Original Paper

Improving Tuberculosis Detection in Chest X-Ray Images Through Transfer Learning and Deep Learning: Comparative Study of Convolutional Neural Network Architectures

Alex Mirugwe[1], BSc, MSci; Lillian Tamale[2], PhD; Juwa Nyirenda[3], PhD

[1]School of Public Health, Makerere University, Kampala, Uganda
[2]Faculty of Science and Technology, Victoria University, Kampala, Uganda
[3]Department of Statistical Sciences, University of Cape Town, Cape Town, South Africa

Corresponding Author:

Alex Mirugwe, BSc, MSci
School of Public Health
Makerere University
Kawalya Kaggwa Close, Plot 20A
Kampala
Uganda
Phone: 256 701120534
Email: mirugwealex@gmail.com

Related Articles:
Preprint (medRxiv): https://www.medrxiv.org/content/10.1101/2024.08.02.24311396v1
Peer-Review Report by Rapeepan Pitakaso (Reviewer AE): https://med.jmirx.org/2025/1/e 77171
Peer-Review Report by Natthapong Nanthasamroeng (Reviewer AI): https://med.jmirx.org/2025/1/e 77174
Authors' Response to Peer-Review Reports: https://med.jmirx.org/2025/1/e 77221

Abstract

Background: Tuberculosis (TB) remains a significant global health challenge, as current diagnostic methods are often resource-intensive, time-consuming, and inaccessible in many high-burden communities, necessitating more efficient and accurate diagnostic methods to improve early detection and treatment outcomes.

Objective: This study aimed to evaluate the performance of 6 convolutional neural network architectures—Visual Geometry Group-16 (VGG16), VGG19, Residual Network-50 (ResNet50), ResNet101, ResNet152, and Inception-ResNet-V2—in classifying chest x-ray (CXR) images as either normal or TB-positive. The impact of data augmentation on model performance, training times, and parameter counts was also assessed.

Methods: The dataset of 4200 CXR images, comprising 700 labeled as TB-positive and 3500 as normal cases, was used to train and test the models. Evaluation metrics included accuracy, precision, recall, F_1-score, and area under the receiver operating characteristic curve. The computational efficiency of each model was analyzed by comparing training times and parameter counts.

Results: VGG16 outperformed the other architectures, achieving an accuracy of 99.4%, precision of 97.9%, recall of 98.6%, F_1-score of 98.3%, and area under the receiver operating characteristic curve of 98.25%. This superior performance is significant because it demonstrates that a simpler model can deliver exceptional diagnostic accuracy while requiring fewer computational resources. Surprisingly, data augmentation did not improve performance, suggesting that the original dataset's diversity was sufficient. Models with large numbers of parameters, such as ResNet152 and Inception-ResNet-V2, required longer training times without yielding proportionally better performance.

Conclusions: Simpler models like VGG16 offer a favorable balance between diagnostic accuracy and computational efficiency for TB detection in CXR images. These findings highlight the need to tailor model selection to task-specific requirements, providing valuable insights for future research and clinical implementations in medical image classification.

JMIRx Med 2025;6:e66029; doi: 10.2196/66029

Keywords: tuberculosis detection; tuberculosis; TB; chest x-ray classification; diagnostic imaging; radiology; medical imaging; convolutional neural networks; data augmentation; deep learning; early warning; early detection; comparative study

Introduction

Background

Tuberculosis (TB) remains one of the leading infectious diseases worldwide, affecting an estimated one-third to one-fourth of the global population with the bacillus *Mycobacterium tuberculosis*, the causative agent of TB [1]. In 2019, it was estimated that over 10 million individuals globally contracted TB; yet, only 71% were detected, diagnosed, and reported through various countries' national TB programs, leaving approximately 29% of cases unreported [2]. According to the World Health Organization's (WHO's) 2023 TB report, TB was identified as the second most common cause of death among infectious diseases [3]. Furthermore, the global incidence rate of TB remains alarmingly high at approximately 133 new cases per 100,000 people annually. This situation underscores the need for prompt, effective, and affordable screening and treatment strategies to meet the WHO's ambitious goals of reducing TB incidence by 80%, decreasing TB mortality by 90%, and eliminating catastrophic financial burdens on families affected by TB by 2030 [4].

The WHO advised member countries to proactively conduct TB screening and detection, especially within the high-risk groups, taking into account their unique epidemic scenarios and financial levels [5]. While bacteriological tests, including sputum cultures, sputum smears, and molecular diagnostics, are considered the gold standard for identifying active TB cases, their applicability on a large scale, particularly among high-risk populations, is not feasible [6]. This limitation is due to the methods being resource-intensive, logistically challenging, and associated with prolonged turnaround times [7]. As a result, chest radiography has become the most prevalent method for early TB detection [8]. However, in countries with limited resources, which also bear the highest TB burden, the availability of chest radiography screenings remains inadequate, primarily due to a shortage of radiologists [6].

In recent years, significant advancements have been made in leveraging artificial intelligence (AI), particularly through machine learning and deep learning techniques, for analyzing chest x-ray (CXR) images to differentiate between TB-positive and TB-negative images [9-15]. This innovation has enabled individuals without radiology expertise to conduct TB screening tests, presenting a significant shift in diagnostic approaches. These technologies have shown promising results, to the extent of outperforming radiologists in the interpretation of CXR images [14,15]. Despite this progress, the adoption of AI-based TB detection in low-income countries faces limitations, including a lack of computational resources, inconsistent data quality, and the need for models tailored to diverse clinical and demographic contexts. Addressing these challenges is critical to ensuring the scalability and utility of AI-driven diagnostic tools in these settings.

This research investigates the effectiveness of different convolutional neural network (CNN) architectures in classifying TB in CXR images. We compare and evaluate the performance of popular CNN models, including Residual Network (ResNet), Inception, and Visual Geometry Group (VGG), and examine the impact of different hyperparameters on classification accuracy. The choice of these architectures is motivated by gaps in existing literature, where limited studies compare the performance of advanced CNN models on larger, diverse datasets. Additionally, we explore the impact of transfer learning and data augmentation techniques, providing insights into their role in optimizing model performance.

To the best of our knowledge, this study is the first to use a larger and more diverse dataset and conduct a comprehensive comparison of the latest CNN architectures, including ResNet101, ResNet152, and Inception-V2, assessed across different parameters. The research aims to address the following questions: (1) How does the choice of CNN architecture affect the classification performance? (2) What is the optimal hyperparameter configuration for each CNN architecture? (3) Can transfer learning be leveraged to improve classification accuracy? (4) How does incorporating data augmentation techniques impact the model's performance compared to training solely on real images?

The rest of the paper is organized as follows. In the Related Work section, we present the literature review, which provides an overview of the current state of research in the field. This is followed by the Methods section, where we describe the deep learning models used in this research along with the techniques for improving training time, such as transfer learning. We also describe the data and analysis procedures used in our study, such as data augmentation to mitigate against imbalance. Next, we present the results of our analysis, including any findings. Finally, we discuss the implications of our results, conclude with a summary of our main findings, and suggest areas for future research.

Related Work

Research in the field of medical imaging, particularly in automating the screening and identification of TB from CXR images, has progressed significantly. Initial investigations explored traditional machine learning techniques, including support vector machines [16,17], decision trees [18,19], random forests [20,21], and extreme gradient boosting [22,23], among others. However, recent advancements have shifted focus toward deep learning methods, such as CNNs, which have demonstrated promising results in image classification comparable to those of radiologists [13-15,24]. Below, we review some of the recent studies that have used deep learning approaches for detecting TB in CXR images.

