

KOCHDET: ARQUITETURA PROFUNDA BASEADA EM BIFPN PARA DETECÇÃO DE BACIOS DE KOCK

KOCHDET: BIFPN-BASED DEEP ARCHITECTURE FOR KOCH BACILLUS DETECTION

Patrick Ryan S. dos Santos
Programa de Pós Graduação em Engenharia Elétrica, Universidade Federal do Piauí - Teresina, Brasil
sales@ufpi.edu.br

Vitória de C. Brito
Programa de Pós Graduação em Engenharia Elétrica, Universidade Federal do Piauí - Teresina, Brasil
vitoriacarvalho@ufpi.edu.br

Antonio Oseas de C. Filho
Programa de Pós Graduação em Engenharia Elétrica, Departamento de Sistemas de Informação, Universidade Federal do Piauí, Brasil
antoniooseas@ufpi.edu.br

ABSTRACT

The bacterial infection known as tuberculosis is caused by Koch's bacillus and is airborne. This disease, which mainly affects the lungs, is considered the second leading cause of death from infection worldwide. Despite this, it is important to stress that TB is curable and that early diagnosis is crucial to ensure the effectiveness of treatment and to prevent the disease from spreading. Traditionally, sputum smear microscopy has been the most commonly used method to diagnose and monitor the treatment of TB. However, with advances in technology, several computational approaches have been developed to aid in this diagnosis, including the analysis of sputum smear images. In this study, we present an approach for bacilli detection using the EfficientDet architecture and exploiting each of its backbones. In addition, we evaluated four different color representations and applied a cross-validation with $k\text{-fold} = 5$. The results were promising, with an IoU of 0.523, recall of 0.925, precision of 0.694, and f1-score of 0.774. These results indicate the great potential of the method to aid in the diagnosis of tuberculosis by detecting bacilli.

Keywords

Aedes aegypti; U-net; Egg count; Parameter selection.

RESUMO

A infecção bacteriana conhecida como tuberculose é causada pelo bacilo de Koch e é transmitida pelo ar. Essa doença, que afeta principalmente os pulmões, é considerada a segunda maior causa de morte por infecção no mundo. Apesar disso, é importante ressaltar que a tuberculose tem cura e que um diagnóstico precoce é fundamental para garan-

tir a eficácia do tratamento e evitar a propagação da doença. Tradicionalmente, a baciloscopia de esfregaço de escarro tem sido o método mais utilizado para diagnosticar e monitorar o tratamento da tuberculose. No entanto, com o avanço da tecnologia, diversas abordagens computacionais têm sido desenvolvidas para auxiliar nesse diagnóstico, incluindo a análise de imagens de baciloscopia. Neste estudo, apresentamos uma abordagem para a detecção de bacilos usando a arquitetura EfficientDet e explorando cada um dos seus backbones. Além disso, avaliamos quatro representações de cor diferentes e aplicamos uma validação cruzada com $k\text{-fold} = 5$. Os resultados foram promissores, com um IoU de 0,523, recall de 0,925, precisão de 0,694 e f1-score de 0,774. Esses resultados indicam o grande potencial do método para auxiliar no diagnóstico da tuberculose por meio da detecção de bacilos.

Palavras-Chave

Tuberculose; bacilo de koch; EfficientDet, Baciloscopia

CCS Concepts

•AUC → Area under the receiver operating characteristic curve; •CNN → Convolutional neural network; •DL → Deep Learning;

1. INTRODUÇÃO

A tuberculose (TB) é uma doença infecciosa causada pela bactéria *Mycobacterium tuberculosis*, também conhecida como bacilo de Koch (Mtb), que afeta principalmente os pulmões, mas também pode afetar outros órgãos. A transmissão da TB ocorre por meio do ar, através da inalação de gotículas e aerossóis produzidos quando uma pessoa infectada fala, espirra ou tosse [7]. Os sintomas da tuberculose em um paciente podem incluir tosse, febre, sudorese noturna e perda de peso. No entanto, a TB ativa pode apresentar sintomas leves por vários meses, o que pode levar a atrasos no diagnóstico e tratamento da infecção. [2].

