# Truncating the EfficientNet-B0 for Computer Aided Diagnosis of Tuberculosis

Noah Anderson * and Nazmul Shahadat

Posted Date: 11 May 2026

doi: 10.20944/preprints202605.0663.v1

Keywords: convolutional neural networks; computer aided diagnosis (CAD); tuberculosis

 doi:10.20944/preprints202605.0663.v1



*Article*

# Truncating the EfficientNet-B0 for Computer Aided Diagnosis of Tuberculosis

**Noah Anderson * and Nazmul Shahadat**

Truman State University
* Correspondence: noahanderson6556@gmail.com

**Abstract**

This study investigates the effectiveness of truncating the EfficientNet-B0 architecture for the computer-aided diagnosis of tuberculosis (TB) on chest radiograph (CXR) images. A series of truncated EfficientNet-B0 models are proposed, systematically removing blocks to reduce model complexity while maintaining diagnostic accuracy. The B0(-3) model, which eliminates three blocks, emerges as a highly efficient configuration, achieving 100% internal test accuracy on the Kaggle dataset and demonstrating robust generalization to an external Mendeley dataset. Bootstrap analysis reveals that the B0(-3) model achieves a mean accuracy of 97.38% (95% CI: 96.94%–97.84%) on the external dataset, with performance statistically overlapping that of the complete B0(-0) model (97.24%, 95% CI: 96.78%–97.69%). Despite this overlap, the B0(-3) model uses 13 times fewer parameters, making it a more efficient alternative without sacrificing accuracy. These results highlight the potential of model truncation to improve efficiency while maintaining performance, positioning B0(-3) as a promising candidate for real-world TB detection.

**Keywords:** convolutional neural networks; computer aided diagnosis (CAD); tuberculosis

## 1. Introduction

In 2023, tuberculosis (TB) claimed 1.25 million lives, marking its return as the world's leading cause of death from a single infectious agent after being surpassed by COVID-19 for three years [1]. TB remains a major global health challenge, particularly in low- and middle-income countries, where 98% of the disease burden is concentrated [2,3]. Despite the availability of effective treatments since the development of the first antibiotic for tuberculosis in 1944, the disease persists due to systemic issues such as global health inequities, resource shortages, and the rise of drug-resistant strains [4,5].

TB is caused by the bacterium *Mycobacterium tuberculosis*, which mainly affects the lungs. Approximately one-third of the world's population has latent tuberculosis, posing a significant risk of reactivation and transmission [6]. Although effective treatments exist, their accessibility is hindered by financial and logistical barriers. In 2023, global investments in TB prevention and care totaled $5.7 billion, falling far short of the $22 billion required to meet the 2027 target for TB elimination [3]. This funding gap disproportionately affects vulnerable populations in regions marked by poverty, conflict, and displacement, such as India, Indonesia, China, the Philippines, Pakistan, Nigeria, Bangladesh, and the Democratic Republic of the Congo [5,7].

While addressing these systemic inequities is critical, this paper focuses on a complementary approach to combating TB: improving diagnostic tools. Early and accurate detection is essential for effective TB control, and chest X-rays (CXRs) have emerged as a key component of TB triage due to their high sensitivity [8]. However, the shortage of trained radiologists in many high-burden regions creates a bottleneck in CXR-based diagnosis. This challenge has spurred the development of computer-aided diagnosis (CAD) systems, which can assist in interpreting CXRs and improve diagnostic efficiency in resource-constrained settings.

 doi:10.20944/preprints202605.0663.v1



The World Health Organization (WHO) sets stringent performance standards for TB diagnostic tools, requiring a minimum sensitivity of 90% and specificity of 70%, with ideal targets being 5% higher for both metrics [9]. Among existing CAD systems, CAD4TB is one of the most widely used. The latest version, CAD4TBv7, achieves a sensitivity of 89.8% (95% CI: 83.4%–94.3%) and a specificity of 68.2% (95% CI: 65.4%–71.0%), with the WHO's minimum requirements being within the confidence intervals but clearly leaves room for improvement [10]. This study explores the potential of truncated EfficientNet-B0 models to advance research on improving efficiency and accuracy in CAD systems for TB detection in CXRs, while acknowledging the limitations of the dataset, which focuses on binary classification (TB vs. normal) and does not fully represent the complexities of real-world diagnostic scenarios.

It is important to emphasize that CAD systems are not a panacea for the TB crisis. Although they can alleviate some diagnostic challenges, they cannot address the increased risk of tuberculosis transmission and activation from factors such as poverty, inequitable resource allocation, and inadequate healthcare infrastructure. However, by improving diagnostic accuracy and efficiency, CAD systems can play a vital role in reducing the global burden of tuberculosis, particularly in regions where radiologists are in short supply. This study seeks to contribute to this effort by proposing a truncated EfficientNet-B0 model that balances performance and efficiency, offering a practical tool for TB diagnosis in resource-constrained settings.

As will be seen in the following subsection (Section 1.1), there is a significant gap between the high-performance reported in deep learning (DL) research for TB detection and the real-world generalizability of these models. Many studies achieved impressive results on controlled datasets but failed to address critical issues such as dataset bias, overfitting, and the Clever Hans effect, where models learn spurious correlations rather than clinically relevant features. These challenges will be explored in detail in Section 1.2.

### *1.1. TB Deep Learning Literature*

Deep learning for active tuberculosis (TB) classification is a well-studied topic, with many studies reporting high accuracy. This can involve binary classification (e.g., TB vs. healthy lungs) or multiclass differentiation (e.g., TB vs. pneumonia vs. COVID-19). This section reviews the literature, highlighting the state-of-the-art performance of ImageNet-pretrained models, the efficiency of EfficientNet, and the understudied potential of truncating pre-trained models for TB detection. These themes reveal a critical gap in the literature: the need to explore how much-pretrained models like EfficientNet-B0 can be truncated without sacrificing performance, particularly for binary TB classification.

ImageNet-pretrained models consistently outperform custom architectures in TB classification, demonstrating higher accuracy and better generalizability. Rahman et al. [11], who compiled the Kaggle dataset used in this study, proposed a two-step approach using U-Net for lung segmentation followed by DenseNet-201, continuing approximately $20M$ parameters [12] for classification. Their method achieved 98.6% accuracy, with pre-trained models consistently outperforming non-pre-trained ones. Similarly, Alshmrani et al. [13] deployed a hybrid model combining VGG19, another ImageNet-pretrained model, with a custom convolutional neural network (CNN) feature extractor. While their overall accuracy was 97.56%, their TB-specific accuracy was significantly lower at 67.3%. These results underscore the importance of leveraging pretrained models, which benefit from transfer learning and robust feature extraction capabilities.

Other studies have explored diverse approaches to TB classification, demonstrating the versatility of deep learning for this task. Ahmed et al. [14] addressed multi-class classification, including TB, pneumonia, COVID-19, and normal cases, achieving 98.72% accuracy for TB classification. Venkataramana et al. [15] developed a multilevel classification system, first distinguishing TB from pneumonia and then further classifying pneumonia subtypes. Their approach achieved 97.4% accuracy for TB classification. Goswami et al. [16] used a subset of the Kaggle dataset, achieving 94% accuracy with an unspecified CNN. While these studies highlight the potential of deep learning for TB detection, they also illustrate the challenges of multi-class systems, such as increased computational demands.

 doi:10.20944/preprints202605.0663.v1



Among ImageNet-pretrained models, EfficientNets stand out for their balance of accuracy and computational efficiency. Kaur et al. [17] employed the EfficientNet-B3, a larger variant of the B0 model used in this study, on a subset of the Kaggle dataset [18]. Despite using an imbalanced dataset, their experiments achieved an impressive 99% accuracy. The EfficientNet-B3 contains approximately 12 million parameters [19], making it relatively efficient compared to other high-performing models. Similarly, Bhosale et al. [20] evaluated the EfficientNet-B7 on the Shenzhen and Montgomery datasets, achieving 98.5% accuracy. However, the B7 model contains 66 million parameters [19], making it the least efficient variant of the EfficientNet family. These studies demonstrate that EfficientNets can achieve state-of-the-art performance while maintaining reasonable efficiency, though the trade-offs between model size and accuracy warrant further investigation.