Hooda et al [13] proposed a 19-layer CNN architecture for detecting TB, consisting of 7 convolutional layers, 7 rectified linear unit (ReLU) layers, 3 fully connected layers, and 2 dropouts layers. The model was trained on a dataset of 800 CXR images, each resized to 224×224 pixels. Using the Adam optimizer, the study achieved notable results, with an overall accuracy of 94.73% and a validation accuracy of 82.09%. Although these results are impressive, the authors identified potential areas for further improvements. They suggested investigating the impacts of data augmentation and transfer learning on the model's performance, highlighting avenues for future research enhancements and potential increases in accuracy.

Ojasvi et al [25] developed a classification algorithm for CXR images of potential patients with TB, aiming to improve upon existing models [26]. To mitigate against dataset imbalances and improve model reliability, they combined the NIH Chest X-ray Dataset, China-Shenzhen Chest X-ray Database, and Montgomery County Chest X-ray Database to train and fine-tune their model. By implementing coarse-to-fine transfer learning and extensive data augmentation techniques, they achieved a remarkable accuracy of 94.89% compared to the accuracy of 89.6% achieved by Cao et al [26]. However, the study acknowledges the challenge of maintaining equivalent precision across CXR images obtained in varied settings, as the model was specifically trained for the Chinese dataset.

Panicker et al [27] introduced a novel 2-stage detection method for TB bacilli, using image binarization and CNN classification to analyze microscopic sputum smear images. The method was evaluated on a diverse dataset of 22 images, and the model demonstrated high effectiveness, achieving a recall rate of 97.13%, a precision of 78.4%, and an F_1-score of 86.76%. However, the study noted that the model's ability to accurately detect overlapping bacilli was limited. In the same year, Stirenko et al [28] explored the application of lung segmentation in CXR images and data augmentation to enhance TB detection from CXR images. Their study highlights the critical role of preprocessing, including lung segmentation and data augmentation, in addressing overfitting issues and improving the effectiveness of computer-aided diagnosis systems in TB identification, particularly when working with limited datasets.

The study by Kazemzadeh et al [15] developed a deep learning algorithm for detecting active pulmonary TB from CXR images. The algorithm was trained and validated on a dataset comprising 165,754 images from 22,284 patients from 10 different countries. The algorithm's performance was compared to that of 14 radiologists on datasets from 4 countries, including a cohort from a South African mining population. It achieved an area under the receiver operating characteristic curve (AUC-ROC) of 0.89, with superior sensitivity (88% vs 75%; P=.05) and comparable specificity (79% vs 84%) to radiologists, demonstrating its potential for TB screening in resource-limited settings. Another study by Nijiati et al [29] used a 3D ResNet-50 CNN architecture to differentiate active from nonactive pulmonary TB using computed tomography images. This study, similar to that of Kazemzadeh et al [15], reported high diagnostic accuracy and efficiency, outperforming conventional radiological methods in terms of speed and precision.

In their 2019 study, Meraj et al [30] used CNN architectures such as VGG16, VGG19, ResNet50, and GoogLeNet to automate the detection of TB manifestations in CXRs using 2 public TB image datasets [31]. Their findings showed that the VGG16 model outperformed other architectures in terms of accuracy and AUC-ROC. However, the study was limited by its reliance on small and unbalanced datasets, raising questions about the generalizability of the results. In contrast, our research builds upon and extends the work of Meraj et al [30] by incorporating a larger and more diverse dataset. We also explore the diagnostic capabilities of more advanced CNN architectures, including ResNet101, ResNet152, and Inception-V2, to assess their effectiveness in TB detection. This approach aims to provide a more comprehensive understanding of how recent deep learning advancements can be leveraged for more accurate TB diagnosis in varied clinical settings. The Methods section details the methodological framework to achieve these objectives.

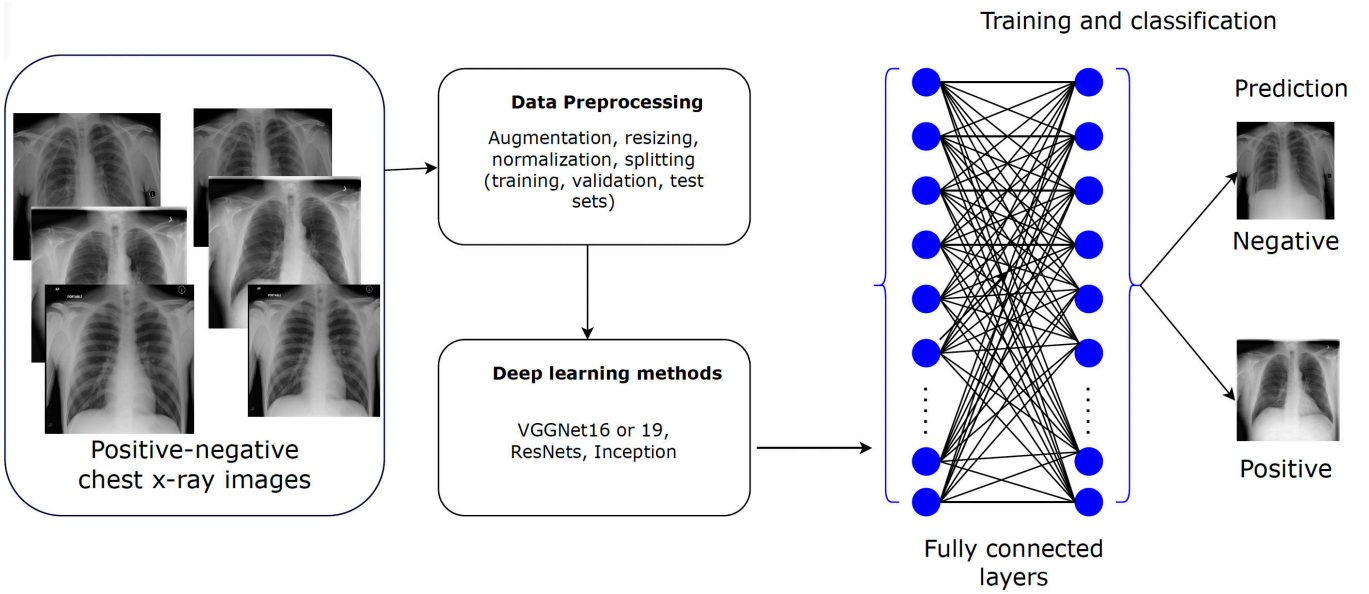
Methods

In this section, we provide a comprehensive overview of the methodologies used in our study, including the dataset and preprocessing, data normalization, data augmentation, the application of transfer learning methods, the architecture of CNNs used, and the evaluation metrics adopted to assess the performance of the models.

Implementation Overview

The implementation framework illustrated in Figure 1 starts with the acquisition of a well-defined dataset, followed by comprehensive data preprocessing, which includes data augmentation, resizing, normalization, and partitioning into training, validation, and test sets. Subsequently, we embark on the development of various deep learning models. These models undergo extensive training and evaluation against different hyperparameters and evaluation metrics to accurately predict and classify CXR images into positive or negative cases of TB.

