mycobacterium tuberculosis from digitized sputum smear microscopic images: A systematic review. *Progress in Biophysics and Molecular Biology*.
- Marques, G., Agarwal, D., and de la Torre Díez, I. (2020). Automated medical diagnosis of covid-19 through efficientnet convolutional neural network. *Applied soft computing*, 96:106691.

- Mekhalfi, M. L., Nicolò, C., Bazi, Y., Al Rahhal, M. M., Alsharif, N. A., and Al Maghayreh, E. (2021). Contrasting yolov5, transformer, and efficientdet detectors for crop circle detection in desert. *IEEE Geoscience and Remote Sensing Letters*, 19:1–5.
- Reist, F. J. D. S., Veloso, M. A., Rodrigues, F. M. M., Brito, V. D. C., Dos Santos, P. R. S., Araújo, J. D. L., Rabelo, R. D. A. L., and De Carvalho Filho, A. O. (2021). Bacillusnet: An automated approach using retinanet for segmentation of pulmonary tuberculosis bacillus. In *2021 IEEE Symposium on Computers and Communications (ISCC)*, pages 1–4. IEEE.
- Rico-Garcia, M., Salazar, A., Madrigal, C.-A., Morantes-Guzman, L.-J., and Cortes-Mancera, F. (2015). Detection of mycobacterium tuberculosis in microscopic images of ziehl-neelsen-stained sputum smears lacnem 2015. In *6th Latin-American Conference on Networked and Electronic Media (LACNEM 2015)*, pages 1–6. IET.
- Shah, M. I., Mishra, S., Sarkar, M., and Rout, C. (2016). Automatic detection and classification of tuberculosis bacilli from zn-stained sputum smear images using watershed segmentation. In *International Conference on Signal Processing (ICSP 2016)*, pages 1–4.
- Shwetha, V., Prasad, K., Mukhopadhyay, C., Banerjee, B., and Chakrabarti, A. (2021). Automatic detection of bacilli bacteria from ziehl-neelsen sputum smear images. In *2021 2nd International Conference on Communication, Computing and Industry 4.0 (C2I4)*, pages 1–5. IEEE.
- Song, S., Jing, J., Huang, Y., and Shi, M. (2021). Efficientdet for fabric defect detection based on edge computing. *Journal of Engineered Fibers and Fabrics*, 16:15589250211008346.
- Tan, M. and Le, Q. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR.
- Tan, M., Pang, R., and Le, Q. V. (2020). Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10781–10790.
- WHO, W. H. O. (2021). Tuberculosis. Retrieved from <https://www.who.int/news-room/fact-sheets/detail/tuberculosis>. Accessed March 05, 2023.
- Wu, M., Heng, C., Zhu, H., and Cai, H. (2022). Covid-19 detection based on computer vision and big data. In *2022 7th International Conference on Intelligent Computing and Signal Processing (ICSP)*, pages 474–477. IEEE.
- Yousefi, H., Mohammadi, F., Mirian, N., and Amini, N. (2020). Tuberculosis bacilli identification: a novel feature extraction approach via statistical shape and color models. In *2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 366–371. IEEE.
- Zachariou, M., Arandjelović, O., Sabiiti, W., Mtafya, B., and Sloan, D. (2022). Tuberculosis bacteria detection and counting in fluorescence microscopy images using a multi-stage deep learning pipeline. *Information*, 13(2):96.