Combining CNN and K-Means for Automated Detection of Tuberculosis Bacilli in Bacilloscopy Images

Thales Francisco Mota Carvalho*, Vívian Ludimila Aguiar Santos+, Lida Jouca de Assis Figueredo†, Silvana Spíndola de Miranda‡, Ricardo de Oliveira Duarte∓, Frederico Gadelha Guimarães⋆
Institute of Engineering, Science and Technology, Federal U. of Vales do Jequitinhonha e Mucuri, Janaúba, Brazil
+Institute of Science and Technology, Federal University of Vales do Jequitinhonha e Mucuri, Diamantina, Brazil
†Central Public Health Laboratory of Minas Gerais, Ezequiel Dias Foundation, Belo Horizonte, Brazil
‡Faculty of Medicine, Federal University of Minas Gerais, Belo Horizonte, Brazil
∓Department of Electronics, Federal University of Minas Gerais, Belo Horizonte, Brazil
⋆Future Lab, Department of Computer Science, Federal University of Minas Gerais, Belo Horizonte, Brazil
Email: *thales.mota@ufvjm.edu.br

***Abstract*—Tuberculosis (TB) remains a major global health challenge, especially in low-resource settings where access to accurate diagnosis is limited. This study investigates the application of Convolutional Neural Networks (CNNs) for the automatic identification of Mycobacterium tuberculosis bacilli in microscopy images. Four CNN architectures from the literature were implemented and evaluated using multiple annotated datasets. Results from image fragment classification tasks showed high performance across all models, with accuracy, sensitivity, and specificity exceeding 95%. However, further experiments simulating full-field bacillus detection using a combination of K-means segmentation and CNN classification revealed a performance drop, notably in precision, highlighting a potential overestimation in controlled fragment-based evaluations. To address this, an enhanced training strategy was proposed by augmenting the dataset with more representative negative samples. This approach significantly improved model precision and F-measure, albeit with a slight reduction in sensitivity. The results suggest that, although CNNs are effective in fragment classification, their application in real-world detection scenarios requires careful evaluation, particularly regarding dataset construction and region of interest selection. The study emphasizes the need for robust and context-aware validation strategies for the deployment of AI tools in TB diagnosis.**

***Index Terms*—Tuberculosis, Deep Learning, CNN, Bacillus Detection, K-Means Segmentation, Bacilloscopy**

I. Introduction

Tuberculosis (TB) is an infectious disease caused by Mycobacterium tuberculosis (Mtb), transmitted primarily through airborne particles, and remains one of the leading causes of death by an infectious agent globally — surpassing even HIV/AIDS[1] [1]. According to the WHO Global Tuberculosis Report 2022, despite improvements in access to treatment, an estimated 10 million people developed TB in 2019. This number dropped to 5.8 million in 2020 due to the impact of the COVID-19 pandemic but rose again to 6.4 million in 2021, with 1.4 million reported deaths [1]. A significant gap persists between the estimated and diagnosed cases, largely due to underreporting and barriers to access, with countries such as India, Indonesia, the Philippines, Pakistan, and Nigeria accounting for 75% of this global discrepancy [1].

In Latin America, Brazil stands out as one of the countries with the highest TB burden and is among the few high-incidence countries with strong treatment coverage rates [2]. Combating TB necessarily involves strengthening diagnostic strategies, including the wider adoption of WHO-recommended methods. Currently, sputum smear microscopy (bacilloscopy) remains widely used in low-resource settings due to its low cost and simplicity, despite its limited sensitivity and heavy reliance on the examiner's expertise [2]. More advanced methods, such as the Molecular Rapid Test for TB (TRM-TB) and culture, offer improved diagnostic accuracy but face limitations in terms of cost, turnaround time, and laboratory infrastructure requirements [3].

Given these constraints, there is growing interest in using Artificial Intelligence (AI), particularly Machine Learning (ML) and Deep Learning (DL), to assist in TB diagnosis through automated image analysis. These approaches aim to increase diagnostic efficiency, reduce the burden on healthcare professionals, and enable large-scale screening. Various studies have demonstrated the successful application of these techniques across medical domains, including the use of Convolutional Neural Networks (CNNs) for detecting breast cancer [4], brain tumors [5], schizophrenia [6], epilepsy [7], and glaucoma [8].

In the specific context of TB, studies such as [9]–[14] have reported promising results in detecting Mtb bacilli in sputum smear images, using various CNN architectures and integrated strategies aimed at reducing false positives and improving accuracy and sensitivity. Additionally, [15] proposed CNN models optimized for dedicated hardware, expanding the feasibility of deploying these technologies in clinical environments with limited resources.

Given this context, the present study aims to evaluate the

[1]HIV stands for Human Immunodeficiency Virus, while AIDS stands for Acquired Immunodeficiency Dysndrome.

performance of CNN-based architectures for the automated detection of TB bacilli, testing their effectiveness across multiple datasets and exploring complementary strategies such as K-means-based image segmentation. The goal is to contribute computational solutions that support early TB diagnosis, particularly in settings with restricted laboratory infrastructure.

II. Materials and Methods

This section describes the datasets and procedures adopted to train and evaluate CNN models for the automated detection of TB bacilli in smear microscopy images. The proposed approach combines the use of multiple datasets from the literature, standardized image preprocessing, and the application of cross-validation and data augmentation techniques to ensure the robustness and reliability of the analyses.

A. Datasets

Four datasets were used to train and evaluate the deep learning (DL) methods explored in this study. Three of them are publicly available in the literature, namely those proposed by [10], [16], [17], while the fourth was more recently developed by [18]. These datasets are referred to throughout the paper as dataset-1, dataset-2, dataset-3, and dataset-4, respectively.

The selection was based on public availability or access through direct communication with the authors. Priority was given to datasets that included manual annotations indicating the locations of TB bacilli.