Segundo dados da Organização Mundial da Saúde, a tuberculose afetou cerca de 10,6 milhões de pessoas em todo o mundo em 2021. No mesmo período, a doença causou a morte de aproximadamente 1,6 milhão de pessoas, sendo

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

classificada como a segunda principal causa de morte por infecção no mundo, ficando atrás apenas da COVID-19 [14]. Embora os dados sejam preocupantes, é importante lembrar que a tuberculose tem cura. Quando diagnosticada precocemente e tratada adequadamente, há maiores chances de recuperação e cura, além de prevenir ou reduzir a transmissão da doença. [17].

Existem diferentes exames laboratoriais que podem ser utilizados para diagnosticar a tuberculose, tais como radiografia de tórax, testes de cultura, testes moleculares rápidos e baciloscopia de escarro (SSM). Historicamente, a baciloscopia de escarro tem sido o método mais comum para diagnóstico e monitoramento do tratamento da TB. Para realizar a baciloscopia de escarro, amostras de escarro (muco pulmonar ou saliva) são fixadas em lâminas e submetidas a técnicas de coloração que identificam os bacilos ácido-resistentes (AFB), que são as células da *Mtb*. [17]. Usualmente, a técnica de coloração utilizada na SSM é a de Ziehl-Neelsen, corando-se o AFB de vermelho e o fundo com azul metileno.

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 [8]. A análise de alguns slides pode ser difícil porque alguns componentes não relacionados às bactérias podem se assemelhar às células da *Mtb* [17]. Portanto, a precisão do resultado depende da habilidade e experiência do profissional especializado na análise das amostras [9]. Diante desse cenário, a automação da análise das imagens das lâminas para detecção e contagem dos bacilos pode ajudar os especialistas em bacteriologia no processo diagnóstico, tornando-o mais rápido, menos cansativo e reduzindo a possibilidade de erros.

Abordagens computacionais que utilizam análise de imagens têm sido desenvolvidas para auxiliar no diagnóstico da tuberculose, sendo que as técnicas baseadas em aprendizado profundo têm se destacado nessa área. Entretanto, a baixa qualidade das imagens microscópicas e a escassez de conjuntos de dados disponíveis ainda representam desafios para a detecção dos bacilos no processamento das imagens. [4]. Neste estudo, propomos a avaliação do desempenho do modelo EfficientDet e seus backbones na tarefa de detecção dos bacilos de Koch, testando diferentes esquemas de cores como pré-processamento das imagens. Dessa forma, buscamos avaliar o impacto da representação dos canais no realce dos bacilos.

A abordagem EfficientDet difere das técnicas tradicionais ao realizar a fusão de características considerando que cada uma delas apresenta contribuições distintas para a saída, dependendo da diferença de resoluções. Com a fusão multi-escala eficiente, o modelo é capaz de ajustar os pesos para aprender a importância das diferentes características de entrada. [15]. Este estudo tem como objetivo aplicar os benefícios da abordagem EfficientDet na detecção dos bacilos da TB, avaliando diferentes representações de cores e realizando uma análise robusta do método.

2. TRABALHOS RELACIONADOS

Existem diversos estudos na literatura voltados para a detecção da tuberculose, com enfoque no aprimoramento das imagens das lâminas coradas, na segmentação dos bacilos, na classificação das lâminas quanto à presença de bacilos da TB e na detecção dos bacilos em si. Nesta seção, apre-

sentaremos trabalhos recentes e relevantes relacionados aos métodos desenvolvidos com esse objetivo, e compararemos com o método proposto neste trabalho

Em [16], os autores propuseram um novo modelo estatístico para identificar bacilos, levando em consideração tanto aspectos de cor quanto de forma. Eles utilizaram um conjunto de dados composto por 43 imagens de SSM e aplicaram técnicas de pré-processamento para melhorar a qualidade das imagens antes de prosseguir para a etapa de segmentação. Após a segmentação, foram extraídas características de cor e forma das imagens para serem utilizadas como entrada para o classificador K-NN. Os resultados obtidos mostraram que o modelo apresentou uma precisão média de 82,7% na identificação de bacilos únicos e sobrepostos, e uma precisão de 99,1% na identificação de bacilos individuais em meio a bacilos sobrepostos e outros objetos.