Recent research suggests that even greater efficiency can be achieved through truncation—removing layers from pretrained models without sacrificing performance. Montalbo [21] explored truncation in several ImageNet-pretrained models, including InceptionResNetV2, ResNet50V2, and EfficientNet-B0. Their truncated InceptionResNetV2 achieved 97.41% accuracy with just 441K parameters, compared to 98.67% accuracy for the full model with 55 million parameters. They also experimented with truncating EfficientNet-B0, modifying the classification layer by increasing the dropout probability from 0.2 to 0.5 and adding a softmax activation layer. This reduced the model to 24,345 parameters, making it 208 times smaller than the base B0. However, the truncated EfficientNet-B0 achieved only 86.03% accuracy, suggesting that such a dramatic reduction may have compromised performance. This result underscores the need for further investigation into the optimal truncation level for EfficientNet-B0.

Ke et al. [22] further investigated truncation in ImageNet-pretrained models using the CheXpert dataset, a large dataset consisting of 224,316 images across 15 classes. Unfortunately, TB is not one of these classes, limiting the direct applicability of their findings to TB detection. They evaluated a wide range of models, including DenseNet121, DenseNet169, DenseNet201, EfficientNetB0, EfficientNetB1, EfficientNetB2, EfficientNetB3, InceptionV3, InceptionV4, MNASNet, MobileNetV2, MobileNetV3, ResNet101, ResNet18, ResNet34, and ResNet50. Their results confirmed that pretrained weights significantly outperformed non-pretrained models, reinforcing the value of transfer learning for CXR analysis.

For truncated models, Ke et al. performed a rigorous evaluation using 1,000 bootstrap iterations to construct confidence intervals [22]. They studied truncated versions of DenseNet121 [12], MNASNet [23], ResNet18 [24], and EfficientNetB0 [19]. For EfficientNetB0, they removed up to two blocks, while for the other models, they removed up to four blocks. They found no significant drop in performance when truncating the final block of EfficientNetB0. However, when truncating the second block, only EfficientNetB0 and ResNet18 maintained performance without significant degradation. In contrast, MNASNet, DenseNet121, and ResNet18 all experienced significant performance drops after removing more than two blocks. Notably, Ke et al. did not explore truncating EfficientNetB0 beyond two blocks, leaving a critical gap in the literature. This gap is particularly relevant for this study, as each block in EfficientNetB0 has a unique structure, and removing different blocks may affect performance differently. Therefore, further exploration of truncation in EfficientNetB0 is warranted.

Ke et al. concluded that ImageNet architectures may be unnecessarily large for CXR analysis and that models can be made 3.25 times more parameter-efficient on average without a statistically significant drop in performance [22]. This finding underscores the potential of truncation to improve efficiency while maintaining accuracy, a key focus of this study.

The literature demonstrates that ImageNet-pretrained models consistently achieve state-of-the-art accuracy in TB classification. EfficientNets are emerging as top performers due to their balance of efficiency and predictive power. However, there is a critical gap in the research: too little attention has been paid to how large these pretrained models need to be. Promising studies on truncation, such as those by Montalbo [21] and Ke et al. [22], suggest that significant reductions in model size are possible without sacrificing performance. This study addresses this gap by thoroughly exploring

 doi:10.20944/preprints202605.0663.v1



the truncation of EfficientNet-B0 for binary TB classification, aiming to achieve maximum efficiency without compromising accuracy.

### *1.2. Clever Hans Effect*

The Clever Hans effect refers to the phenomenon where models learn spurious correlations or "shortcuts" in the data rather than clinically relevant features for classification. The term originates from a German horse in the early 20th century that appeared to perform arithmetic but was actually responding to subtle cues from its trainer. In computer vision, this effect is particularly problematic in radiological deep learning, where models may rely on non-clinically relevant features, such as text labels, hospital markings, or imaging artifacts, to make predictions.

Vásquez-Venegas et al. highlight that the Clever Hans effect is endemic to the field of radiological deep learning, with a majority of studies failing to address or even acknowledge the potential for such shortcuts [25]. Degrave et al. further emphasize this issue, demonstrating that many models fail to generalize to datasets collected from external sources, underscoring the importance of robust validation practices [26].

Though they remain underutilized, several techniques have been proposed to mitigate the Clever Hans effect. These include:

- **Feature Decoupling and Regularization**: Techniques like L2 regularization can help reduce overfitting to spurious features.
- **Shortcut Removal with Attention and Heatmaps**: Visualizing model attention can reveal reliance on non-clinically relevant features, such as text labels or imaging artifacts.
- **Lung Masking**: Extracting lung regions from CXR ensures that the models focus on clinically relevant features. For example, Rahman et al. used lung segmentation to improve model performance on the Kaggle dataset [11].
- **Text and Artifact Removal**: Automated frameworks, such as YOLOv3 developed by Pedrosa et al., can obscure written labels, markers, and equipment identifiers in CXR images, reducing the risk of shortcut learning [27].

For this study, implementing advanced techniques like lung masking or text scrubbing was not feasible due to practical constraints. Rahman et al. did not publish the ground truth lung masks for the Kaggle dataset, making lung segmentation not possible [11]. Similarly, the computational resources required for text scrubbing using YOLOv3 were prohibitive [27]. As a minimum check for generalization, the study evaluated models on both the Kaggle dataset, which contains minimal written markings, and an external dataset from Pakistan compiled by Mendeley [28], as detailed in Section 2. While these steps do not entirely eliminate the risk of the Clever Hans effect, they provide a baseline level of robustness to spurious, dataset-centric correlations.

## 2. Methodology

This section outlines the methodology used to evaluate the performance of truncated EfficientNet-B0 models for tuberculosis (TB) detection in chest X-ray (CXR) images. The study leverages the EfficientNet-B0 architecture, a state-of-the-art convolutional neural network (CNN) known for its efficiency and accuracy. We propose a series of truncated variants of EfficientNet-B0, systematically removing blocks to reduce model complexity while maintaining performance. The methodology is divided into three key components: (1) an in-depth analysis of the EfficientNet-B0 architecture, including its mobile inverted bottleneck (MBConv) layers and compound scaling mechanism; (2) the design and implementation of truncated models, focusing on the trade-offs between efficiency and accuracy; and (3) a rigorous experimental framework for training, validation, and testing using both internal (Kaggle) and external (Mendeley) datasets. This experiment aims to identify the optimal balance between model efficiency and diagnostic accuracy for TB detection in CXR images.

 doi:10.20944/preprints202605.0663.v1



### 2.1. Model Architecture

EfficientNet is a family of pretrained convolutional neural networks (ConvNets) that has achieved state-of-the-art accuracy on the ImageNet dataset. Notably, the largest model in the family, EfficientNet-B7, achieves top-1% accuracy while being 8.4 times smaller and 6.1 times faster to train than previous ConvNets such as ResNet, AlexNet, and MobileNet [19].

EfficientNet introduces a novel approach to model scaling, termed Compound scaling, which uniformly scales three key dimensions of a neural network: width (number of channels), depth (number of layers), and resolution (input image size). Unlike traditional methods that arbitrarily scale one dimension at a time, compound scaling balances all three dimensions using a fixed set of scaling coefficients ($\alpha$, $\beta$, $\gamma$). These coefficients are determined through a grid search on the baseline model (EfficientNet-B0) to optimize accuracy and efficiency. The B-series of EfficientNet models (B0–B7) are constructed by progressively applying compound scaling, with higher-numbered models achieving greater accuracy at the cost of increased computational resources [19].

For this study, we focus on EfficientNet-B0, the baseline model of the EfficientNet family. While compound scaling enables the creation of larger models (e.g., B1–B7), B0 is sufficient for our purposes due to the uniformity of chest X-ray (CXR) images compared to the diverse 1000 classes of the ImageNet dataset. Unlike ImageNet, which contains highly varied images (e.g., baseballs, frogs, landscapes), CXR images exhibit consistent structural patterns, reducing the need for larger, more complex models. Instead, our primary objective is to further reduce the parameter count of EfficientNet-B0 without significantly compromising performance or generalization.