Dataset-1 consists of 120 annotated images containing true, uncertain, and clustered bacilli. Dataset-2 includes 1,265 images with annotated TB bacilli. Dataset-3 contains 300 images divided into three subgroups (MS-1, MS-2, and MS-3); however, only MS-1 and MS-2 were used in this study, following the approach adopted in previous works [14], [19], [20]. Dataset-4, developed by the authors of this work, comprises 200 meticulously annotated images.

For the purposes of training and testing, all different categories of bacilli (e.g., isolated, clustered, or occluded) were consolidated into a single class labeled "true bacillus."

1) Training and Testing Data for CNN: To train the CNN classification models, it was necessary to generate both positive fragments (containing annotated TB bacilli) and negative fragments (without bacilli). Since the datasets do not provide explicit annotations for negative regions, the adopted strategy consisted of extracting fragments of the same size that do not intersect with any annotated regions. Whenever possible, areas near the annotated bacilli were prioritized (Figure 1). This approach aims to reduce the likelihood of including unannotated bacilli, especially in images with a high density of bacilli.

The models were evaluated using 5-fold cross-validation, with 10% of the training set used for internal validation in each fold. To improve generalization, data augmentation was applied, generating seven new fragments for each original one through 90°, 180°, and 270° rotations, as well as horizontal flipping, as illustrated in Figure 2. This augmentation step was applied prior to the data splitting for cross-validation, increasing the diversity of examples exposed to the model.

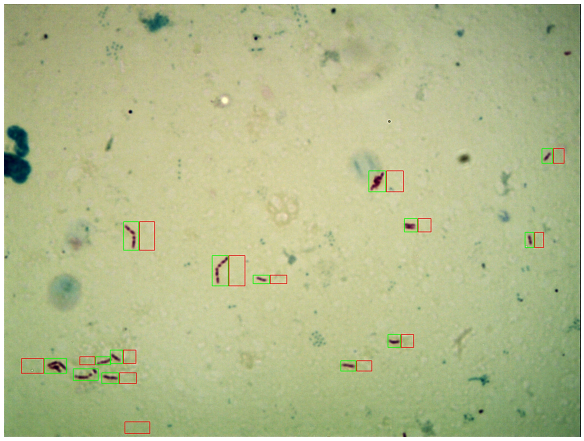

Fig. 1: Generation of negative fragments (red boxes) from the location and size of positive fragments (green boxes). Figure based on images from the dataset proposed by [17].

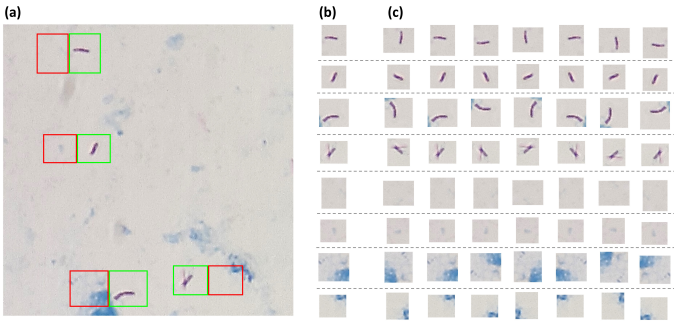

Fig. 2: Illustration of the data augmentation process: (a) shows the location of bacilli (positive class, green boxes) and non-bacilli (negative class, red boxes) in an image from dataset-4; (b) shows 8 fragments extracted from the original image; (c) shows the generation of 7 augmented versions for each fragment, resulting in 64 total images for training/testing.

TABLE I: Average number of positive (with bacilli)(+) and negative (without bacilli)(-) fragments per dataset, considering 5-fold cross-validation and data augmentation.

	dataset-1	dataset-2	dataset-3	dataset-4
Training	55506 (+) 55506 (-)	56739 (+) 56739 (-)	10149 (+) 10149 (-)	17934 (+) 17934 (-)
Validation	3330 (+) 3330 (-)	6397 (+) 6397 (-)	1314 (+) 1314 (-)	2194 (+) 2194 (-)
Testing	14709 (+) 14709 (-)	15784 (+) 15784 (-)	2866 (+) 2866 (-)	5032 (+) 5032 (-)

The average number of positive and negative fragments per dataset, after applying data augmentation and 5-fold cross-validation, is detailed in Table I. These values reflect the variation in bacillus density across datasets and directly influence the number of fragments generated per image.

B. Replication of CNN Architectures Proposed in the Literature

In this stage of the study, CNN architectures previously described in the literature were replicated with the goal of

evaluating their performance in detecting TB bacilli from image fragments. The replication was based on the systematic review conducted by [21], with modifications to the original implementation. Specifically, a five-fold cross-validation strategy was adopted in place of the original three-way data split, aiming to enhance the reliability and generalizability of the results.

1) Selected Studies: Four CNN architectures were selected from well-established studies in the field: [9]–[11], [22]. The selection was based on strict criteria to ensure experimental feasibility and reproducibility under equal conditions. The selected architectures met the following requirements: (i) complete specifications of the CNN layers were provided, enabling accurate or near-accurate replication; (ii) compatibility with TensorFlow v.2.4.1 and Keras v.2.4, ensuring a standardized implementation environment; (iii) no use of hybrid methods, which simplifies replication and comparison; and (iv) no use of pre-trained weights, avoiding external bias in model evaluation.

These criteria were applied to guarantee that the experiments would be reproducible, computationally feasible, and fairly comparable. As the original source codes were not publicly available, the implementations were carried out based on the architectural descriptions provided in the publications, using TensorFlow resources to closely replicate the original designs.