O método de [10] também utilizado também envolve a utilização de técnicas de pré-processamento, segmentação e classificação dos bacilos presentes nas imagens. Para a segmentação, foi utilizado o algoritmo K-means para agrupar as imagens, e em seguida, foram avaliadas três redes neurais convolucionais pré-treinadas (VGG16, ResNet50 e SqueezeNet) para classificar as imagens segmentadas em bacilos ou células. O conjunto de dados utilizado para o experimento foi o ZN sputum smear microscopy image database (ZNSM-iDB), que contém aproximadamente 2000 conjuntos de imagens em diferentes colorações. Os resultados obtidos pelo método foram bastante satisfatórios, alcançando uma precisão geral de 97% utilizando a rede neural SqueezeNet.

Diferente dos trabalhos anteriores, o método de [7] utiliza um modelo de detecção de objetos chamado RetinaNet para explorar a detecção de bacilos da TB. O modelo foi treinado e testado com um conjunto de dados público que contém 928 imagens de SSM, o mesmo conjunto de dados utilizado neste estudo. Além disso, os autores aplicaram uma etapa de pré-processamento para melhorar a qualidade das imagens. Os resultados obtidos mostraram um mAP de 0,794, recall de 0,8656, precisão de 0,671 e f1-score de 0,7561.

O trabalho de [2] enfatiza a importância do pré-processamento das imagens de SSM, no qual os autores aplicaram técnicas de ajuste de brilho e contraste nas imagens antes de prosseguir para a etapa de segmentação dos bacilos. Para a segmentação, foram avaliados dois métodos: Artificial Neural Network (ANN) e classificador Bayesiano com modelo de mistura gaussiana (GMM). O trabalho alcançou uma precisão de 99,4%, um índice J de 99,7% e uma AUC de 0,9999. Além disso, os autores realizaram a aquisição das imagens utilizadas no estudo.

O trabalho de [18] propõe um pipeline composto por quatro estágios para detectar bacilos. Em primeiro lugar, os autores utilizam uma etapa de processamento de imagens para aprimorar as imagens das lâminas. Em seguida, as Generative Adversarial Networks (GANs) são aplicadas para segmentação semântica das lâminas. A partir das imagens segmentadas, são extraídos patches das lâminas que são classificados. Por fim, os autores usaram regressão para contar a quantidade de bactérias presentes em um patch. O conjunto de dados utilizado neste estudo contém informações de 46 pacientes ao longo de 6 meses. Os resultados obtidos pelos autores mostraram uma acurácia de 99,74%, recall de 0,967, precisão de 0,990 e f1-score de 0,960 na classificação dos patches e um MSE de 0,0054 na contagem das bactérias.

Em [1], explora a detecção de bacilos através de um ap-

rimoramento da YOLOv5, chamado pelos autores de DA-YOLO. Essa abordagem combina dois mecanismos de atenção para obter características mais robustas através do backbone da rede. O conjunto de dados utilizado para avaliar o método é composto por 1265 imagens de SSM, com 9969 marcações de bacilos. O método obteve uma mAP de 0,876, recall de 0,80 e precisão de 0,81.

Ao analisar os estudos descritos, fica evidente a importância do pré-processamento de imagens para a identificação de bacilos, o que é uma etapa essencial para o método proposto. Em relação aos demais trabalhos, os resultados na segmentação e detecção são promissores, mas muitos deles se baseiam em métodos clássicos, incluindo processamento de imagens. Embora esses métodos possam fornecer resultados relevantes, podem ser limitados pelas características específicas do conjunto de dados, dificultando a generalização da solução para outras imagens ou colorações.