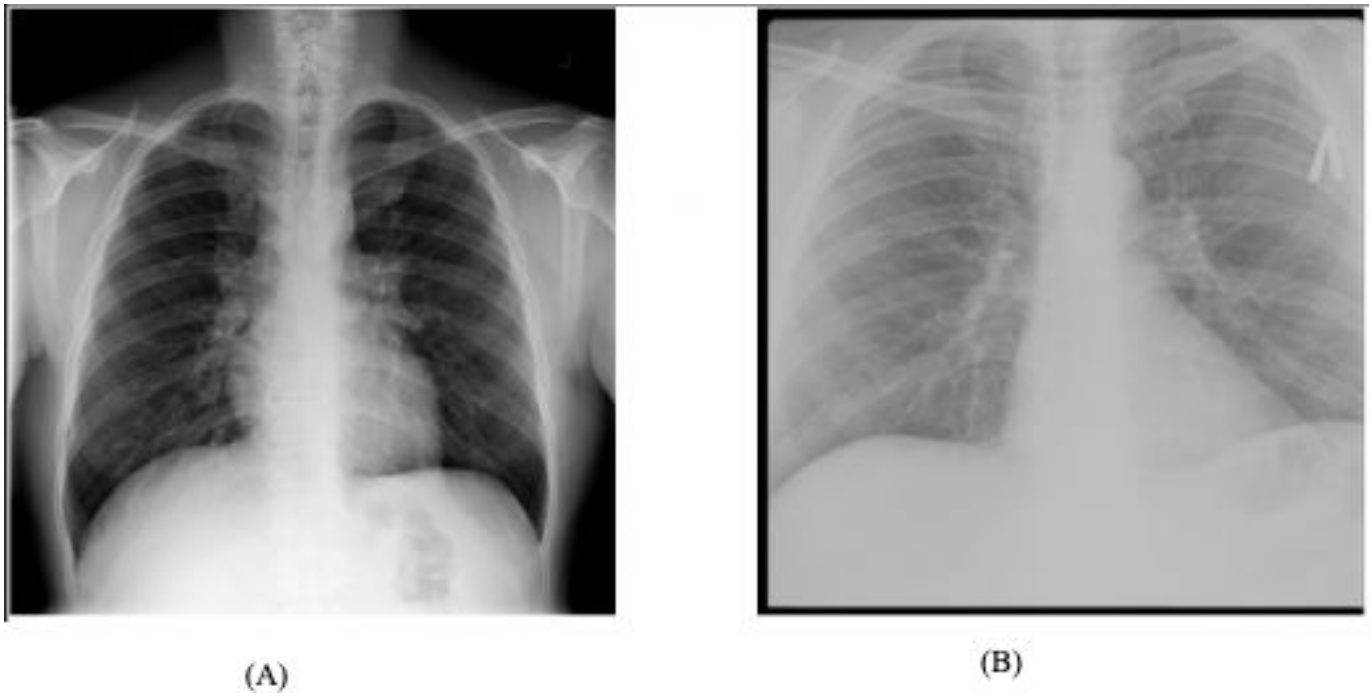
Figure 1. The implementation flow of the deep learning classification methodology. ResNet: Residual Network; VGG: Visual Geometry Group.



Dataset

The dataset used in this research comprises 4200 CXR images sourced from a public Kaggle data repository. The dataset was compiled through a collaborative effort between researchers from Qatar University (Doha, Qatar) and the University of Dhaka (Bangladesh) and collaborators from Malaysia. They worked closely with medical professionals from the Hamad Medical Corporation (Doha, Qatar) and various health care institutions in Bangladesh. The dataset consists of 700 CXR images indicative of TB and 3500 CXR images classified as normal, with all images having a resolution of 512×512 pixels [32]. This composition provides a substantial foundation for evaluating the effectiveness of CNN models in the detection of TB from CXR images. Figure 2 presents some of the images from the dataset.

Figure 2. The chest x-ray sample images. (A) Tuberculosis-negative and (B) tuberculosis-positive.



Preprocessing

To optimize the performance and efficiency of our models, we implemented key preprocessing techniques, specifically data normalization and augmentation, before training the models.

Data Normalization

In the preprocessing stage of image analysis, normalization is a critical step to standardize the input data, facilitating the model's learning process. This study applies normalization to CXR images, which initially possess pixel intensity values in the range of 0 to 255, common for grayscale images [33]. The goal of normalization is to adjust these intensity values to a standardized scale that improves computational efficiency and model convergence during training. The normalization process is mathematically represented as follows:

$$I' = \frac{I - I_{\min}}{I_{\max} - I_{\min}} \quad (1)$$

where I represents the original pixel intensity of the image, $I_{\min}$ and $I_{\max}$ are the minimum and maximum possible intensity values in the original image, respectively, and I is the normalized pixel intensity.

For grayscale images, $I_{\min}$=0 and $I_{\max}$=255. This equation effectively rescales the pixel intensity values to the range (0-1), making the input data more suitable for processing by the neural network layers. This normalization technique is advantageous because it ensures that each input parameter (pixel, in this case) contributes equally to the analysis, preventing features with initially larger ranges from dominating the learning process [34]. It also helps to stabilize the gradient descent optimization algorithm by maintaining a consistent scale for all gradients [35]. Previous studies have shown that normalization significantly improves convergence rates and ensures model stability, particularly in image classification tasks involving deep learning [34,35].

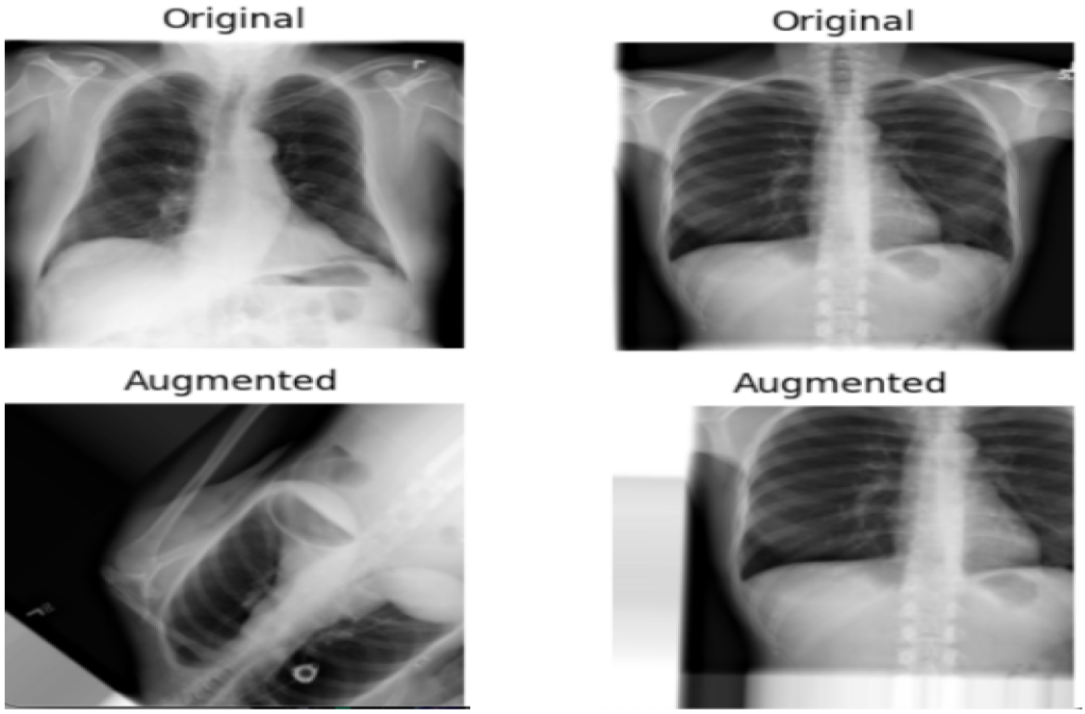
Data Augmentation

Data augmentation represents a powerful regularization strategy designed to artificially increase the dataset through label-preserving transformations, thereby incorporating more invariant examples into the training set [36]. This approach, characterized by its computational efficiency, has been previously used to reduce overfitting when training CNNs, such as in the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC), where it contributed to achieving state-of-the-art results [37]. This method enhances the robustness and generalizability of deep learning models by exposing them to a wider array of variations, simulating real-world variability.