The core building block of EfficientNet-B0 is the mobile inverted bottleneck (MBConv) layer, first introduced in MobileNetV2 [29]. MBConv layers leverage depthwise separable convolutions to achieve significant computational efficiency. Unlike standard convolutions, which simultaneously process spatial and cross-channel relationships, depthwise separable convolutions decouple these operations into two steps:

1. Depthwise Convolution: A single filter is applied independently to each input channel, extracting spatial features without mixing channels. This step drastically reduces the number of multiplications compared to standard convolutions.
2. Pointwise Convolution: A $1 \times 1$ convolution is used to mix information across channels, synthesizing higher-dimensional features. This step is computationally lightweight compared to spatial convolutions.

For example, a standard $5 \times 5$ convolution with 256 filters requires $\sim$1.2 million multiplications, whereas the equivalent depthwise separable convolution requires only $\sim$54,000 multiplications, a 23 times reduction. This factorization of spatial and channel operations minimizes redundant computations while preserving representational power [19,29].

Additionally, MBConv layers incorporate an inverted residual bottleneck structure, enhancing efficiency. This structure first expands the input channels using a $1 \times 1$ convolution, applies depthwise convolution to the expanded feature space, and then compresses the channels back to the original dimension. This design ensures that spatial operations are performed in a high-dimensional space, improving feature extraction while maintaining computational efficiency [23,29].

The EfficientNet-B0 architecture employs two primary variants of the MBConv layer: MBConv1 and MBConv6. Both variants share a similar structure but differ in the inclusion of an expansion step.

- The MBConv1 consists of the following layers:
    1. A $k \times k$ depthwise separable convolution layer, where spatial filtering is applied independently to each input channel, is used.
    2. A batch normalization layer followed by a Swish activation function.
    3. An optional squeeze-and-excitation (SE) layer reweights channels to enhance feature representation.

 doi:10.20944/preprints202605.0663.v1



4. A $1 \times 1$ pointwise convolution layer to project features back to the original channel dimension.
5. A dropout layer mitigates the overfitting problem.

- MBConv6 is structurally similar to MBConv1 but includes an additional expansion step:
  1. A $1 \times 1$ pointwise convolution layer expands the input channels by a factor of 6 (e.g., 32 channels $\rightarrow$ 192 channels).
  2. A $k \times k$ depthwise separable convolution layer is applied to the expanded feature space.
  3. A batch normalization layer followed by a Swish activation function is used.
  4. An optional squeeze-and-excitation (SE) layer is introduced.
  5. A $1 \times 1$ pointwise convolution layer projects features back to the original channel dimension.
  6. A dropout layer mitigates the overfitting problem.

The key difference between MBConv1 and MBConv6 lies in the expansion step. MBConv6 first expands the input channels, allowing the depthwise convolution to operate in a higher-dimensional space, which enhances feature representation. In contrast, MBConv1 skips this step, making it more computationally efficient but less expressive. The MBConv 1 and 6 are visualized in Figure 1.

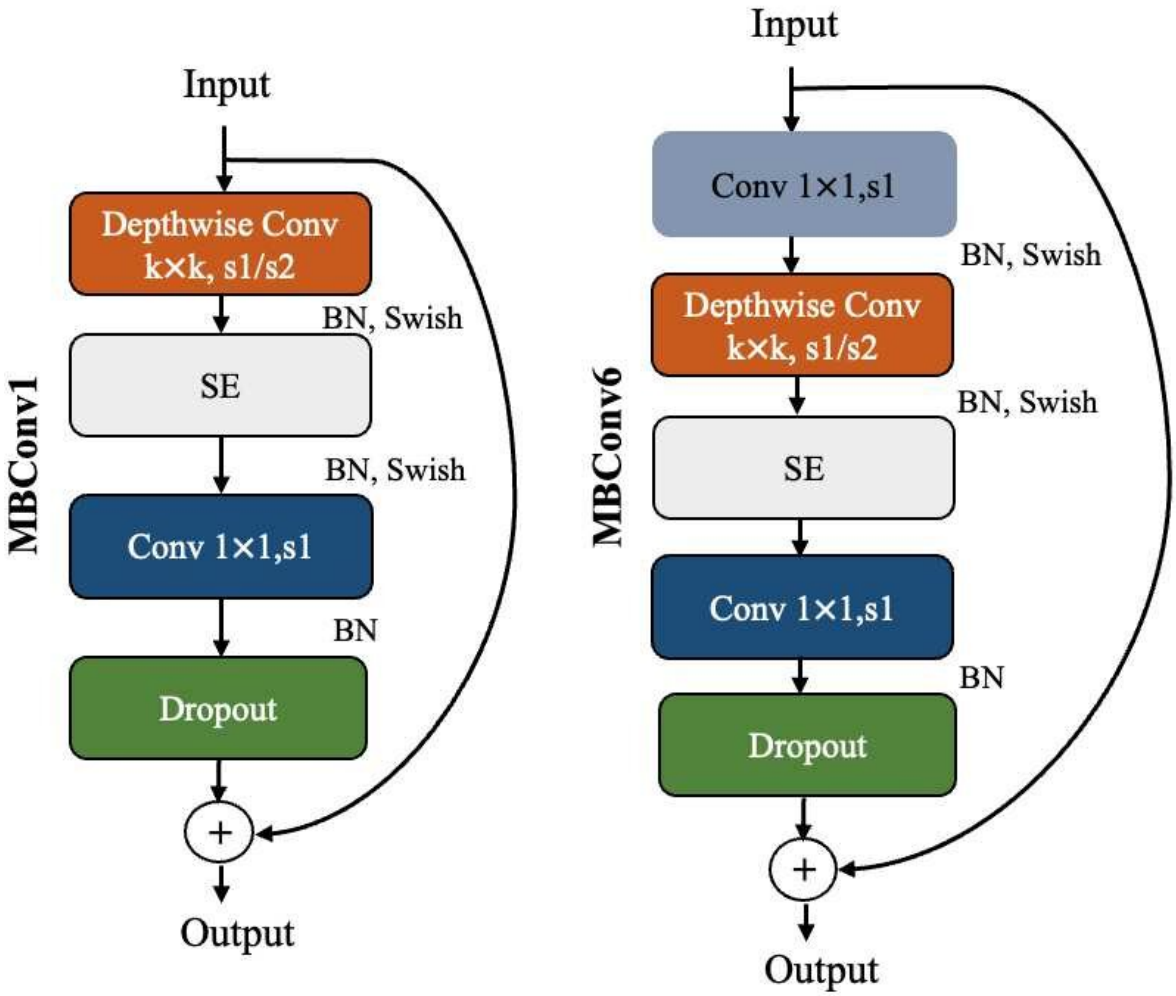


**Figure 1.** MBConv1 (Left) and MBConv6 (Right) layer architectures [30].

The full EfficientNet-B0 architecture, as shown in Table 1, consists of 10 blocks [19]. The first block is a $3 \times 3$ convolutional layer with batch normalization and Swish activation, followed by block 2, which is an MBConv1 layer with a $3 \times 3$ kernel and a single layer. The remaining blocks primarily consist of MBConv6 layers with either $3 \times 3$ or $5 \times 5$ kernels, each containing varying numbers of layers. Block 9 serves as a pointwise expansion layer, increasing the channel dimension to 320. Finally,

 doi:10.20944/preprints202605.0663.v1



block 10 includes a $1 \times 1$ convolution, global average pooling, and a fully connected classification layer, along with a dropout layer with a probability of 0.2 to prevent overfitting.