Em termos gerais, os estudos analisados são focados em segmentação e detecção de bacilos, com uma ênfase maior em segmentação. Os métodos de detecção exploram modelos comumente utilizados para detecção de objetos, e empregam imagens similares às usadas neste estudo. No entanto, propomos a utilização de um modelo de detecção de objetos que equilibre eficiência e precisão, examinando diversas opções de backbones para obter a melhor representação das características. Além disso, buscamos otimizar a representação dos canais de imagem para o modelo.

3. MÉTODO PROPOSTO

Esta seção descreve as etapas realizadas neste trabalho para detectar os bacilos da TB em imagens de SSM. Em primeiro lugar, o dataset de imagens de SSM disponível publicamente foi utilizado para a aquisição das imagens. Em seguida, as imagens 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. Depois da normalização das imagens, experimentos com a EfficientDet e seus backbones foram realizados em uma etapa de treinamento e teste. Para avaliar o desempenho do método proposto, a validação cruzada com 5 folds e métricas amplamente aplicadas na literatura foram utilizadas. A Figura 1 ilustra o fluxo da metodologia apresentada e, a seguir, cada uma das etapas é descrita.

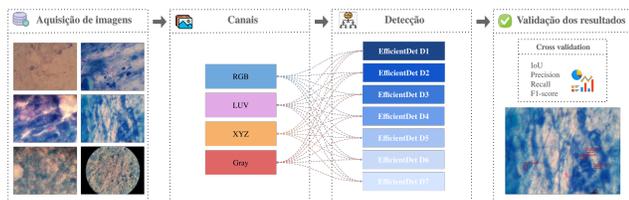


Figure 1: Representação da metodologia aplicada.

3.1 Aquisição de Imagens

Para a avaliação do método proposto neste trabalho, foram utilizadas imagens de SSM obtidas de um dataset público disponibilizado pelo AI Research and Automated Laboratory Diagnostics no Kaggle. [3]. O dataset utilizado neste

trabalho contém um total de 928 imagens com resolução de 1224x1623. Cada imagem vem acompanhada de um arquivo XML que contém as marcações dos bacilos no 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.

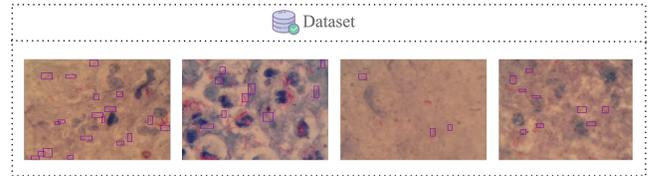


Figure 2: Exemplos de imagens do dataset com as marcações dos bacilos.

3.2 Pré-processamento

Neste trabalho, foi realizado um estudo para avaliar o impacto da representação de cor na detecção de bacilos em imagens de SSM. Foram utilizados quatro esquemas de cores tradicionais: RGB, HSV, LUV e XYZ, com o objetivo de identificar qual proporcionaria os melhores resultados ao modelo, destacando melhor as estruturas dos bacilos. Além disso, outras operações, como ajuste de brilho e equalização de histograma, foram aplicadas apenas no aumento de dados da rede, a fim de dificultar o treinamento do modelo e incentivá-lo a aprender diferentes padrões das imagens.

3.3 EfficientDet

A EfficientDet é um modelo de detecção de objetos proposto pelo Google Brain Team [13]. A arquitetura EfficientDet é caracterizada pelo uso de backbones e da BiFPN, juntamente com ajustes de otimização. Essa arquitetura permite uma fusão eficiente dos diferentes mapas de características multiescala, o que resulta em um modelo altamente preciso e eficiente em termos de processamento de dados. De fato, a EfficientDet superou modelos do estado da arte em termos de precisão e eficiência, tornando-se uma das arquiteturas mais populares para detecção de objetos. [13].