In our study, to address the imbalance between TB-positive and TB-negative images and to introduce different variations, we randomly augmented 210 (30%) TB-positive images and 175 (5%) TB-negative images. The data augmentation techniques applied included random rotation within a range of 0 to 60 degrees, random width and height shifts of up to 0.2 times the image size, and random zooming of up to 0.2 times the original size, alongside horizontal and vertical flipping. To manage the newly created pixels from such transformations, a "fill mode" strategy was used, ensuring integrity and consistency in the augmented images. These augmentations were performed using Keras's ImageDataGenerator, a comprehensive data augmentation suite [38].

While data augmentation techniques are widely adopted in deep learning research, our implementation aligns with prior studies that highlight their utility in addressing dataset imbalance and improving model generalization in medical imaging tasks [36,37]. Additionally, the augmentation strategy in this study was tailored to reflect the variability commonly observed in real-world CXR data, enhancing the robustness of our models. Figure 3 shows a sample of real images and their corresponding augmented outputs.

Figure 3. Sample of real and corresponding augmented images.



Transfer Learning

Transfer learning is a machine learning technique where a model developed for a specific task is repurposed as the starting point for a model on a second, related task [39]. This technique leverages the knowledge gained during the initial training phase in one domain to enhance learning in another potentially unrelated domain. It operates under the principle that information learned in one context can be exploited to accelerate or improve the optimization process in another, essentially allowing for the transfer of learned features and patterns across different but related problems [39].

In this study, we propose an implementation that capitalizes on the transfer learning paradigm by using pretrained models such as Inception-V3, ResNet (50, 101, and 152), and VGG (16 and 19), which were initially trained on the ImageNet dataset [37]. This adaptation involves fine-tuning and customizing the models' last layers to suit our classification task, effectively tailoring the robust, prelearned representations of the ImageNet dataset to recognize and interpret the specific patterns and anomalies associated with TB in CXR images.

We opted for transfer learning over training models from scratch due to its significant advantages, particularly in the context of medical imaging. Training deep learning models from scratch requires large datasets, extensive computational resources, and longer training times. These requirements often pose challenges in health care–related research, especially when working with relatively small or domain-specific datasets like CXRs. Transfer learning allows us to leverage the rich feature representations of pretrained models while reducing training time and computational demands. Furthermore, studies have shown that transfer learning enhances model performance in medical imaging tasks by effectively repurposing features learned from general image datasets like ImageNet to domain-specific tasks [37,39].

CNN Architectures

In the next subsections, we provide a brief description of the VGG and ResNet families of CNN architectures as well as the Inception ResNet architecture that is considered in this study.

VGGNet

Introduced by Simonyan and Zisserman from the University of Oxford's Visual Geometry Group in 2014, the VGGNet architecture marked a significant milestone in the field of deep learning [40]. Known for its outstanding performance in the ILSVRC of that year, VGGNet is characterized by its use of 3×3 filters in all convolutional layers, simulating the effects of larger receptive fields. This architecture is available in 2 variants, VGG16 and VGG19, differing in depth and the number of layers, with VGG19 being the deeper model.

In our research, we used both the VGG16 and VGG19 architectures to train models on datasets consisting of solely real CXR images and a combination of augmented and real images. This approach aimed to assess the impact of incorporating augmented images on the performance of these 2 architectures. Images were resized to 256×256 pixels before being input into the networks. We extended the architectures by adding a flattening layer, followed by a dense layer of 512 neurons with a ReLU activation function and a dropout layer with a dropout rate of 0.2 to mitigate overfitting. A softmax activation function was used in the output layer for binary classification. We used the Adam optimizer with the binary cross-entropy loss function for optimization. The training was conducted over 15 epochs with a batch size of 32 for both models. This rigorous approach ensured that

both architectures could classify between TB-positive and TB-negative CXR images accurately.

ResNet

He et al [41] introduced the deep residual network (ResNet) architecture in their 2016 seminal paper. This architecture greatly improved the performance of deep neural networks and went on to win the Common Objects in Context object detection challenge and the 2015 ILSVRC. To date, several variants of the ResNet architecture exist, including ResNet50, ResNet101, and ResNet152, which vary in depth and number of layers. ResNet architectures are very deep models [41,42]. The core idea behind ResNet is the use of residual connections, also known as shortcuts, which bypass 1 or more layers. By resolving the vanishing gradient issue, these shortcuts maintain the gradient flow across the network and facilitate the training of much deeper networks [41].

The CXR images in this study were classified using the ResNet50, ResNet101, and ResNet152 architectures. We added 3 more layers to the ResNet50 model, 2, each with 256 units and 1 with 512 units, using batch normalization and ReLU activation in each layer. To reduce overfitting, dropout layers were added with dropout rates of 0.3, 0.25, and 0.2, respectively. The binary cross-entropy loss function was used to compile the model, while the Adam optimizer was used to optimize the model at a learning rate of 0.001. Two units with a softmax activation function made up the output layer, which classified the images as either TB-positive or TB-negative. Training for this model involved 16 batch sizes and 100 epochs.

ResNet101 was trained using the same settings as ResNet50, as preliminary training showed that the same parameter values used for ResNet50 also yielded optimal results for the ResNet101 architecture. For ResNet152, a selective fine-tuning approach was adopted, where only the last 10 layers of the network were trainable, enhancing the model's focus on more feature-specific adjustments in the later stages of the network. This model shared the augmentation layers of ResNet50 but was trained for only 50 epochs, incorporating a learning rate scheduler, ReduceLROnPlateau, which adjusted the rate based on the validation loss with a factor of 0.1, patience of 5, and a minimum learning rate of 1×10^{-6}, thereby optimizing the training dynamics. The details of the models' configuration are shown in Table 1.

Table 1. Training hyperparameters of ResNet[a] models.

Hyperparameter	ResNet50	ResNet101	ResNet152
Layers, n	53 (50 base +3 extra)	104 (101 base +3 extra)	155 (152 base +3 extra)
Units per layer	256, 256, 512	256, 256, 512	256, 256, 512
Activation	ReLU[b]	ReLU	ReLU
Batch normalization	Yes	Yes	Yes
Dropout rate	0.3, 0.25, 0.2	0.3, 0.25, 0.2	0.3, 0.25, 0.2
Optimizer	Adam	Adam	Adam
Learning rate	0.001	0.001	Variable (ReduceLROnPlateau)
Loss function	Binary cross-entropy	Binary cross-entropy	Binary cross-entropy
Training epochs	100	100	50
Batch size	16	16	16

[a]ResNet: Residual Network.
[b]ReLU: rectified linear unit.

Inception-ResNet

The Inception networks, introduced by Szegedy et al [43], have greatly advanced the field of CNN, as they have achieved state-of-the-art performance in a number of computer vision problems [43-45]. The original Inception-V1, also known as GoogLeNet, was first introduced in 2014 and won the ILSVRC of that year. The architecture introduced a novel approach of using multiple convolutional filter sizes in parallel, allowing the network to capture various spatial features of different scales with improved use of computing resources [43].

In this study, we used Inception-ResNet-V2 architecture, a hybrid model that combines the benefits of both the Inception and residual networks. This hybrid approach enables the architecture to learn more complex features with improved training stability and faster convergence [43]. The Inception-ResNet-V2 also leverages residual connections to skip certain layers during training, which helps it improve gradient flow, accelerate training times, and reduce the likelihood of vanishing gradient problems in deep networks [46]. We selected Inception-ResNet-V2 due to its demonstrated state-of-the-art results in several medical imaging tasks [45].