2) Implementation and Evaluation: Each author proposed distinct training strategies and parameter configurations. To ensure a fair comparison across architectures, a unified training protocol was defined as follows: (i) number of epochs set to 100; (ii) batch size of 25; (iii) Stochastic Gradient Descent (SGD) optimizer with a learning rate of 0.001 and momentum of 0.9; (iv) steps per epoch equal to the size of the training data; and (v) a fixed seed value of 42 to shuffle image order during training. Default TensorFlow parameters were used for all other configurations. Input images were resized to the dimensions specified in each original study before training.

The models were implemented in Python v.3.8.5 using TensorFlow v.2.4.1, Keras v.2.4.0, and Scikit-learn v.1.0.2, among other supporting libraries, all installed via the Anaconda distribution.

To evaluate the CNN architectures, a 5-fold cross-validation procedure was applied as described in Section II-A1. After training, model performance was assessed using standard classification metrics: accuracy, sensitivity, specificity, precision, F-measure, and the Area Under the ROC Curve (AUC). Each architecture was tested on all available datasets.

C. Bacillus Detection Using K-Means and CNN

Although CNNs have shown high accuracy in classifying image patches containing tuberculosis (TB) bacilli, these models are not directly applicable to whole-slide images representing the complete microscopic field of view. For practical deployment in real-world settings, a preliminary segmentation step is required to identify potential Regions of Interest (ROIs) where bacilli may be present.

In this context, the K-Means algorithm offers an efficient strategy for segmenting images based on pixel similarity, enabling the isolation of areas with a higher likelihood of containing bacilli. By integrating K-Means with CNNs, the approach aims to enhance detection performance by focusing the model's attention on more relevant image regions, thereby reducing background noise and irrelevant features.

This section presents the implementation and evaluation of this hybrid approach, combining K-Means segmentation with CNN-based classification. The objective is to assess the applicability of CNNs in the context of whole microscopy images, bringing the methodology closer to real-world clinical scenarios.

1) Segmentation and Classification Process: In this stage, the K-Means algorithm is employed as an image segmentation technique, leveraging its ability to cluster pixels based on chromatic features. This approach is particularly effective for images stained using the Ziehl-Neelsen (ZN) method, where TB bacilli appear as bright red structures against a bluish background, facilitating the distinction between potentially relevant regions and non-informative areas.

Based on this, we propose a two-phase detection method: image segmentation using K-Means, followed by classification of the segmented regions using a CNN. The full workflow is illustrated in Figure 3, which details how both methods are integrated for automated TB bacillus detection. The process is outlined in the following steps:

- Color Enhancement: The original image (Figure 3a) undergoes saturation boosting to accentuate the bacillus staining (Figure 3b).
- Color Filtering in HSV Space: A color filter is applied in the Hue-Saturation-Value (HSV) color space to isolate reddish/pinkish/purplish tones typical of bacilli (Figure 3c), based on manually defined ranges (Figure 4).
- K-Means Segmentation: K-Means clustering (OpenCV library) is applied with $K = 5$ to segment the image into five dominant color regions (centroids) in the Red-Green-Blue (RGB) space (Figure3d).
- Image Generation per Centroid: For each centroid: (i) binarization is performed using the centroid RGB value as a threshold (Figure 3e); (ii) object boundaries are detected using OpenCV methods including Canny, dilate, and findContours (Figure 3f).
- ROI Extraction: Contours are used to extract image patches ROIs using findContours in OpenCV (Figure 3h).
- Size Filtering: Patches larger than 0.5% of the image area are discarded (Figure 3i).
- CNN Classification: Remaining patches are classified as "bacillus" or "non-bacillus" by a pre-trained CNN (Figure 3k).
- Overlap Removal: Non-Maximum Suppression is applied to avoid multiple detections of the same bacillus (Figure 3l).
- Final Marking: The coordinates of patches classified as bacilli are mapped onto the original microscopy image.

Figure 5 summarizes the detection and classification process, with the following visual steps: (a) input image representing a microscopy field of view; (b) image after saturation enhancement and color filtering, highlighting the bacilli; (c, d, e) segmented outputs from K-Means with $K = 2$ and subsequent binarization using the centroid threshold; (f) object detection using edge and contour methods with red bounding boxes; (g) filtered ROIs extracted from previous steps; (h) classification of patches by CNN and merging of overlapping boxes; (i) final output showing detected bacilli highlighted in green bounding boxes on the original image.

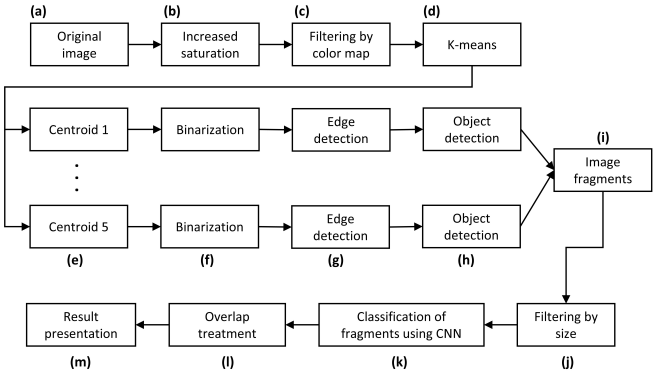

Fig. 3: Proposed method for TB bacillus detection using K-Means for image segmentation and CNN for classification.

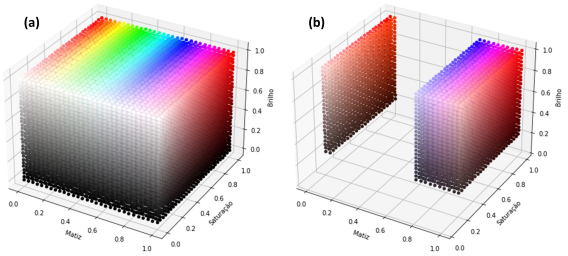

Fig. 4: Illustration of the HSV color space, where: (a) shows the HSV model representation, and (b) indicates the color range used for filtering, as applied in Figure 3c.