**Table 1.** Full sized EfficientNet-B0 with non-pretrained fully connected layer [19].

| Block | Operator | Resolution | #Channels | #Layers | Pretrained Weights |
|---|---|---|---|---|---|
| 1 | Conv3x3 | $224 \times 224$ | 32 | 1 | True |
| 2 | MBConv1, k3x3 | $112 \times 112$ | 16 | 1 | True |
| 3 | MBConv6, k3x3 | $112 \times 112$ | 24 | 2 | True |
| 4 | MBConv6, k5x5 | $56 \times 56$ | 40 | 2 | True |
| 5 | MBConv6, k3x3 | $28 \times 28$ | 80 | 3 | True |
| 6 | MBConv6, k5x5 | $14 \times 14$ | 112 | 3 | True |
| 7 | MBConv6, k5x5 | $14 \times 14$ | 192 | 4 | True |
| 8 | MBConv6, k3x3 | $7 \times 7$ | 320 | 1 | True |
| 9 | Conv1x1 | $7 \times 7$ | 1280 | 1 | True |
| 10 | Pooling & FC | $7 \times 7$ | 1280 | 1 | False |

#### 2.1.1. Proposed Truncated EfficientNet-B0

The proposed model, EfficientNet-B0(-3), is a truncated variant of EfficientNet-B0 in which blocks 6, 7, and 8 are removed. To replace the functionality of the missing layers, a new untrained Conv1 $\times$ 1 pointwise block with 112 output channels is introduced at the end of block 6. This serves as a substitute for the original pointwise expansion layer, which had 1280 output channels. The reduction in output channels from 1280 to 112 aligns with the expected dimensionality of the removed layers, ensuring a more gradual transition in feature extraction.

Because EfficientNet-B0's architecture does not allow direct reconnection to the final classification layers after truncation, the original Conv1 $\times$ 1 layer cannot simply be retained. Instead, it is redefined from scratch to serve the same role while maintaining the adjusted output dimensionality. While the base model's Conv1 $\times$ 1 block incorporates batch normalization and a Swish activation function, the redefined version adopts the same structure but with fewer output channels to match the preceding truncated layers.

Table 2 shows that the truncated architecture consists of only six blocks. Despite the reduction in depth, the model achieves performance within the same range as the full EfficientNet-B0, with the truncated model even achieving a slightly higher mean accuracy. This is supported by overlapping bootstrap confidence intervals 95%, as demonstrated in the results section. In particular, the truncated model has 13 times fewer parameters (319,286) compared to the original B0 (4,171,774), making it significantly more efficient with no reduction in accuracy.

**Table 2.** Proposed EfficientNet-B0(-3) architecture.

| Block weights | Operator | Resolution | #Channels | #Layers | Pretrained |
|---|---|---|---|---|---|
| 1 | Conv3x3 | $224 \times 224$ | 32 | 1 | True |
| 2 | MBConv1, k3x3 | $112 \times 112$ | 16 | 1 | True |
| 3 | MBConv6, k3x3 | $112 \times 112$ | 24 | 2 | True |
| 4 | MBConv6, k5x5 | $56 \times 56$ | 40 | 2 | True |
| 5 | MBConv6, k3x3 | $28 \times 28$ | 80 | 3 | True |
| 6 | Conv1x1 | $7 \times 7$ | 112 | 1 | False |
| 7 | Pooling & FC | $7 \times 7$ | 112 | 1 | False |

### *2.2. Data*

The experimental design utilizes two distinct datasets. The primary dataset, sourced from Kaggle and compiled by Rahman et al. [18], is used for traditional training, validation, and testing. This

 doi:10.20944/preprints202605.0663.v1



dataset comprises 3,500 TB-positive and 3,500 healthy lung chest X-ray (CXR) images. It is aggregated from four sources:

- The National Institute of Allergy and Infectious Diseases (NAID), providing TB-positive CXR images [31].
- The Ministry of Health of Belarus, collected by NIAD, also containing TB-positive CXR images [32].
- The RSNA Pneumonia Detection Challenge dataset on Kaggle, contributing healthy lung CXRs [33].
- A dataset from Jaeger et al., providing additional healthy CXR images [34].

To further evaluate model generalization, an external test set was constructed using two Mendeley datasets:

- A dataset from Kiran & Jabeen [28], containing 2494 TB-positive and 514 healthy CXR images from Pakistan.
- A dataset from Kumar [35], contributing 1802 healthy CXR images to balance the external evaluation set.

The Kaggle dataset [18] was chosen for training due to its cleaner images, which exhibit minimal written markings. In contrast, the Pakistani Mendeley dataset [28] includes numerous TB-positive images with Urdu text annotations, as shown in Figure 2. This association between TB-positive images and Urdu text poses a significant risk of the model learning Clever Hans shortcuts—relying on non-clinically relevant features (e.g., the presence of Urdu text) rather than actual pathological indicators for classification. Training on a dataset free of such artifacts makes the model more likely to learn clinically relevant features for TB classification. Using the Mendeley dataset for external validation provides a realistic test of the model's ability to generalize to data encountered in clinical settings, while highlighting potential pitfalls of shortcut learning in real-world applications.

**Figure 2.** TB-positive lung CXR with Urdu writing annotations. The presence of Urdu text in TB-positive images may lead to Clever Hans shortcuts, where the model learns to associate text markings with TB rather than pathological features [28].

### *2.3. Data Augmentation*

No data augmentation techniques, such as Gaussian Blur, random flips, rotations, or jittering of brightness and contrast, were used. The model generalized well against both the internal test data from Kaggle and the external test set from Mendeley, but future generalization tests against different

 doi:10.20944/preprints202605.0663.v1



datasets may require data augmentation, so the publicly available code base for this paper is set up to experiment with different augmentation techniques.

### *2.4. Experimental Design*

A total of 5 models are trained and evaluated, including the full EfficientNet-B0 and four truncated variants, each with an increasing number of blocks removed (from 1 to 4). The experiment is divided into three main stages:

#### 2.4.1. Stage 1: Internal Training and Validation

The Kaggle dataset is split into training, validation, and test sets using an 80:10:10 ratio. Each model is trained for 40 epochs on the training set, with validation performance evaluated at the end of each epoch. After training, the models are evaluated on the Kaggle test set, which is held out and not used during training. Throughout model training, a learning rate scheduler reduces by a factor of 0.1 on plateaus with a patience of 5. The initial learning rate is 0.001. The model also uses the Adam optimizer. The following metrics are computed: cross-entropy loss, accuracy, sensitivity, and specificity.

#### 2.4.2. Stage 2: Generalization to External Data

The models are tested on an external dataset composed of the combined Mendeley datasets to evaluate generalization. This dataset contains less tidy images, including those with Urdu text annotations, simulating real-world clinical data. While this is not a substitute for clinical testing, it is a more rigorous evaluation of the model's generalization ability. Performance in this external dataset is considered the primary indicator of model robustness, surpassing the Kaggle test set results in importance.

#### 2.4.3. Stage 3: Bootstrapping for Robust Evaluation

To ensure a rigorous comparison, 1,000 bootstrap iterations are performed on the external Mendeley test set. Bootstrapping involves randomly resampling the dataset with replacement and evaluating the model on each resampled set. This process constructs 95% confidence intervals for the performance metrics, providing a robust estimate of model performance and variability. Bootstrapping is particularly relevant here because it accounts for the inherent variability in the dataset and provides a more reliable measure of model performance than a single test set evaluation. Due to computational constraints, bootstrapping is applied only to the top-performing models identified in Stage 2.

## 3. Results

This section presents the experimental results of evaluating the truncated EfficientNet-B0 models, focusing on their performance on internal (Kaggle) and external (Mendeley) datasets. The internal test results demonstrate perfect accuracy for most models, with several achieving 100% accuracy on the Kaggle dataset. However, external testing on the Mendeley dataset reveals significant variability in model performance, highlighting the importance of evaluating generalization to real-world, heterogeneous data. Bootstrap analysis further refines the comparison, showing that the B0(-3) model achieves the highest mean accuracy while being 13 times more efficient than the full B0(-0) model. Additionally, training and validation curves confirm the stability and convergence of the top-performing model. Together, these results underscore the trade-offs between model complexity, efficiency, and generalization, with the B0(-4) emerging as the optimal choice.

### *3.1. Internal Test Results*

The internal test results for the Kaggle dataset, shown in Table 3, demonstrate impressive accuracy across all truncated models. All models achieved 100% accuracy on the unseen internal test set, outperforming the previous top performance on the Kaggle dataset of 98.6% achieved by Rahman et al. [11]. These results suggest that up to four blocks can be removed from the EfficientNet-B0 architecture

 doi:10.20944/preprints202605.0663.v1



without compromising performance, at least on the internal test set. However, caution is advised, as high performance on the Kaggle dataset does not guarantee robust generalization to external datasets, as demonstrated in the following section.