For our implementation, the Inception-ResNet-V2 architecture was initialized with weights pretrained on the ImageNet dataset. Similar to our approach with the ResNet152 model, all layers except the last 10 were frozen to retain the pretrained features from ImageNet. The last 10 layers were set to be trainable, enabling the model to learn specific features from the CXR images. We added 3 new layers: 2 with 256 units each and 1 with 512 units, all using ReLU activations and batch normalization. Each of these layers was followed by dropout layers with rates of 0.4, 0.35, and 0.3, respectively, to introduce nonlinearity and reduce overfitting. The final output layer consisted of 2 units with a softmax activation function for binary classification. The

model was then compiled using binary cross-entropy as the loss function and the Adam optimizer with a learning rate of 0.0001. Training was conducted for 50 epochs with a batch size of 16.

The parameters used in the training of all these CNN architectures, including dropout rates, learning rates, batch sizes, and the number of epochs, were determined through a rigorous iterative process of experimentation. This approach involved fine-tuning each parameter to optimize model performance while avoiding overfitting. The configurations presented reflect the parameter values that consistently yielded good performance across the different architectures.

Evaluation Metrics

The performance of the CNN architectures in classifying CXR images into TB-positive and TB-negative categories was assessed using several standard performance metrics, including accuracy, precision, recall, F_1-score, and the AUC-ROC. Each metric provides unique insights into the model's classification abilities, considering both the true and false predictions.

Accuracy

This metric measures the proportion of true positive (TP) and true negative (TN) results among the total number of cases examined:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (2)$$

where TP is the number of TB-positive images that are correctly identified as TB-positive by the model, TN is the number of TB-negative images that are correctly identified as TB-negative by the model, FP (false positives) is the number of TB-negative images that are incorrectly identified as TB-positive by the model, and FN (false negatives) is the number of TB-positive images that are incorrectly identified as TB-negative by the model.

Precision

Also known as positive predictive value, precision is the ratio of correctly identified TB cases to all cases that were diagnosed as TB by the model. It measures the model's accuracy in diagnosing a patient with TB when the model predicts the disease. High precision indicates a low rate of false TB diagnoses. Mathematically, it is defined as:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (3)$$

Recall

Recall, or sensitivity, is especially critical in medical diagnostics, as it quantifies the model's ability to correctly identify all actual TB cases. It represents the proportion of actual TB cases that were correctly identified by the model and aims to minimize the risk of missing a true TB case. It is computed as:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (4)$$

F_1-Score

The F_1-score is the harmonic mean of precision and recall, providing a single measure that balances both the FP and FN. In TB diagnosis, it is particularly useful because it creates a balance between precision (minimizing false TB diagnoses) and recall (minimizing missed TB diagnoses), which is crucial for medical screening tests. It is defined as:

$$F_1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (5)$$

AUC-ROC

The AUC-ROC measures a model's ability to discern between positive and negative classes. In the context of our problem, that specifically refers to distinguishing between TB-positive and TB-negative CXR images. The AUC-ROC is a plot of the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings. The AUC-ROC provides an aggregated measure of the model's performance across all classification thresholds, with a value of 1 representing a perfect model and a value of 0.5 representing a model with no discriminatory power. The approximate AUC-ROC is calculated by using the following formula:

$$\text{AUC} \approx \sum_{i=1}^{n} \frac{(\text{FPR}_i - \text{FPR}_{i-1}) \times (\text{TPR}_i + \text{TPR}_{I-1})}{2} \quad (6)$$

where i is the current data point or threshold, FPR_i and TPR_i are the false positive and true positive rates at the ith threshold, respectively, and n is the number of data points or thresholds used to calculate the AUC-ROC. Each term in the sum represents the area of a trapezoid, where ($\text{FPR}_i - \text{FPR}_{i-1}$) is the base of the trapezoid and ($\text{TPR}_i + \text{TPR}_{I-1}$)/2 is the average height of the trapezoid. The formula calculates the AUC-ROC by summing the areas of trapezoids formed by connecting consecutive points on the AUC-ROC.

Computational Environment

The implementation and findings of this study were based on using the Keras 3.3.3 and TensorFlow 2.16.1 frameworks. The experiments were conducted on a single GPU MSI GL75 Leopard 10SFR laptop with 32 GB of RAM and an 8 GB NVIDIA GEFORCE RTX 2070 GDDR6 card. The system was operated using the CUDA 12.1 and cuDNN SDK 8.7.0 platforms to ensure efficient GPU acceleration and deep learning model training.

These methodological choices, including dataset selection, preprocessing techniques, CNN architectures, and model evaluation techniques, were designed to ensure a rigorous and comprehensive analysis of CNN performance for TB detection. The results of these analyses are presented in the following section.

Ethical Considerations

This study used a publicly available, deidentified dataset from Kaggle. As such, it did not require institutional review board approval. The dataset does not contain any personally identifiable information, and informed consent was not applicable. No participants were directly involved in this study, and no compensation was provided.

Results

Overview

The study aimed to analyze and compare the performance of various CNN architectures, including VGG16, VGG19, ResNet50, ResNet101, ResNet152, and Inception-ResNet-V2, in classifying CXR images as either TB-positive or TB-negative. Additionally, we also investigated whether data augmentation could further improve the classification performance of these models by comparing the performance of models trained on only real images versus those trained on a combination of real and augmented data. We went further to examine the training time and the number of parameters for each architecture to understand the computational efficiency and resource demands for each model. This analysis is important for practical implementation, particularly in resource-constrained settings where training time and computational costs are significant considerations. By evaluating these parameters, we aimed to identify models that not only perform well but also offer a balanced trade-off between accuracy and efficiency, making them suitable for real-world applications in diverse health care environments.

Table 2 summarizes the performance of CNN architectures across accuracy, precision, recall, and F_1-score, highlighting the impact of training on real images versus a combination of real and augmented data. Table 3 shows the performance of these models when evaluated using the AUC-ROC score metric. It was observed that the VGG16 outperformed all other architectures across all metrics, with an accuracy of 99.4%, precision of 97.9%, recall of 98.6%, F_1-score of 98.3%, and area under the curve of 98.25%. Its performance was superior consistently, irrespective of whether the models were trained with or without data augmentation.

Table 2. Evaluation of convolutional neural network (CNN) architectures across key evaluation metrics[a].

Architecture	Accuracy (%)	Precision (%)	Recall (%)	F_1-score (%)
VGG16[b]	99.4	97.9	98.6	98.3
VGG16[c]	99.3	96.6	99.3	97.9
VGG19	99.2	96.6	98.6	97.6
VGG19[c]	99.2	96.6	98.6	97.6
ResNet50[d]	96.1	81.3	96.9	88.4
ResNet50[c]	89	97.5	30	45.9
ResNet101	96.9	94.8	84.6	89.3
ResNet101[c]	97.3	92.1	90	91.1
ResNet152	97.9	93.6	93.6	93.6
ResNet152[c]	97.5	87.6	96.6	92.1
Inception ResNet-v2	99	95.9	98.6	97.2
Inception ResNet-v2[c]	99.2	97.2	97.9	97.5

[a]This table summarizes the performance of various CNN architectures according to precision, recall, and F_1-score.
[b]VGG: Visual Geometry Group.
[c]Models were trained using a combination of real and augmented data, showcasing the impact of data augmentation on model performance.
[d]ResNet: Residual Network.

Table 3. The models' area under the curve (AUC) scores.

Model	AUC (without data augmentation)	AUC (with data augmentation)
VGG16[a]	98.25	97.95
VGG19	97.6	97.6
ResNet50[b]	85.65	63.75
ResNet101	89.6	91.05
ResNet152	93.45	89.85
Inception ResNet-v2	92.75	97.55

[a]VGG: Visual Geometry Group.
[b]ResNet: Residual Network.