A EfficientDet é um modelo de detecção escalável e leve que possui oito backbones, desde o D0 até o D7. À medida que o tamanho do modelo aumenta, há um aumento na precisão, mas também na complexidade temporal. A EfficientNet é o modelo utilizado como backbone na EfficientDet. [12], que aplica convoluções separáveis para deixar o modelo mais leve [11]. Em geral, a EfficientNet, também proposta pelo Google Brain Team, alcança melhor precisão e eficiência quando comparado às CNNs existentes [5].

A arquitetura da EfficientDet emprega a BiFPN, que aprende pesos para penalizar as características com base em sua relevância para a saída. Essa abordagem permite a fusão eficiente das características multiescala, utilizando fluxo de baixo para cima e de cima para baixo. [6]. A arquitetura da EfficientDet inclui um método de escalonamento composto que aumenta simultaneamente a resolução, profundidade e largura. Esse método visa garantir que cada componente da arquitetura (backbone, rede de características e rede de classificação e localização de objetos) tenha um único fator de escala composto, que é controlado por regras heurísticas

que controlam todas as dimensões de escala. [13].

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

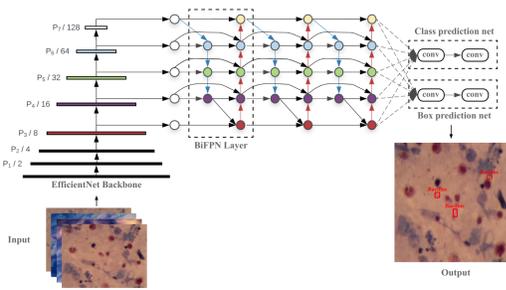


Figure 3: Arquitetura da EfficientDet. Adaptada de [13, 6].

3.4 Detalhes de Implementação

Após a aquisição das imagens, a primeira etapa realizada foi o pré-processamento, que consistiu na avaliação de quatro modelos de cor diferentes: RGB, HSV, LUV e XYZ. Para o treinamento, foram considerados sete backbones, nomeados de B1 a B7 de acordo com a nomenclatura da EfficientNet, correspondendo aos backbones D1 a D7x da EfficientDet. Cada backbone apresenta diferenças em termos de profundidade, dimensão de entrada e, conseqüentemente, precisão e eficiência, e foram treinados para cada modelo de cor avaliado.

Em cada experimento, foram utilizados alguns hiperparâmetros definidos empiricamente, a saber: $\gamma = 0.1$, $\text{batch size} = 4$ e número de épocas = 120. O tamanho da entrada foi definido como 512×512 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, além de avaliar os diferentes esquemas de cores, foi realizado um aumento de dados para melhorar a robustez e generalização do modelo. O aumento de dados foi realizado dinamicamente durante o treinamento, o que significa que as transformações foram aplicadas em tempo de execução. As transformações incluíram flip horizontal, flip vertical, ajuste aleatório de brilho e contraste, rotação aleatória em 90 graus, transposição, CLAHE, adição de ruído gaussiano e mudança aleatória de valores de matriz, saturação e valor.

Neste trabalho, o código foi implementado em Python utilizando a biblioteca PyTorch e executado 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

Optamos por avaliar a habilidade de generalização do modelo no conjunto de dados utilizado por meio de uma validação cruzada com $k = 5$. O conjunto de dados foi dividido aleatoriamente em 5 partições de tamanho semelhante. A validação cruzada foi executada para cada backbone em cada esquema de cor testado. Para avaliar o desempenho do

modelo, foram utilizadas as métricas de Intersection Over Union (IoU), recall, precisão e f1-score, que são calculadas 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 ??.