**Table 3.** Internal Test Model Metrics.

| Truncated Blocks | Test Accuracy | Test Loss | Sensitivity (%) | Specificity (%) |
|---|---|---|---|---|
| 0 | 100.00 | $5.053 \times 10^{-4}$ | 100.00 | 100.00 |
| 1 | 100.00 | $2.600 \times 10^{-5}$ | 100.00 | 100.00 |
| 2 | 100.00 | $5.106 \times 10^{-4}$ | 100.00 | 100.00 |
| 3 | 100.00 | $7.710 \times 10^{-5}$ | 100.00 | 100.00 |
| 4 | 100.00 | $1.263 \times 10^{-4}$ | 100.00 | 100.00 |

### *3.2. External Test Results*

The external test results, shown in Table 4 and visualized in Figure 3, present a less optimistic picture for the B0(-1) and B0(-4) models. When evaluated on the Mendeley dataset, the top-performing models are the B0(-0), B0(-2), and B0(-3). The B0(-1) model shows a dramatic decline in performance, with a 22.18% reduction in accuracy and a specificity of 54.92%, falling outside the range accepted by the World Health Organization's guidelines of 70%.

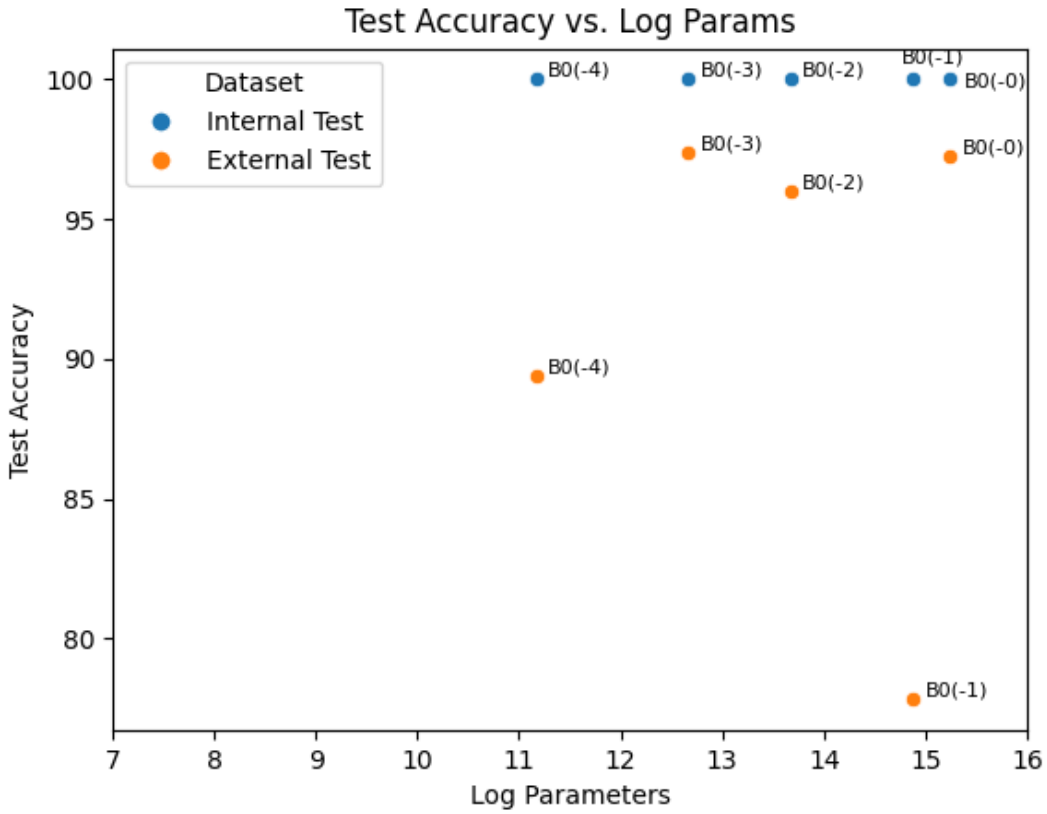


**Figure 3.** Test Accuracy is plotted here against log parameters. You can see the divergence in the B0(-1) and the B0(-4) in internal and external test accuracy. The B0(-3) stands out along with the B0(-0) as having the smallest gap between internal and external accuracy, but with B0(-3) having far fewer parameters.

 doi:10.20944/preprints202605.0663.v1



**Table 4.** External Test Model Metrics.

| Truncated Layers | External Test Accuracy | External Test Loss | External Sensitivity | External Specificity |
|---|---|---|---|---|
| 0 | 97.26 | 0.0906 | 98.72 | 95.68 |
| 1 | 77.82 | 0.7556 | 99.07 | 54.92 |
| 2 | 96.02 | 0.1025 | 96.07 | 95.98 |
| 3 | 97.38 | 0.0721 | 98.96 | 95.68 |
| 4 | 89.42 | 0.4626 | 93.95 | 84.54 |

The underperformance of the B0(-1) model may be attributed to the output channel size of the final Conv1x1 layer, which was set to 112. In the base model, the Conv1x1 layer maps to a much higher feature space with 1,280 output channels, which is used in conjunction with batch normalization. For the truncated models, the output channels were reduced to avoid a sudden jump from 192 channels to 1280 for the B0(-2) or from 80 to 1280 for the B0(-4). While a better choice of output channels for the Conv1x1 layer might improve the generalization of the B0(-1), this was not the primary focus of this study, which prioritizes optimizing performance for more heavily truncated models.

Among the top three models—B0(-0), B0(-2), and B0(-3)—the B0(-3) stands out with the highest accuracy (97.38%) and sensitivity (98.96%). High sensitivity is particularly important per the WHO guidelines, as there is a greater risk of missing positive TB diagnoses. The B0(-2) was not explored in the bootstrap results section for two reasons: (1) its lower performance, although it may not be statistically significantly worse, and (2) the study's focus on reducing parameters without sacrificing accuracy. The B0(-3) appears to be more accurate and efficient, making it a more compelling candidate. The B0(-0) and B0(-3) were examined in the bootstrap section to determine if there is a statistically significant difference between the two models.

### *3.3. External Bootstrap Results*

To rigorously compare the best-performing models, 1,000 bootstrap samples were generated, and the 95% confidence intervals (CI) were calculated for the external Mendeley dataset. As shown in Table 5, the B0(-3) model achieves the highest mean accuracy, with the B0(-0) close behind with overlapping confidence intervals. Given these results, the B0(-3) is the best-performing model overall. While its performance statistically overlaps with the full B0(-0), it is chosen as the superior model due to its 13 times fewer parameters, representing a significant leap in efficiency without sacrificing accuracy.

**Table 5.** Bootstrap Accuracies.

| **Metric** | **Truncated Blocks = 0** | **Truncated Blocks = 3** |
|---|---|---|
| Accuracy (%) | 97.24 CI(96.78, 97.69) | 97.39 CI(96.94, 97.84) |
| Sensitivity (%) | 98.71 CI(98.23, 99.16) | 98.97 CI(98.53, 99.32) |
| Specificity (%) | 95.67 CI(94.82, 96.46) | 95.70 CI(94.87, 96.48) |

### *3.4. Epoch Training and Validation*

To demonstrate model convergence for the top-performing model, B0(-3), the training and validation loss and accuracy are plotted over 40 epochs. As shown in Figure 4, the training loss decreases rapidly in the early epochs and stabilizes around epoch 10, indicating that the model has effectively learned the underlying patterns in the training data. Similarly, the validation loss follows a comparable trend, suggesting that the model generalizes well to unseen data during training.