Surprisingly, increasing the dataset size through data augmentation did not correspond with an increase in the performance of the models across all architectures, as seen in Table 2. This was also observed in other models, such

as ResNet50, where when augmented data were included, the AUC-ROC score dropped significantly from 85.65% to 63.75%, as shown in Table 3. This suggests that the introduction of augmented data may have introduced noise or overcomplicated the training process for certain architectures, negatively impacting their ability to generalize effectively.

Training Time

We also tracked each model's training time with a combination of data augmentation and real images versus training with only real images, as shown in Table 4. As expected, training with data augmentation requires more time due to the increased size of the dataset. For example, training the ResNet152 with data augmentation took 356.6 minutes, whereas training without augmentation took 345.7 minutes. This observation highlights the trade-off between longer training times and the potential benefits of data augmentation. However, data augmentation did not improve performance in our case, indicating that the additional training time did not translate into better model generalization.

Table 4. Training time for the models.

Model	AUC[a] (real images)	AUC (real and augmented data)
VGG16[b]	98.25	97.95
VGG19	97.6	97.6
ResNet50[c]	85.65	63.75
ResNet101	89.6	91.05
ResNet152	93.45	89.85
Inception ResNet-v2	92.75	97.55

[a]AUC: area under the curve.
[b]VGG: Visual Geometry Group.
[c]ResNet: Residual Network.

Model Parameters

In addition to our analysis, we provide a detailed breakdown of the parameter count for each model used in our study, as shown in Table 5. The number of parameters in a model reflects its complexity and capacity to learn from data. Consequently, it has a direct impact on both training time and the computational resources required, influencing the model's overall efficiency and scalability.

Table 5. Parameters of each model.

Model	Parameters, n
Inception-ResNet-V2	54,336,736
ResNet152[a]	58,370,944
ResNet101	42,658,176
ResNet50	23,587,712
VGG19[b]	20,024,384
VGG16	14,714,688

[a]ResNet: Residual Network.
[b]VGG: Visual Geometry Group.

The results highlight the superior performance of VGG16 in terms of diagnostic accuracy and computational efficiency, challenging the hypothesis that more complex models always yield better results. These findings and their broader implications for TB diagnostics are explored in the Discussion section.

Discussion

Principal Findings

The findings from this study provide significant insights into the performance and efficiency of several CNN architectures in the classification of CXR images for TB detection. The architectures evaluated included VGG16, VGG19, ResNet50, ResNet101, ResNet152, and Inception-ResNet-V2. Of these, the VGG16 consistently achieved the highest performance across all metrics, such as accuracy, precision, recall, and F_1-score. This consistent performance suggests that VGG16 effectively captures the necessary features for distinguishing between TB-positive and TB-negative CXR images, even with fewer parameters compared to the deeper models. VGG16's superior performance is significant, as it demonstrates that a simpler model can achieve exceptional diagnostic accuracy while requiring minimal computational resources. This makes it a practical and scalable solution for deployment in resource-constrained settings with limited access to high-performance hardware.

The computational time observed across models has implications for clinical settings, particularly in resource-limited environments. Longer training times, as seen with complex architectures like ResNet152, increase resource

demands, potentially impacting cost-effectiveness. Importantly, since data augmentation did not improve model performance in this study, the additional computational burden may not be justifiable in such settings. Simpler models, like VGG16 or ResNet50, may offer a more feasible balance between efficiency and diagnostic accuracy, making them better suited for practical implementation.

Comparison to Prior Work

The findings also highlight the fact that while data augmentation is often used to improve the performance of CNN models by expanding the dataset and introducing variability, it does not necessarily lead to performance improvements if the base dataset already provides sufficient diversity for training. In our study, the original dataset appeared robust enough, and the addition of augmented data did not enhance model performance. This aligns with findings from previous studies, such as the study by Shorten and Khoshgoftaar [47], which emphasize that the effectiveness of data augmentation is highly dependent on the initial dataset's characteristics, particularly its size and variability. When the base dataset is sufficiently diverse, as in our case, augmentation may introduce unnecessary redundancy or even noise, potentially disrupting the model's ability to generalize effectively.

However, our findings also contrast with studies in domains where datasets are inherently limited or imbalanced, such as biomedical imaging, where augmentation has been shown to significantly improve performance by addressing underrepresented classes and introducing variability. For instance, a study by Perez and Wang [48] demonstrated that data augmentation improved model generalization for small datasets by simulating real-world variability. The discrepancy between our results and these studies highlights the context-dependent nature of augmentation's effectiveness and the need for tailoring augmentation strategies to specific datasets and tasks.

It is commonly observed in several studies that models with a higher number of parameters, such as ResNet152 and Inception-ResNet-V2, are capable of capturing more deep patterns in the data [41,43]. However, this comes at the cost of requiring more computational resources and longer training times. Interestingly, in our study, despite having fewer parameters, VGG16 outperformed the more complex models. This suggests that for our specific task of classifying CXR images into TB-positive and TB-negative categories, VGG16 efficiently captured the relevant features without necessitating excessive complexity. This finding highlights the importance of selecting the appropriate model architecture based on the specific characteristics and requirements of the task at hand rather than simply opting for the model with the most parameters. This result also aligns with the principle that simpler models can often perform competitively when they are well-matched to the data and the problem domain [40].

Strengths and Limitations

The findings from this study show that a simpler model like VGG16 can deliver strong performance while keeping computational requirements low. This makes it suitable for use in low-resource environments. The study also measured training time across different architectures, which helps evaluate practical efficiency.

The study used a publicly available dataset from Kaggle. While the dataset is extensive, it may not reflect the full range of clinical variability found in real-world populations. Only one data augmentation approach was applied, and results might vary with other techniques or combinations.

Conclusions

This study presents a comprehensive evaluation of several CNN architectures—VGG16, VGG19, ResNet50, ResNet101, ResNet152, and Inception-ResNet-V2—in classifying CXR images as either TB-positive or TB-negative. The findings showed that the VGG16 architecture consistently outperformed the other models across all the evaluation metrics, achieving superior performance despite having fewer parameters compared to the more complex architectures such as ResNet152 and Inception-ResNet-V2. These results align with previous studies, such as those by Meraj et al [30] and Lakhani and Sundaram [12], which also highlighted the high diagnostic accuracy and efficiency of simpler architectures like VGG16 for TB detection in CXR images. However, our study extends these findings by demonstrating that VGG16 performs robustly even on larger, more diverse datasets, further validating its applicability to real-world scenarios.

Our results also showed limited benefits of data augmentation in this context, suggesting that the original dataset provided sufficient diversity for effective training. This finding is consistent with previous research emphasizing that the utility of data augmentation is highly context-dependent and may not always lead to performance improvements, particularly when the dataset already exhibits sufficient variability. However, it contrasts with studies where augmentation proved essential for improving performance in smaller, imbalanced datasets, highlighting the need for task-specific augmentation strategies. Furthermore, the study demonstrated significant trade-offs between model complexity, training time, and performance. Models with higher parameters, such as ResNet152 and Inception-ResNet-V2, required longer training times and more computational resources without corresponding improvements in classification performance across all evaluation metrics. This emphasizes the importance of selecting model architectures based on task requirements rather than defaulting to more complex models. Simpler models like VGG16 not only achieved higher accuracy but also demonstrated computational efficiency, making them particularly suitable for resource-constrained environments. The practical implications of this finding are significant: VGG16's lower computational requirements and superior performance enable its deployment in low-resource health care settings, where access to high-performance hardware and technical expertise may be limited.