2) Training Data Enhancement: During preliminary experiments involving K-Means and CNNs, an overestimation in classification performance was observed, primarily attributed to the underrepresentation of the negative class (non-bacillus) in the training data. To address this bias, a training data enhancement strategy was proposed, which consists of identifying and expanding representative negative patterns, followed by retraining the models. A detailed analysis of this behavior and the improvements achieved is provided in Subsections III-B and III-C.

The strategy for enhancing the training data and improving CNN generalization involves four main steps, as illustrated in Figure 6:

- Initial training: The dataset is split into training/validation and test sets (Figure 6a). Image fragments from the training/validation set are extracted (Figure 6b) and used to train the initial CNN model (Figures 6c and 6d), with binary classification into "bacillus" and "non-bacillus," as described in Subsection II-A1.
- False Positive identification: the images from the training/validation set (Figure 6e) are segmented using K-Means (Figure 6f), generating ROIs (Figure 6g), which are then classified by the CNN model (Figure 6h). Fragments incorrectly classified as bacilli (False Positives) are identified and stored (Figure 6i).
- Retraining (refinement): the identified False Positives (Figure 6i) are combined with the original training data (Figure 6b) to form a new training/validation set (Figure 6j). A new model is then trained from this dataset (Figure 6l), resulting in the refined CNN model ("CNN* model") (Figure 6m).
- Final evaluation: the refined "CNN* model" is evaluated using the test set (Figures 6a and 6n). Images are segmented via K-Means (Figure 6o), ROIs are extracted (Figure 6p), and classified by the "CNN* model" (Figure 6q). The results are then compared with ground-truth annotations, and performance metrics are computed (Figure 6r).

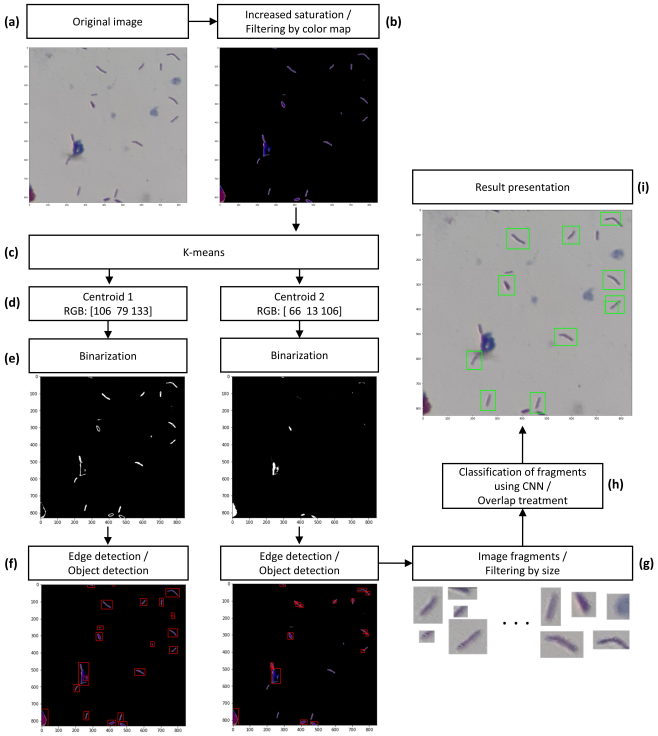

Fig. 5: Graphical representation highlighting selected output images from each stage of the proposed method.

Through this proposed workflow, it becomes possible to expand the datasets, making them more representative and, consequently, enhancing the performance of CNNs in the task of TB bacillus detection.

3) Implementation and Evaluation: In this stage, the value of $K = 5$ was adopted for the K-Means algorithm, based on empirical tests that indicated the best trade-off between

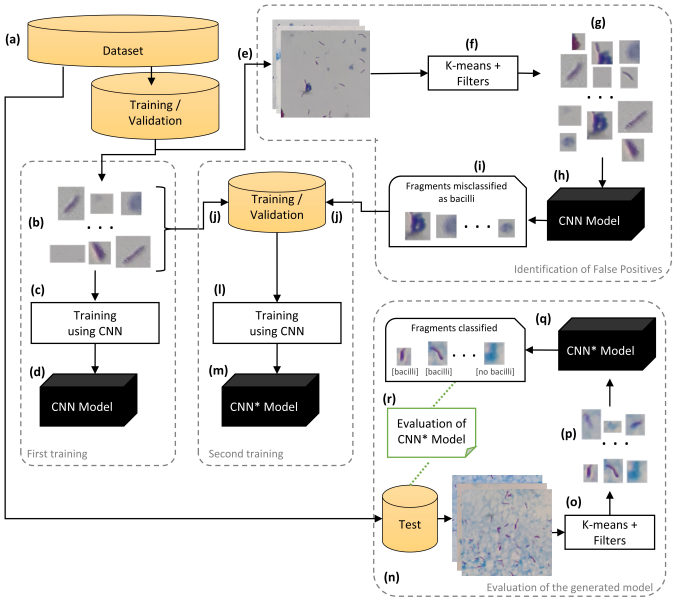

Fig. 6: Illustrative diagram of the strategy employed to expand a dataset in order to make it more representative, consisting of four stages aimed at improving the generalization ability of CNN-based models.

performance and computational cost. The image fragments generated by the segmentation process were subsequently classified by a CNN trained to detect the presence of TB bacilli.

The CNN architecture selected for this integration corresponds to the one that demonstrated the best overall performance across the four datasets, as previously described in Subsection II-B.

The implementation was carried out using Python v.3.8.5, with libraries including TensorFlow v.2.4.1, Keras v.2.4.0, Scikit-learn v.1.0.2, Pillow v.9.0.1, and OpenCV v.4.5.4.