 doi:10.20944/preprints202605.0663.v1



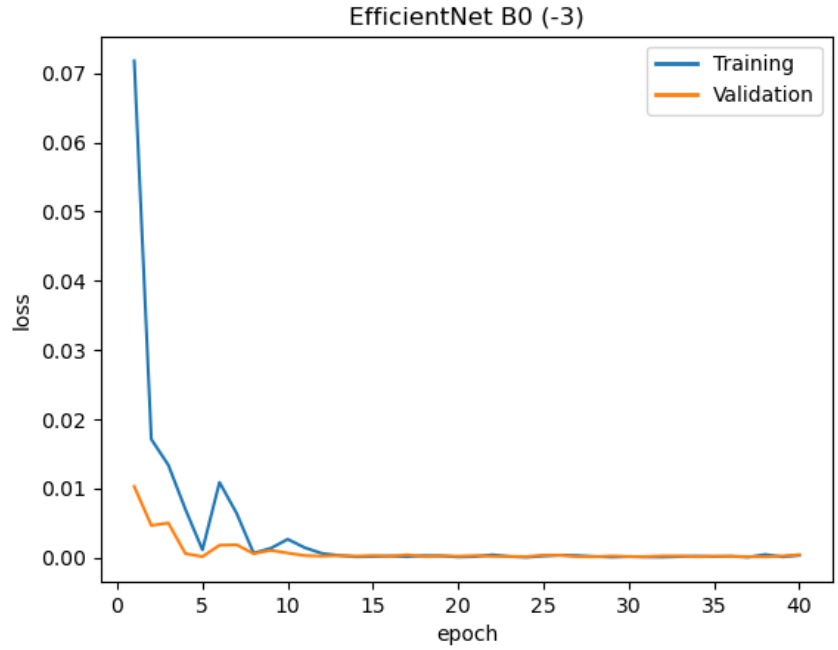


**Figure 4.** Training and validation loss for the B0(-3) model showing good model saturation after 10 epochs.

Figure 5 shows the training and validation accuracy over the same 40 epochs. Both curves exhibit a steady increase, with the training accuracy approaching 100% and the validation accuracy stabilizing at a high level. This behavior confirms that the B0(-3) model achieves strong performance without overfitting, as evidenced by the close alignment between training and validation accuracy. The convergence of both loss and accuracy metrics around epoch 10 supports the choice of 40 epochs as a sufficient training duration for this task.

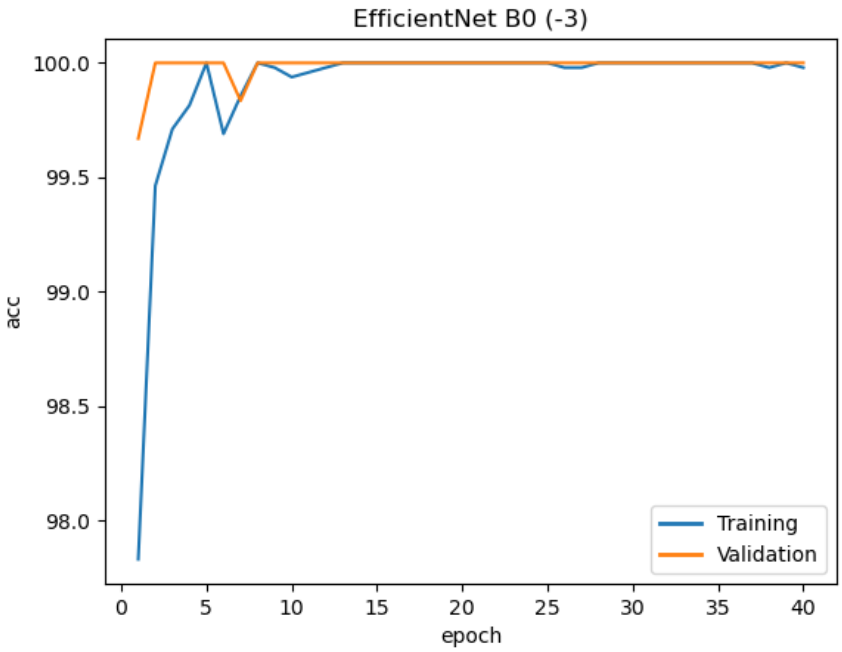


**Figure 5.** Training and validation accuracy for the B0(-3) model showing good model saturation after 10 epochs.

Figures 6–12 show the epoch loss and accuracy for the remaining models, with most exhibiting similar trends. Notably, the B0(-1) model shows a large spike in validation loss at epoch 24, as seen in

 doi:10.20944/preprints202605.0663.v1



Figure 8. This anomaly may indicate underlying issues with the model's training dynamics, potentially explaining its poor generalization to the external test data. Similarly, the B0(-2) model exhibits a slightly different trend, with the validation loss stalling around 0.01, as shown in Figure 10. This results in a sustained gap between the training and validation loss, a pattern not observed in the other models. These deviations highlight the importance of monitoring training dynamics to identify potential model convergence and generalization issues.

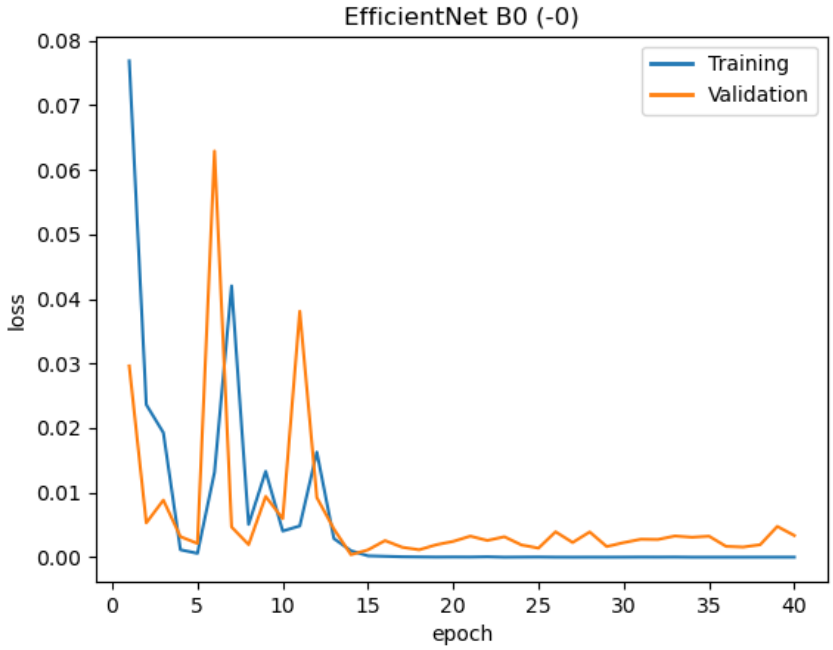


**Figure 6.** Training and validation loss over 40 epochs for the B0(-0) model. The loss stabilizes around epoch 15 after some early training volatility, indicating effective learning.