Overall, our research contributes to the growing body of evidence supporting the effectiveness of deep learning models in medical image classification and provides actionable

insights into optimizing these models for TB detection in CXR images. By addressing key considerations such as dataset diversity, model complexity, and computational efficiency, this study offers practical guidance for implementing AI-driven TB diagnostic tools in real-world clinical environments.

Acknowledgments

The authors would like to extend their sincere gratitude to the team of researchers from Qatar University, Doha, Qatar, and the University of Dhaka, Bangladesh, along with their collaborators from Malaysia and medical doctors from Hamad Medical Corporation and various health care institutions in Bangladesh for creating and sharing the chest x-ray image database for tuberculosis. Their effort in compiling this comprehensive dataset has significantly contributed to our research. The authors are grateful for their contributions and their dedication to advancing tuberculosis diagnosis and treatment through the provision of this valuable public dataset.

Data Availability

The dataset analyzed during this study is publicly available and was obtained from the Kaggle repository [32].

Conflicts of Interest

None declared.

References

1. Assefa Y, Woldeyohannes S, Gelaw YA, Hamada Y, Getahun H. Screening tools to exclude active pulmonary TB in high TB burden countries: systematic review and meta-analysis. Int J Tuberc Lung Dis. 2019;23(6):728-734. [doi: 10.5588/ijtld.18.0547]
2. Chakaya J, Khan M, Ntoumi F, et al. Global Tuberculosis Report 2020—reflections on the Global TB burden, treatment and prevention efforts. Int J Infect Dis. Dec 2021;113 Suppl 1(Suppl 1):S7-S12. [doi: 10.1016/j.ijid.2021.02.107] [Medline: 33716195]
3. Global tuberculosis report. World Health Organization. 2023. URL: https://www.who.int/teams/global-programme-on-tuberculosis-and-lung-health/tb-reports/global-tuberculosis-report-2023 [Accessed 2025-05-31]
4. Mukund P, Diana W, Knut L, et al. WHO's new end TB strategy. Lancet. 2015;385:1799-1801. [doi: 10.1016/S0140-6736(15)60570-0]
5. Systematic screening for active tuberculosis: an operational guide. World Health Organization. URL: https://www.who.int/publications/i/item/9789241549172 [Accessed 2025-05-31]
6. Liao Q, Feng H, Li Y, et al. Evaluation of an artificial intelligence (AI) system to detect tuberculosis on chest X-ray at a pilot active screening project in Guangdong, China in 2019. J Xray Sci Technol. 2022;30(2):221-230. [doi: 10.3233/XST-211019] [Medline: 34924433]
7. van't Hoog AH, Meme HK, Laserson KF, et al. Screening strategies for tuberculosis prevalence surveys: the value of chest radiography and symptoms. PLoS One. 2012;7(7):e38691. [doi: 10.1371/journal.pone.0038691] [Medline: 22792158]
8. Pande T, Pai M, Khan FA, Denkinger CM. Use of chest radiography in the 22 highest tuberculosis burden countries. Eur Respir J. Dec 2015;46(6):1816-1819. [doi: 10.1183/13993003.01064-2015]
9. Kant S, Srivastava MM. Towards automated tuberculosis detection using deep learning. Presented at: 2018 IEEE Symposium Series on Computational Intelligence (SSCI); Nov 18-21, 2018:1250-1253; Bangalore, India. [doi: 10.1109/SSCI.2018.8628800]
10. Sangheum H, Hyo-Eun K, Jihoon J, Hee-Jin K. A novel approach for tuberculosis screening based on deep convolutional neural networks. Presented at: Medical Imaging 2016: Computer-Aided Diagnosis; Mar 27-3, 2016:750-757; San Diego, CA, United States. [doi: 10.1117/12.2216198]
11. Kim TK, Yi PH, Hager GD, Lin CT. Refining dataset curation methods for deep learning-based automated tuberculosis screening. J Thorac Dis. Sep 2020;12(9):5078-5085. [doi: 10.21037/jtd.2019.08.34] [Medline: 33145084]
12. Lakhani P, Sundaram B. Deep learning at chest radiography: automated classification of pulmonary tuberculosis by using convolutional neural networks. Radiology. Aug 2017;284(2):574-582. [doi: 10.1148/radiol.2017162326]
13. Hooda R, Sofat S, Kaur S, Mittal A, Meriaudeau F. Deep-learning: a potential method for tuberculosis detection using chest radiography. Presented at: 2017 IEEE International Conference on Signal and Image Processing Applications (ICSIPA); Sep 12-14, 2017:497-502; Kuching, Malaysia. 2017.[doi: 10.1109/ICSIPA.2017.8120663]
14. Khanh HTK, Jeonghwan G, Om P, Jong-In S, Min PC. Utilizing pre-trained deep learning models for automated pulmonary tuberculosis detection using chest radiography. Presented at: Intelligent Information and Database Systems: 11th Asian Conference, ACIIDS 2019; Apr 8-11, 2019:395-403; Yogyakarta, Indonesia.
15. Kazemzadeh S, Yu J, Jamshy S, et al. Deep learning detection of active pulmonary tuberculosis at chest radiography matched the clinical performance of radiologists. Radiology. Jan 2023;306(1):124-137. [doi: 10.1148/radiol.212213]