To evaluate the combined K-Means and CNN approach, a 5-fold cross-validation procedure was performed, as described in Subsection II-A1. After training, model performance was assessed using the following metrics: accuracy, sensitivity, specificity, precision, F-measure, and AUC. Each method was tested on all available datasets.

Similarly, to assess the K-Means approach in conjunction with the enhanced CNN training strategy, a 5-fold cross-validation procedure was also applied. However, in this case, the training/validation datasets described in Subsection II-A1 were expanded as outlined in Subsection II-C2. The results of these evaluations are presented in Sections III-B and III-C.

III. Results and Discussion

This section presents and discusses the results obtained from the experiments conducted to evaluate DL strategies for the identification of Mycobacterium tuberculosis bacilli.

Experiment 1 (Subsection III-A): evaluates the performance of various CNN architectures replicated from the literature, applied to multiple annotated datasets. Experiment 2 (Subsection III-B): investigates the effectiveness of combining K-Means segmentation with CNN classification for bacilli detection directly on full smear microscopy images. Experiment 3 (Subsection III-C): explores the impact of enhancing the training dataset by incorporating false positives, aiming to improve the generalization capability of the CNN models.

All experiments were executed in a mid-range computing environment equipped with an AMD Ryzen 9 5900X processor (12 cores / 24 threads), 64 GB of RAM, and an Nvidia GeForce RTX 3060 GPU (12 GB RAM), ensuring adequate stability and performance to support the computational demands of the applied DL methods.

A. Experiment 1: Evaluation of CNN Models Across Multiple Datasets

In this experiment, four CNN architectures [9]–[11], [22] were evaluated, as defined in Subsection II-B. For simplicity, these models were referred to as CNN-A, CNN-B, CNN-C, and CNN-D, respectively. They were applied to different datasets to classify image fragments containing Mycobacterium tuberculosis bacilli. The goal was to identify the architecture with the best overall performance and to analyze the impact of dataset heterogeneity on model behavior.

The experiment was subdivided into four parts (1.1 through 1.4), each corresponding to one of the datasets described in Section II-A. A 5-fold cross-validation approach was adopted to enhance result reliability and mitigate overfitting risks. This structure enabled a thorough evaluation of each model's robustness, consistency, and generalization capacity across different contexts—essential for selecting the most appropriate architecture for subsequent experiments.

In Experiment 1.1, dataset-1 was used with 5-fold cross-validation to assess model performance. Due to the large volume of image fragments, the total execution time was approximately 71 hours. Average evaluation metrics over the five folds are shown in Table II. The results were consistent with those reported in the literature, even with adjustments in parameters and data. Differences in accuracy and AUC were around 0.03 compared to the original studies [9]–[11], [22]. Among the models tested, CNN-B showed the best performance, likely due to its original design and tuning based on the same dataset [16]. Nevertheless, all architectures achieved values above 0.95 across all metrics, indicating high and consistent performance.

Experiment 1.2 used dataset-2, following the same methodology as Experiment 1.1. The total training and testing process took about 80 hours. The average results are presented in Table III and show strong consistency with original studies, with accuracy and AUC variations of approximately 0.02 [9]–[11], [22]. No single architecture stood out as clearly superior, although CNN-B and CNN-D performed similarly, with comparable accuracy, F-measure, and AUC. CNN-D had a slight edge in specificity and precision. Overall, all CNNs demonstrated strong generalization ability, with scores above 0.96 across all evaluated metrics.

In Experiment 1.3, dataset-3 was employed, and the CNN training and testing followed the same protocol. The total execution time was approximately 13 hours. CNN-D achieved the best overall performance, surpassing other architectures across nearly all metrics, as shown in Table IV. All CNNs reached values above 0.97, in line with those reported in prior work. The superior results from dataset-3 may be attributed to: (i) higher image acquisition quality (sample type, camera, microscope); (ii) a larger number of fragments used in training/testing; and (iii) a more rigorous annotation method.

In Experiment 1.4, dataset-4 was used, following the same methodological framework. The total execution time was around 20 hours. According to Table V, CNN-D again exhibited the best overall performance among the tested architectures. This superiority may be due to its automated optimization design, which tailors its layers to the specific task of TB bacilli classification. All CNNs, however, performed well, with scores above 0.90 across all metrics, confirming their effectiveness even on a novel dataset.

In summary, all CNN models demonstrated strong performance, with high sensitivity and specificity—critical traits for medical diagnostic applications. The results were consistent with those from the original studies, reinforcing the robustness and reproducibility of the implemented models. However, to comprehensively and accurately assess the CNNs' ability to detect TB bacilli, it is necessary to go beyond isolated fragment classification and conduct experiments focusing on full-image detection, as explored in subsequent sections.

TABLE II: Performance per CNN architectures on dataset-1.

Metric	CNN-A	CNN-B	CNN-C	CNN-D
Accuracy	0.9664 (±0.0067)	**0.9740** (±0.0075)	0.9554 (±0.0053)	0.9666 (±0.0081)
Sensitivity	0.9607 (±0.0183)	0.9725 (±0.0136)	0.9573 (±0.0184)	**0.9741** (±0.0154)
Specificity	0.9721 (±0.0097)	**0.9756** (±0.0080)	0.9536 (±0.0133)	0.9592 (±0.0161)
Precision	0.9719 (±0.0094)	**0.9754** (±0.0080)	0.9540 (±0.0118)	0.9600 (±0.0148)
F-measure	0.9661 (±0.0072)	**0.9739** (±0.0077)	0.9554 (±0.0057)	0.9668 (±0.0081)
AUC	0.9664 (±0.0067)	**0.9740** (±0.0075)	0.9555 (±0.0053)	0.9666 (±0.0081)

TABLE III: Performance per CNN architectures on dataset-2.