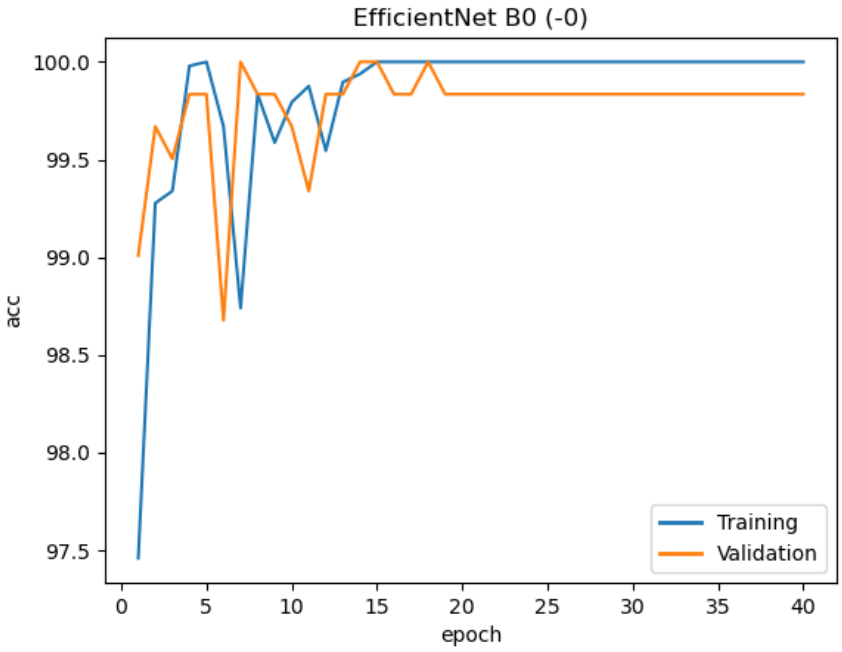


**Figure 7.** Training and validation accuracy over 40 epochs for the B0(-0) model.

 doi:10.20944/preprints202605.0663.v1



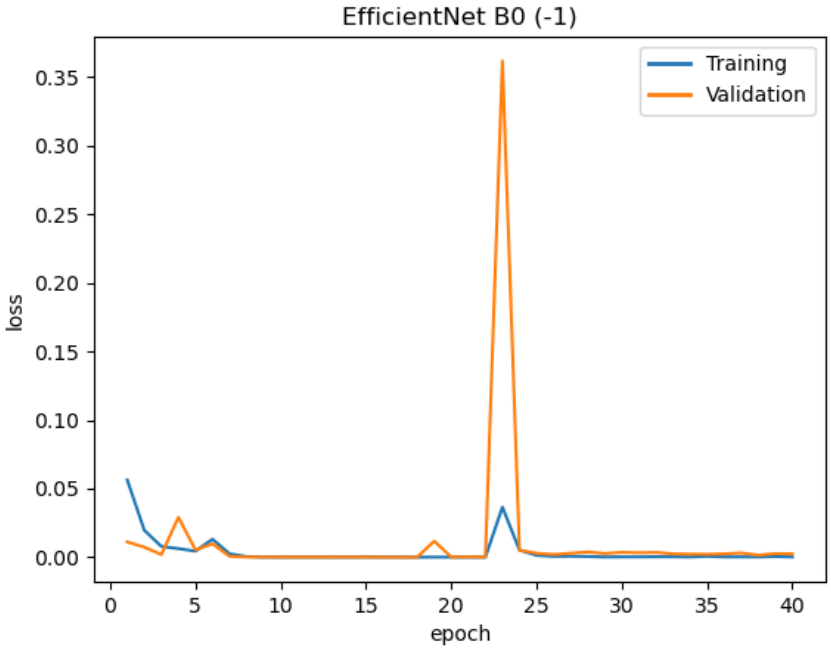


**Figure 8.** Training and validation loss for the B0(-1) model. The large spike at epoch 24 may indicate a larger problem with the model that may explain the model's poor performance on the external test set.

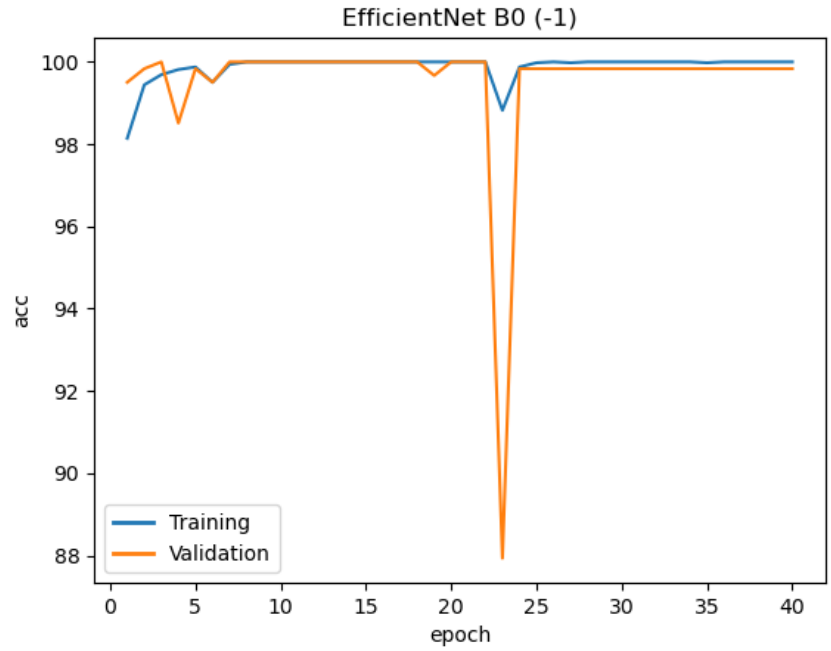


**Figure 9.** Training and validation accuracy for the B0(-1) model.

 doi:10.20944/preprints202605.0663.v1



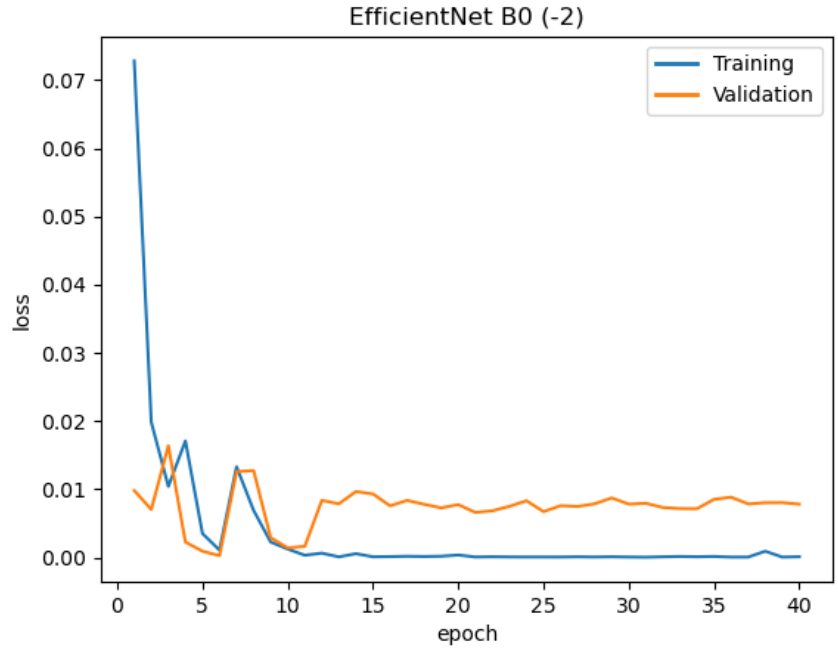


**Figure 10.** Training and validation loss and accuracy for the B0(-2) model. Loss stabilization suggests effective convergence, but with a noticeable gap with training loss not seen in the other models.