Table 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
LUV	IOU	0.742 ± 0.013	0.767 ± 0.009	0.662 ± 0.037	0.772 ± 0.015	0.673 ± 0.022	0.664 ± 0.057	0.697 ± 0.018
	Rec	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
	Prec	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
XYZ	IOU	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
	Rec	0.742 ± 0.066	0.708 ± 0.010	0.774 ± 0.015	0.675 ± 0.032	0.571 ± 0.211	0.505 ± 0.284	0.702 ± 0.009
	Prec	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
GRAY	IOU	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
	Rec	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	0.766 ± 0.014	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
GRAY	IOU	0.698 ± 0.010	0.701 ± 0.020	0.708 ± 0.021	0.574 ± 0.032	0.712 ± 0.030	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.

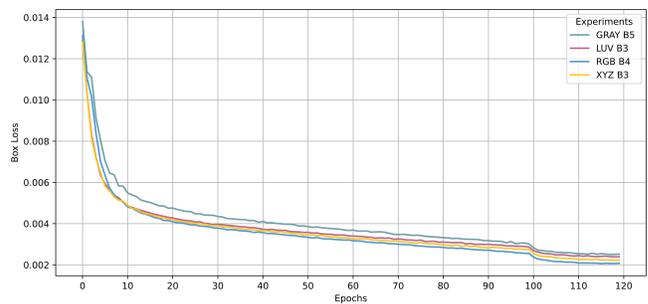


Figure 4: Gráfico de treinamento desempenho dos melhores resultados da Tabela 1.

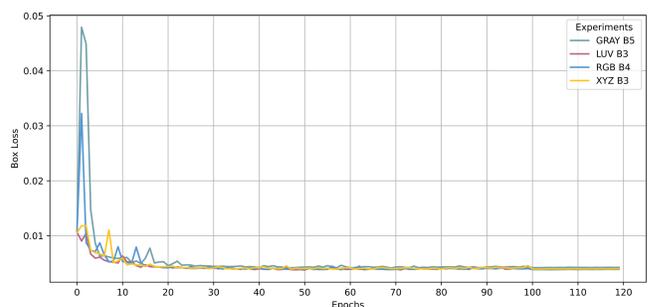


Figure 5: Gráfico de validação desempenho dos melhores resultados da Tabela 1.

Analisando a Tabela 1, observa-se que os resultados mais promissores encontram-se nas colunas que correspondem aos backbones B3 a B5. A fim de realizar essa comparação, foi utilizada a métrica f1-score, que é uma média harmônica entre precisão e recall, com o intuito de tornar a comparação mais confiável. Com relação aos diferentes esquemas de cores utilizados, o canal LUV apresentou o melhor de-

sempenho, mas houve um empate técnico em relação aos melhores resultados nos outros canais.

Ao observarmos os gráficos de desempenho dos experimentos que apresentaram os melhores resultados (Figura 4 e 5), é possível notar que o desempenho do treinamento condiz com os resultados de teste. O treinamento (Figura 4) terminou com uma diferença de loss 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 5), 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 nossos resultados, apresentamos nas Figuras 6 e 7, casos de sucesso e falhas na detecção dos bacilos pelo modelo apresentado neste trabalho. Para gerar as Figuras 6 and 7, utilizamos os modelos de cores RGB com o backbone 4. Em azul, marcamos 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)).

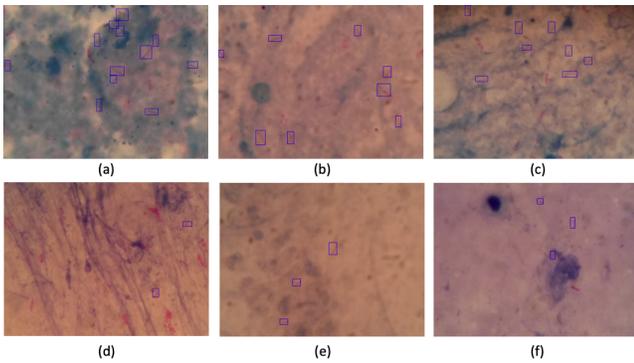


Figure 6: Casos de sucesso na detecção dos bacilos.