Metric	CNN-A	CNN-B	CNN-C	CNN-D
Accuracy	0.9789 (±0.0022)	**0.9790** (±0.0029)	0.9709 (±0.0034)	**0.9790** (±0.0036)
Sensitivity	0.9782 (±0.0037)	**0.9830** (±0.0017)	0.9645 (±0.0081)	0.9796 (±0.0030)
Specificity	**0.9797** (±0.0049)	0.9749 (±0.0069)	0.9774 (±0.0039)	0.9783 (±0.0070)
Precision	**0.9796** (±0.0048)	0.9751 (±0.0066)	0.9771 (±0.0037)	0.9784 (±0.0068)
F-measure	0.9789 (±0.0022)	**0.9790** (±0.0028)	0.9707 (±0.0036)	**0.9790** (±0.0036)
AUC	0.9789 (±0.0022)	**0.9790** (±0.0029)	0.9709 (±0.0034)	**0.9790** (±0.0036)

TABLE IV: Performance per CNN architectures on dataset-3.

Metric	CNN-A	CNN-B	CNN-C	CNN-D
Accuracy	0.9896 (±0.0031)	0.9821 (±0.0106)	0.9851 (±0.0044)	**0.9905** (±0.0031)
Sensitivity	0.9888 (±0.0031)	**0.9907** (±0.0031)	0.9857 (±0.0032)	0.9903 (±0.0013)
Specificity	0.9904 (±0.0073)	0.9736 (±0.0232)	0.9846 (±0.0074)	**0.9906** (±0.0055)
Precision	0.9905 (±0.0071)	0.9745 (±0.0216)	0.9846 (±0.0074)	**0.9906** (±0.0055)
F-measure	0.9896 (±0.0031)	0.9824 (±0.0101)	0.9851 (±0.0043)	**0.9905** (±0.0031)
AUC	0.9896 (±0.0032)	0.9821 (±0.0106)	0.9851 (±0.0044)	**0.9905** (±0.0031)

TABLE V: Performance per CNN architectures on dataset-4.

Metric	CNN-A	CNN-B	CNN-C	CNN-D
Accuracy	0.9525 (±0.0057)	0.9208 (±0.0445)	0.9452 (±0.0096)	**0.9665** (±0.0039)
Sensitivity	0.9603 (±0.0205)	0.9077 (±0.1018)	0.9568 (±0.0137)	**0.9664** (±0.0098)
Specificity	0.9448 (±0.0220)	0.9339 (±0.0280)	0.9337 (±0.0216)	**0.9667** (±0.0149)
Precision	0.9463 (±0.0191)	0.9335 (±0.0221)	0.9356 (±0.0195)	**0.9669** (±0.0139)
F-measure	0.9528 (±0.0055)	0.9166 (±0.0538)	0.9458 (±0.0091)	**0.9665** (±0.0036)
AUC	0.9525 (±0.0057)	0.9208 (±0.0445)	0.9452 (±0.0096)	**0.9665** (±0.0039)

B. Experiment 2: TB Bacilli Detection Using K-Means and CNN

Although the CNNs demonstrated excellent performance in classifying isolated fragments (as observed in Experiment 1), it is essential to evaluate their effectiveness in a more realistic scenario—one that involves not only classification, but also the actual detection of bacilli in full microscopic slide images. To address this, Experiment 2 proposes an integrated approach: the K-Means algorithm is employed to segment the images and identify ROIs, which are then classified by a CNN.

The methodology adopted follows the procedures detailed in Subsection II-C. Performance evaluation was conducted using 5-fold cross-validation, employing the same data splits used in Experiments 1.1 through 1.4. This ensured consistency and comparability of the results.

Due to computational time constraints and the specific focus of this study, Experiment 2 was conducted exclusively with the CNN-D architecture proposed by [22]. This choice is justified by the superior performance of CNN-D in the previous experiments, as well as its original design, which includes optimizations tailored for bacilli detection in microscopy images.

The results of Experiment 2, summarized in Table VI, indicate suboptimal performance when combining K-Means with CNN-D for bacilli detection. The average metrics obtained were: precision = 0.4261, sensitivity = 0.8800, F-measure = 0.5658, and average precision (AP) = 0.4670. While the high sensitivity suggests that K-Means effectively identified most bacilli-containing regions, the low precision highlights the CNN's difficulty in distinguishing true bacilli from false

positives, leading to a high misclassification rate. As the previously trained models and data splits were reused, the total execution time for this experiment was approximately 4 hours.

It is important to note that the evaluation metrics used for detection differ from those used for classification. In detection scenarios, computing the number of True Negatives (TNs) is impractical. Background fragments that are not classified as bacilli could be incorrectly interpreted as TNs, potentially inflating the metric values and impairing their interpretability. Therefore, precision, sensitivity, F-measure, and AP were used for evaluating detection performance in Experiment 2.

The results of Experiment 2 indicate a potential overestimation of the CNN's performance, as the effectiveness observed in the classification of isolated fragments did not carry over to the more complex task of bacilli detection in full microscopy images. This discrepancy may be attributed to differences in the nature of the data or the way the datasets were constructed. While classification occurs in a more controlled and idealized environment, detection requires the model to operate under conditions that more closely resemble real-world clinical scenarios. These findings underscore the importance of validating models across diverse contexts to ensure that their performance remains robust and reliable in practical applications.

TABLE VI: Performance per dataset when using the K-Means + CNN-D approach for bacilli detection.

Metric	dataset-1	dataset-2	dataset-3	dataset-4
Precision	0.3323 (±0.1244)	0.3896 (±0.0391)	0.4849 (±0.0487)	0.4975 (±0.0527)
Sensitivity	0.9576 (±0.0047)	0.7866 (±0.0153)	0.9178 (±0.0062)	0.8578 (±0.0398)
F-measure	0.4808 (±0.1334)	0.5200 (±0.0359)	0.6330 (±0.0405)	0.6292 (±0.0524)
AP	0.5031 (±0.1105)	0.3708 (±0.0236)	0.5171 (±0.0353)	0.4770 (±0.0619)

C. Experiment 3: CNN Enhancement Using Expanded Negative Patterns

Given the limitations observed in the detection tests, particularly those related to the composition of negative fragments, Experiment 3 was conducted to evaluate the impact of enhancing the training data, as proposed in Subsection II-C2. The strategy consists of expanding the training/validation set with more representative negative patterns, aiming to improve the model's precision and robustness in the detection task.