16. Zulvia FE, Kuo RJ, Roflin E. An initial screening method for tuberculosis diseases using a multi-objective gradient evolution-based support vector machine and C5.0 decision tree. Presented at: 2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC); Jul 4-8, 2017:204-209; Turin, Italy. [doi: 10.1109/COMPSAC.2017.57]
17. Saybani MR, Shamshirband S, Golzari Hormozi S, et al. Diagnosing tuberculosis with a novel support vector machine-based artificial immune recognition system. Iran Red Crescent Med J. Apr 2015;17(4):e24557. [doi: 10.5812/ircmj.17(4)2015.24557] [Medline: 26023340]
18. Fahadulla HS, Fareed Z, Adeel ZM, Aasia K, Khan Imran H, Raza A. Decision-tree inspired classification algorithm to detect tuberculosis (TB). Presented at: 21st Pacific-Asia Conference on Information Systems (PACIS 2017); Jul 16-20, 2017; Langkawi Island, Malaysia. 2017.URL: https://aisel.aisnet.org/pacis2017/182 [Accessed 2025-06-20]
19. Mithra KS, Emmanuel WRS. FHDT: fuzzy and hyco-entropy-based decision tree classifier for tuberculosis diagnosis from sputum images. Sādhanā. Aug 2018;43(8). [doi: 10.1007/s12046-018-0878-y]
20. Ayas S, Ekinci M. Random forest-based tuberculosis bacteria classification in images of ZN-stained sputum smear samples. SIViP. Dec 2014;8(S1):49-61. [doi: 10.1007/s11760-014-0708-6]
21. Chi Z, Jingxin L, Guoping Q. Tuberculosis bacteria detection based on random forest using fluorescent images. Presented at: 2016 9th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI); Oct 15-17, 2016:553-558; Datong, China.
22. Rahman M, Cao Y, Sun X, Li B, Hao Y. Deep pre-trained networks as a feature extractor with XGBoost to detect tuberculosis from chest X-ray. Comput Electr Eng. Jul 2021;93:107252. [doi: 10.1016/j.compeleceng.2021.107252]
23. Sebhatu S, Nand P. Intelligent system for diagnosis of pulmonary tuberculosis using XGBoosting method. Presented at: International Conference on Ubiquitous Computing and Intelligent Information Systems; Mar 10-11, 2022:493-511; Tamil Nadu, India.
24. Kotei E, Thirunavukarasu R. A comprehensive review on advancement in deep learning techniques for automatic detection of tuberculosis from chest X-ray images. Arch Computat Methods Eng. Jan 2024;31(1):455-474. [doi: 10.1007/s11831-023-09987-w]
25. Ojasvi Y, Kalpdrum P, Chakresh Kumar J. Using deep learning to classify X-ray images of potential tuberculosis patients. Presented at: 2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM); Dec 2-6, 2018:2368-2375; Madrid, Spain. [doi: 10.1109/BIBM.2018.8621525]
26. Cao YU, Liu C, Liu B, et al. Improving tuberculosis diagnostics using deep learning and mobile health technologies among resource-poor and marginalized communities. Presented at: 2016 IEEE First International Conference on Connected Health; Jun 27-29, 2016:274-281; Washington, DC, United States. [doi: 10.1109/CHASE.2016.18]
27. Panicker RO, Kalmady KS, Rajan J, Sabu MK. Automatic detection of tuberculosis bacilli from microscopic sputum smear images using deep learning methods. Biocybern Biomed Eng. 2018;38(3):691-699. [doi: 10.1016/j.bbe.2018.05.007]
28. Stirenko S, Kochura Y, Alienin O, et al. Chest X-ray analysis of tuberculosis by deep learning with segmentation and augmentation. Presented at: 2018 IEEE 38th International Conference on Electronics and Nanotechnology (ELNANO); Apr 24-26, 2018:422-428; Kyiv, Ukraine. [doi: 10.1109/ELNANO.2018.8477564]
29. Nijiati M, Zhou R, Damaola M, et al. Deep learning based CT images automatic analysis model for active/non-active pulmonary tuberculosis differential diagnosis. Front Mol Biosci. 2022;9:1086047. [doi: 10.3389/fmolb.2022.1086047] [Medline: 36545511]
30. Meraj SS, Yaakob R, Azman A, et al. Detection of pulmonary tuberculosis manifestation in chest X-rays using different convolutional neural network (CNN) models. Int J Eng Adv Technol. 2019;9(1):2270-2275. [doi: 10.35940/ijeat.A2632.109119]
31. Jaeger S, Candemir S, Antani S, Wáng YXJ, Lu PX, Thoma G. Two public chest X-ray datasets for computer-aided screening of pulmonary diseases. Quant Imaging Med Surg. Dec 2014;4(6):475-477. [doi: 10.3978/j.issn.2223-4292.2014.11.20] [Medline: 25525580]
32. Rahman T, Khandakar A, Kadir MA, et al. Reliable tuberculosis detection using chest X-ray with deep learning, segmentation and visualization. IEEE Access. 2020;8:191586-191601. [doi: 10.1109/ACCESS.2020.3031384]
33. Rajan S, Sowmya V, Govind D, Soman KP. Dependency of various color and intensity planes on CNN based image classification. Presented at: Advances in Signal Processing and Intelligent Recognition Systems: Proceedings of Third International Symposium on Signal Processing and Intelligent Recognition Systems (SIRS-2017); Sep 13-16, 2017:1167-1177; Manipal, India.
34. Araújo T, Aresta G, Castro E, et al. Classification of breast cancer histology images using convolutional neural networks. PLoS ONE. 2017;12(6):e0177544. [doi: 10.1371/journal.pone.0177544] [Medline: 28570557]

35. Shibani S, Tsipras D, Ilyas A, Madry A. How does batch normalization help optimization. Presented at: Advances in Neural Information Processing Systems 31 (NeurIPS 2018); Dec 3-8, 2018; Montréal, Canada. Apr 15, 2019.[doi: 10.48550/arXiv.1805.11604]
36. Taylor L, Nitschke G. Improving deep learning with generic data augmentation. Presented at: 2018 IEEE Symposium Series on Computational Intelligence (SSCI); Nov 18-21, 2018:1542-1547; Bangalore, India. [doi: 10.1109/SSCI.2018.8628742]
37. Russakovsky O, Deng J, Su H, et al. ImageNet Large Scale Visual Recognition Challenge. Int J Comput Vis. Dec 2015;115(3):211-252. [doi: 10.1007/s11263-015-0816-y]
38. Antonio G, Sujit P. Deep Learning with Keras. Packt Publishing Ltd; 2017.
39. Mahbub H, Bird Jordan J, Faria Diego R. A study on CNN transfer learning for image classification. Presented at: Advances in Computational Intelligence Systems: Contributions Presented at the 18th UK Workshop on Computational Intelligence; Sep 5-7, 2018:191-202; Nottingham, United Kingdom.
40. Karen S, Andrew Z. Very deep convolutional networks for large-scale image recognition. arXiv. Preprint posted online on Apr 10, 2015. [doi: 10.48550/arXiv.1409.1556]
41. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. Presented at: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); Jun 27-30, 2016:770-778; Las Vegas, NV, United States. [doi: 10.1109/CVPR.2016.90]
42. Kaiming H, Xiangyu Z. Identity mappings in deep residual networks. Presented at: Computer Vision–ECCV 2016: 14th European Conference; Oct 11-14, 2016:630-645; Amsterdam, The Netherlands.
43. Szegedy C, Liu W, Jia Y, et al. Going deeper with convolutions. Presented at: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); Jun 7-12, 2015; Boston, MA, United States. [doi: 10.1109/CVPR.2015.7298594]
44. Christian S, Vincent V, Sergey I, Jon S. Rethinking the inception architecture for computer vision. Presented at: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); Jun 27-30, 2016; Las Vegas, NV, United States.
45. Neshat M, Ahmed M, Askari H, Thilakaratne M, Mirjalili S. Hybrid Inception architecture with residual connection: fine-tuned Inception-ResNet deep learning model for lung inflammation diagnosis from chest radiographs. Procedia Comput Sci. 2024;235:1841-1850. [doi: 10.1016/j.procs.2024.04.175]
46. Christian S, SergeyI, VincentV, AlexA. Inception-v4, Inception-ResNet and the impact of residual connections on learning. Presented at: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence; Feb 4-9, 2017:1; San Francisco, CA, United States. 2017.[doi: 10.1609/aaai.v31i1.11231]
47. Shorten C, Khoshgoftaar TM. A survey on image data augmentation for deep learning. J Big Data. Dec 2019;6(1):1-48. [doi: 10.1186/s40537-019-0197-0]
48. Perez L, Wang J. The effectiveness of data augmentation in image classification using deep learning. arXiv. Preprint posted online on Dec 13, 2017. [doi: 10.48550/arXiv.1712.04621]

Abbreviations

AI: artificial intelligence
AUC-ROC: area under the receiver operating characteristic curve
CNN: convolutional neural network
CXR: chest x-ray
FN: false negative
FP: false positive
FPR: false positive rate
ILSVRC: ImageNet Large-Scale Visual Recognition Challenge
ReLU: rectified linear unit
ResNet: Residual Network
TB: tuberculosis
TN: true negative
TP: true positive
TPR: true positive rate
VGG: Visual Geometry Group
WHO: World Health Organization

Edited by Saeed Amal; peer-reviewed by Natthapong Nanthasamroeng, Rapeepan Pitakaso; submitted 02.09.2024; final revised version received 27.03.2025; accepted 16.04.2025; published 01.07.2025

Please cite as:
Mirugwe A, Tamale L, Nyirenda J
Improving Tuberculosis Detection in Chest X-Ray Images Through Transfer Learning and Deep Learning: Comparative Study of Convolutional Neural Network Architectures
JMIRx Med 2025;6:e66029
URL: https://med.jmirx.org/2025/1/e66029
doi: 10.2196/66029