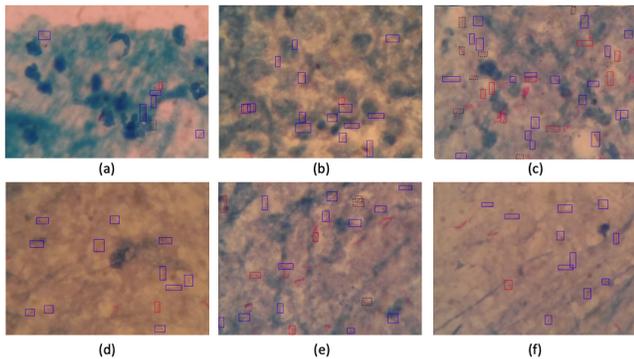


Figure 7: Casos de falha na detecção dos bacilos.

Como podemos observar na Figura 6, 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 7, 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, acreditamos 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 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 7, Acreditamos que a grande similaridade entre os bacilos pode levar o modelo a receber penalidades quando identifica erroneamente um bacilo que não possui uma marcação correspondente. Por esse motivo, a rede tenta aprender outras características, como orientação ou posição, para compensar a falta de aprendizado de características relevantes para o problema. Essa abordagem pode resultar em uma identificação insatisfatória de alguns bacilos que se encontram mais isolados ou que possuem características um pouco diferentes dos demais.

5. DISCUSSÕES

Nesta seção, faremos uma breve análise dos resultados obtidos, abordando alguns aspectos relevantes do método proposto e realizando comparações com estudos anteriores. Na Tabela 2, temos um resumo dos trabalhos relacionados e seus respectivos resultados.

Table 2: Comparação do método proposto com estudos relacionados.

Trabalho	Objetivo	Amostra	Conjunto de dados público	Métricas de validação
[16]	Segmentação e classificação dos bacilos usando um modelo estatístico de cor e forma	43	Não	Average accuracy = 0.827
[10]	Pré-processamento, segmentação e classificação dos bacilos usando K-means e 3 CNNs pré-treinadas	2000	Sim	Overall accuracy = 0.97
[7]	Pré-processamento + detecção dos bacilos da TB usando RetinaNet	928	Sim	mAP = 0.794 Recall = 0.8656 Precision = 0.671 F1-score = 0.7561
[2]	Pré-processamento + segmentação dos bacilos usando ANN e classificador Bayesiano com GMM	160	Não	Accuracy = 0.994 J. index = 0.997 AUC = 0.9999
[18]	Pré-processamento + segmentação de lâminas usando GAN + extração e classificação dos patches + regressão para contar a quantidade de bactérias em uma patch	500	Não informado	Accuracy = 0.9974 Recall = 0.967 Precision = 0.990 F1-score = 0.960 MSE = 0.0054
[1]	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 backbones	928	Sim	IoU = 0.523 Recall = 0.925 Precision = 0.694 F1-score = 0.774

Dos trabalhos apresentados na Tabela 2, apenas o de [7] utilizou o mesmo dataset do trabalho proposto, mas nosso método 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 promissores. Nosso 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 [1], que também utiliza uma rede de detecção em sua abordagem, nosso método alcançou resultados próximos, inclusive com um melhor recall.

Em geral, é possível perceber que o método alcançou re-

sultados promissores. A seguir, destacamos alguns pontos sobre o método proposto:

- Exploramos 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 todos os backbones da EfficientDet (D0 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;
- Utilizamos um dataset público de imagens, facilitando a reprodução e comparação dos resultados com outros métodos.