Table VII presents the number of additional negative fragments incorporated into each dataset following the expansion process. It is important to note that the test sets were kept unchanged, thus preserving their integrity and ensuring complete independence from the training data.

The CNN-D architecture was retrained using the augmented training/validation sets that included additional negative patterns, resulting in a new model referred to as CNN-D*. The training procedure followed the same 5-fold cross-validation approach adopted in Experiment 1, ensuring consistency and comparability. The average performance metrics of CNN-D* across each dataset are presented in Table IX. Subsequently, Experiment 2 was repeated using CNN-D* in conjunction with K-Means to detect bacilli in full field microscopy images. The total runtime was approximately 72 hours for training and 4 hours for the detection phase. The updated detection results using CNN-D* are shown in Table VIII.

TABLE VII: Average quantities of expanded positive (+) and negative (–) fragments for each dataset.

	dataset-1	dataset-2	dataset-3	dataset-4
Training	55.506 (+) 192.880 (-)	56.739 (+) 136.559 (-)	10.149 (+) 20.740 (-)	17.934 (+) 34.655 (-)
Validation	3.330 (+) 12.764 (-)	6.397 (+) 16.124 (-)	1.314 (+) 2.749 (-)	2.194 (+) 3.795 (-)
Testing	14.709 (+) 14.709 (-)	15.784 (+) 15.756 (-)	2.866 (+) 2.866 (-)	5.032 (+) 5.032 (-)

The analysis of the results demonstrated an overall improvement in method performance following the enhancement of the training data. The average values for precision, sensitivity, F-measure, and AP were 0.7844, 0.7024, 0.7198, and 0.6124, respectively. When comparing the values in Tables VI and VIII, a significant increase in precision and a decrease in sensitivity are observed. Nonetheless, the rise in the F-measure suggests an improvement in the balance between precision and sensitivity, reflecting a more robust overall performance. This behavior is expected: as the model becomes more stringent in its detections (higher precision), it may fail to identify some bacilli (lower sensitivity), which represents a common trade-off in detection tasks.

The findings further support the hypothesis of model overestimation in CNNs, largely attributed to the underrepresentation of the negative class in the training data. Additionally, the potential presence of unannotated bacilli may have adversely affected precision, as these regions were likely misclassified as false positives during evaluation. This scenario underscores the importance of assessing models in broader and more realistic settings that include a larger volume of image fragments, as achieved through the K-Means-based approach. Although more selective ROI extraction methods may obscure such overestimation, they can pose risks when deployed in real-world TB diagnostic environments.

TABLE VIII: Results for each dataset using the K-means + CNN-D* combination.

Metric	dataset-1	dataset-2	dataset-3	dataset-4
Precision	0.7783 (±0.0783)	0.7269 (±0.0642)	0.8202 (±0.0632)	0.8121 (±0.0381)
Sensitivity	0.9168 (±0.0124)	0.3765 (±0.2188)	0.7582 (±0.0401)	0.7579 (±0.0358)
F-measure	0.8394 (±0.0422)	0.4710 (±0.1751)	0.7859 (±0.0355)	0.7830 (±0.0234)
AP	0.7868 (±0.0456)	0.3116 (±0.2112)	0.6812 (±0.0414)	0.6699 (±0.0362)

IV. Conclusion

This study investigated the use of CNNs for the identification of TB bacilli in bacilloscopy images, evaluating their performance in both isolated fragment classification and a more realistic detection scenario involving complete slide images, supported by the K-Means segmentation algorithm.

TABLE IX: Results for each dataset using the CNN-D with an expanded number of negative fragments (CNN-D*).

Metric	dataset-1	dataset-2	dataset-3	dataset-4
Accuracy	0.9668 (±0.0064)	0.9347 (±0.0148)	0.9608 (±0.0111)	0.9437 (±0.0053)
Sensitivity	0.9633 (±0.0207)	0.9938 (±0.0019)	0.9940 (±0.0011)	0.9706 (±0.0084)
Specificity	0.9702 (±0.0122)	0.8758 (±0.0300)	0.9277 (±0.0219)	0.9169 (±0.0124)
Precision	0.9702 (±0.0115)	0.8893 (±0.0241)	0.9324 (±0.0192)	0.9211 (±0.0110)
F-measure	0.9665 (±0.0069)	0.9385 (±0.0132)	0.9621 (±0.0104)	0.9451 (±0.0051)
AUC	0.9668 (±0.0064)	0.9348 (±0.0147)	0.9608 (±0.0110)	0.9437 (±0.0052)

The results from Experiment 1 confirmed the robustness and high accuracy of CNNs in the classification task, with all evaluated architectures achieving metrics above 0.95. Among them, CNN-D stood out as the most effective model and was selected for the subsequent experiments.

In Experiment 2, when applying the trained models in a detection scenario, a significant drop in precision was observed. This revealed a potential overestimation of model performance when evaluated solely on pre-defined fragments. Such findings highlight the limitations of relying only on classification metrics in controlled environments.

Experiment 3 addressed this limitation by enhancing the training data with more representative negative fragments. This strategy led to a notable improvement in overall performance, reflected by increased F-measure and precision. Although there was a slight decrease in sensitivity, this is a natural trade-off due to the model's more stringent detection criteria.