6. CONCLUSÃO

Neste estudo, descrevemos uma abordagem para a detecção de bacilos da TB utilizando a arquitetura EfficientDet, empregando diferentes backbones em nossos experimentos. Fizemos uso de um dataset público contendo imagens de SSM e exploramos 4 modelos de cores distintos durante o treinamento, que foram avaliados por meio de validação cruzada com 5 folds. Embora o problema apresente complexidade, devido à variação das imagens e dos bacilos em termos de características visuais e estruturais, nosso método obteve um desempenho satisfatório, superando alguns resultados da literatura relacionada. Assim, podemos afirmar que a EfficientDet possui potencial para a detecção de bacilos da TB e que os diferentes backbones avaliados 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 (013/2021 - 004/2020); CAPES - www.capes.gov.br; e CNPq - www.cnpq.br (305025/2021-0).

7. REFERENCES

- [1] L. An, K. Peng, X. Yang, P. Feng, and P. Huang. 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, 2022.
- [2] J. L. Díaz-Huerta, A. C. Téllez-Anguiano, J. A. Gutiérrez-Gnecchi, S. Arellano-Calderón, and J. C. Olivares-Rojas. Analysis and comparison of image processing and artificial intelligence algorithms to detect afb in pulmonary tuberculosis images. *Tuberculosis*, 134:102196, 2022.
- [3] Kaggle. Tuberculosis image dataset. Retrieved from <https://www.kaggle.com/saife245/tuberculosis-image-datasets>, 2020. Accessed November 13, 2022.
- [4] E. Kotei and R. Thirunavukarasu. Computational techniques for the automated detection of mycobacterium tuberculosis from digitized sputum smear microscopic images: A systematic review. *Progress in Biophysics and Molecular Biology*, 2022.
- [5] G. Marques, D. Agarwal, and I. de la Torre Díez. Automated medical diagnosis of covid-19 through efficientnet convolutional neural network. *Applied soft computing*, 96:106691, 2020.
- [6] M. L. Mekhalfi, C. Nicolò, Y. Bazi, M. M. Al Rahhal, N. A. Alsharif, and E. Al Maghayreh. Contrasting yolov5, transformer, and efficientdet detectors for crop circle detection in desert. *IEEE Geoscience and Remote Sensing Letters*, 19:1–5, 2021.
- [7] F. J. D. S. Reist, M. A. Veloso, F. M. M. Rodrigues, V. D. C. Brito, P. R. S. Dos Santos, J. D. L. Araújo, R. D. A. L. Rabelo, and A. O. De Carvalho Filho. 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, 2021.
- [8] M. Rico-Garcia, A. Salazar, C.-A. Madrigal, L.-J. Morantes-Guzman, and F. Cortes-Mancera. 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, 2015.
- [9] M. I. Shah, S. Mishra, M. Sarkar, and C. Rout. 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, 2016.
- [10] V. Shwetha, K. Prasad, C. Mukhopadhyay, B. Banerjee, and A. Chakrabarti. 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, 2021.

- [11] S. Song, J. Jing, Y. Huang, and M. Shi. Efficientdet for fabric defect detection based on edge computing. *Journal of Engineered Fibers and Fabrics*, 16:15589250211008346, 2021.
- [12] M. Tan and Q. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019.
- [13] M. Tan, R. Pang, and Q. V. Le. Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10781–10790, 2020.
- [14] W. H. O. WHO. Tuberculosis. Retrieved from <https://www.who.int/news-room/fact-sheets/detail/tuberculosis>, 2021. Accessed March 05, 2023.
- [15] M. Wu, C. Heng, H. Zhu, and H. Cai. 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, 2022.
- [16] H. Yousefi, F. Mohammadi, N. Mirian, and N. Amini. 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, 2020.
- [17] M. Zachariou, O. Arandjelović, W. Sabiiti, B. Mtafya, and D. Sloan. Tuberculosis bacteria detection and counting in fluorescence microscopy images using a multi-stage deep learning pipeline. *Information*, 13(2):96, 2022.
- [18] M. Zachariou, O. Arandjelović, W. Sabiiti, B. Mtafya, and D. Sloan. Tuberculosis bacteria detection and counting in fluorescence microscopy images using a multi-stage deep learning pipeline. *Information*, 13(2), 2022.