Overall, the results indicate that while CNNs are promising tools to support TB diagnosis, their practical application requires evaluation in broader and more realistic scenarios, taking into account segmentation challenges, data representativeness, and the presence of unannotated bacilli. This work contributes to the understanding of the challenges and limitations associated with using deep learning in medical imaging, emphasizing the need for comprehensive validation strategies grounded in real-world applications.

ACKNOWLEDGMENTS

The authors thank the institutions Federal University of Vales do Jequitinhonha e Mucuri and Federal University of Minas Gerias for the support and incentive in the development of this work. This work has been supported by the Brazilian agencies (i) National Council for Scientific and Technological Development (CNPq), Grant no. 304856/2025-8, "*Aprendizado de Máquina Colaborativo e Proteção à Privacidade*"; (ii) Coordination for the Improvement of Higher Education Personnel (CAPES) through the Academic Excellence Program (PROEX).

REFERENCES

[1] WHO, *Global tuberculosis report 2022*. Licence: CC BY-NC-SA 3.0 IGO: Geneva: World Health Organization, 2022.

[2] Brasil, "Manual de recomendações para o controle da tuberculose no brasil," *Ministério da Saúde. Secretaria de Vigilância em Saúde. Departamento de Vigilância das Doenças Transmissíveis, Brasilia, DF, Brasil.*, 2022.

[3] WHO, *Global tuberculosis report 2020*. Licence: CC BY-NC-SA 3.0 IGO: Geneva: World Health Organization, 2020.

[4] C. B. Gonçalves *et al.*, "Detecção de câncer de mama utilizando imagens termográficas," 2017.

[5] C. Neto and S. A. de La Hidalga, "Reconhecimento de tumores cerebrais utilizando redes neurais convolucionais," 2017.

[6] R. F. Vergara, "Detecção de alterações cerebrais anatômicas associadas à esquizofrenia com base em redes convolucionais aplicadas a imagens de ressonância magnética," 2018.

[7] J. M. S. C. F. Silva *et al.*, "Deteção de convulsões epiléticas em eletroencefalogramas usando deep learning," Ph.D. dissertation, 2017.

[8] A. C. d. M. LIMA *et al.*, "Aprendizagem profunda aplicada ao diagnóstico do glaucoma," 2019.

[9] R. O. Panicker, K. S. Kalmady, J. Rajan, and M. Sabu, "Automatic detection of tuberculosis bacilli from microscopic sputum smear images using deep learning methods," *Biocybernetics and Biomedical Engineering*, vol. 38, no. 3, pp. 691–699, 2018.

[10] J. A. Quinn, R. Nakasi, P. K. Mugagga, P. Byanyima, W. Lubega, and A. Andama, "Deep convolutional neural networks for microscopy-based point of care diagnostics," in *Machine Learning for Healthcare Conference*. PMLR, 2016, pp. 271–281.

[11] A. Simon, R. Vinayakumar, V. Sowmya, and K. Soman, "Shallow cnn with lstm layer for tuberculosis detection in microscopic images," *Machine learning for Biomedical Applications*, 2019.

[12] K. Mithra and W. S. Emmanuel, "Automated identification of mycobacterium bacillus from sputum images for tuberculosis diagnosis," *Signal, Image and Video Processing*, vol. 13, no. 8, pp. 1585–1592, 2019.

[13] C.-P. Kuok, M.-H. Horng, Y.-M. Liao, N.-H. Chow, and Y.-N. Sun, "An effective and accurate identification system of mycobacterium tuberculosis using convolution neural networks," *Microscopy research and technique*, vol. 82, no. 6, pp. 709–719, 2019.

[14] M. El-Melegy, D. Mohamed, and T. ElMelegy, "Automatic detection of tuberculosis bacilli from microscopic sputum smear images using faster r-cnn, transfer learning and augmentation," in *Iberian Conference on Pattern Recognition and Image Analysis*. Springer, 2019, pp. 270–278.

[15] K. Sethi, V. Parmar, and M. Suri, "Low-power hardware-based deep-learning diagnostics support case study," in *2018 IEEE Biomedical Circuits and Systems Conference (BioCAS)*. IEEE, 2018, pp. 1–4.

[16] M. G. Costa, C. F. Costa Filho, A. Kimura, P. C. Levy, C. M. Xavier, and L. Fujimoto, "A sputum smear microscopy image database for automatic bacilli detection in conventional microscopy," in *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, 2014, pp. 2841–2844.

[17] M. I. Shah, S. Mishra, V. K. Yadav, A. Chauhan, M. Sarkar, S. K. Sharma, and C. Rout, "Ziehl–neelsen sputum smear microscopy image database: a resource to facilitate automated bacilli detection for tuberculosis diagnosis," *Journal of Medical Imaging*, vol. 4, no. 2, p. 027503, 2017.

[18] T. F. M. Carvalho *et al.*, "Técnicas de inteligência computacional aplicadas ao diagnóstico de tuberculose," 2023.

[19] S. Kant and M. M. Srivastava, "Towards automated tuberculosis detection using deep learning," in *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2018, pp. 1250–1253.

[20] M. El-Melegy, D. Mohamed, T. ElMelegy, and M. Abdelrahman, "Identification of tuberculosis bacilli in zn-stained sputum smear images: A deep learning approach," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019, pp. 1131–1137.

[21] T. F. M. Carvalho, V. L. A. Santos, J. C. F. Silva, L. J. de Assis Figueredo, S. S. de Miranda, R. de Oliveira Duarte, and F. G. Guimarães, "A systematic review and repeatability study on the use of deep learning for classifying and detecting tuberculosis bacilli in microscopic images," *Progress in Biophysics and Molecular Biology*, vol. 180, pp. 1–18, 2023.

[22] E. Udegova, I. Shelomentseva, and S. Chentsov, "Optimizing convolutional neural network architecture for microscopy image recognition for tuberculosis diagnosis," in *International Conference on Neuroinformatics*. Springer, 2021, pp. 204–209.