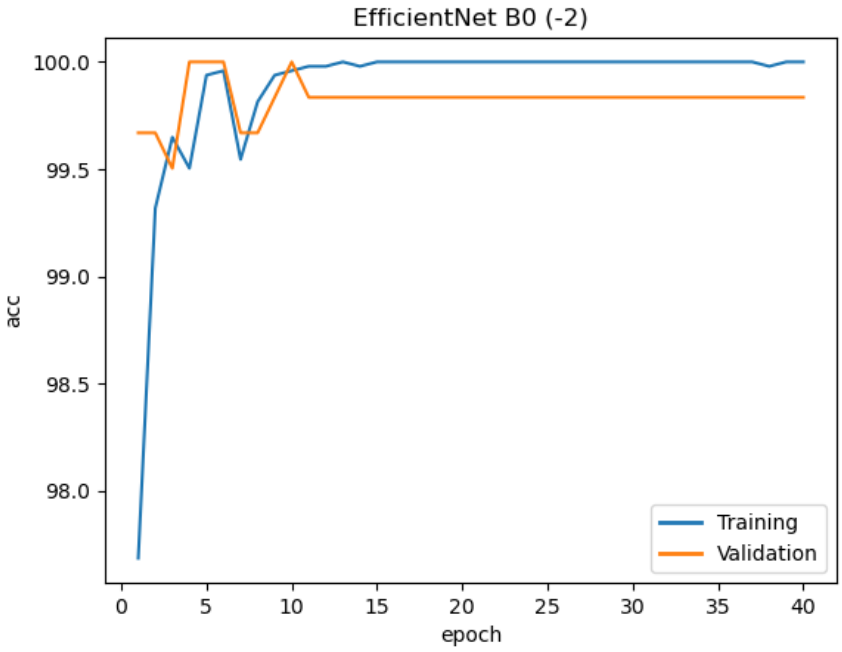


**Figure 11.** Training and validation accuracy for the B0(-2) model.

 doi:10.20944/preprints202605.0663.v1



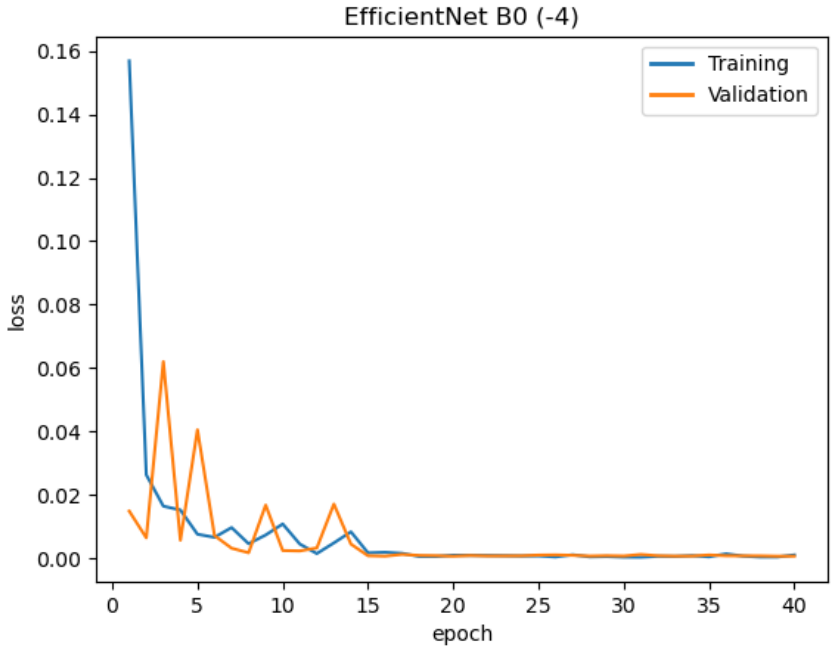


**Figure 12.** Training and validation loss for the B0(-4) model.

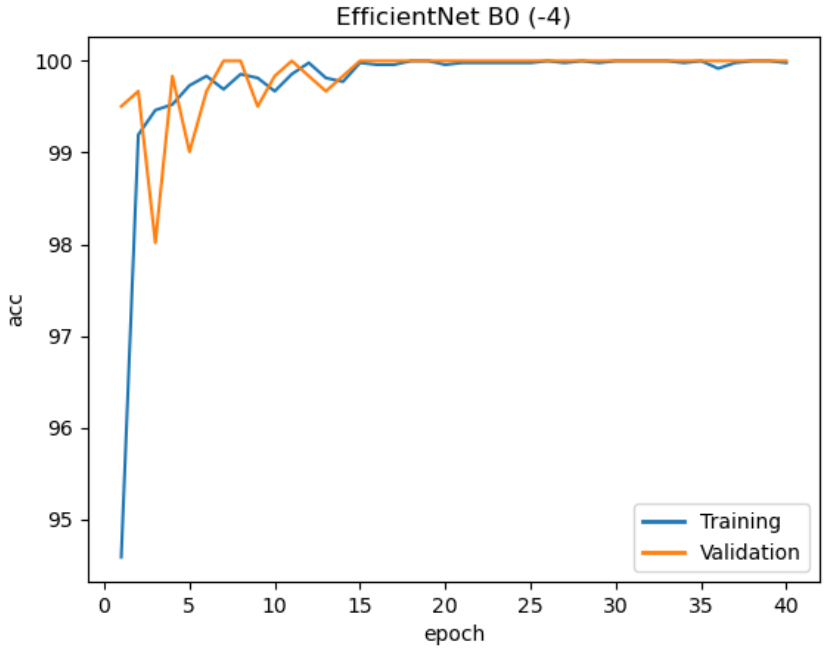


**Figure 13.** Training and validation accuracy for the B0(-4) model.

## 4. Discussion

The performance of the truncated EfficientNet-B0(-3) is highly promising, achieving state-of-the-art accuracy while being 13 times more efficient than the base B0 model and approximately 63 times more efficient than the top-performing model on the same dataset in the literature, the DenseNet-201 [11]. However, it is important to note that Rahman et al. employed a U-Net lung segmentation step before classification, which may improve generalization [11]. Since their model was not tested on external data, a direct comparison of generalization performance is not possible. However, B0 (-3)

 doi:10.20944/preprints202605.0663.v1



shows that significant reductions in model size can be achieved without sacrificing accuracy, at least on the datasets used in this study.

The external sensitivity and specificity of the B0(-3) model—98.71% (95% CI: 98.23%–99.16%) and 98.97% (95% CI: 98.53%–99.32%), respectively—far exceed the World Health Organization's guidelines of 90% sensitivity and 70% specificity. While these results are encouraging, they must be interpreted with caution due to several limitations of this study.

### *4.1. Limitations and Future Directions*

The most significant limitation of this study is that TB classification is only compared to healthy lungs. While the near-perfect internal and external sensitivities and specificities are impressive, they would likely decrease significantly in a more realistic setting with additional classes, such as pneumonia, COVID-19, or the 15 classes in the CheXpert dataset. A logical next step would be to create a composite dataset that includes a broader range of pathologies, better reflecting the challenges faced by CAD systems in real-world applications.

Another limitation is the lack of explicit mitigation techniques for the Clever Hans effect, where models may rely on spurious correlations rather than clinically relevant features. While external validation on the Mendeley dataset provides a baseline check for generalization, it does not fully rule out the possibility of shortcut learning. Future work could incorporate techniques such as lung segmentation using U-Net, as demonstrated by Rahman et al. [11], or text scrubbing using YOLOv3 [27] to reduce the risk of spurious correlations.

Despite these limitations, the results of this study are highly encouraging. The B0(-3) demonstrates that state-of-the-art performance can be achieved with dramatic reductions in model size, further supporting the hypothesis that ImageNet-pretrained models, while effective, may be unnecessarily large for CXR analysis. This finding aligns with recent work by Ke et al., who showed that ImageNet architectures can be made significantly more parameter-efficient without a statistically significant drop in performance [22]. The B0(-3) success suggests that smaller, more efficient models are viable and may be preferable for CAD applications, particularly in resource-constrained settings.

### *4.2. Conclusion*

In summary, this study demonstrates that truncating EfficientNet-B0 can achieve state-of-the-art performance for TB detection while significantly reducing model size. However, the study's limitations, particularly the lack of class diversity and Clever Hans mitigation techniques, highlight the need for further research. Future work should focus on evaluating truncated models on more diverse datasets and incorporating techniques to reduce the risk of shortcut learning. These steps will be critical for developing accurate and robust CAD systems in real-world clinical settings.

**Funding:** This research received no external funding.

**Data Availability Statement:** The datasets analyzed in this study are publicly available from previously published sources. The primary training, validation, and internal testing dataset was obtained from the Kaggle tuberculosis chest X-ray dataset compiled by Rahman et al. [18], which aggregates data from the NIAID TB Portals Program [31], the Belarus Tuberculosis Portal [32], the RSNA Pneumonia Detection Challenge dataset [33], and the dataset published by Jaeger et al. [34]. The external evaluation dataset was constructed using publicly available Mendeley Data repositories from Kiran and Jabeen [28] and Kumar [35]. All datasets used in this study are publicly accessible through the URLs and DOIs provided in the References section. The code used for data preprocessing, model training, evaluation, and figure generation is publicly available at: https://github.com/noahba65/cxr_thesis.

## References

1. World Health Organization. Tuberculosis, 2025. Accessed: 2025-03-14.
2. Reuters. Tuberculosis returns as top infectious disease killer, WHO says. *Reuters* **2024**. Accessed: 2025-03-14.
3. Organization, W.H. Financing for TB Prevention, Diagnostic and Treatment Services. https://www.who.int/teams/global-tuberculosis-programme/tb-reports/global-tuberculosis-report-2024

 doi:10.20944/preprints202605.0663.v1



/tb-financing/4-1-financing-for-tb-prevention--diagnostic-and-treatment-services, 2024. Accessed: 2025-03-14.
4. Rutgers, The State University of New Jersey. History of Tuberculosis, 2024. Accessed: 2025-03-14.
5. Szkwarko, D.; Bouton, T.C.; Rybak, N.R.; Carter, E.J.; Chiang, S.S. Tuberculosis: an epidemic perpetuated by health inequalities. *Rhode Island medical journal (2013)* **2019**, *102*, 47.
6. World Health Organization. Tuberculosis: Questions and Answers, 2025. Accessed: 2025-03-14.
7. País, E. La tuberculosis vuelve a ser la enfermedad infecciosa que más muertes causa. *El País* **2024**. Accessed: 2025-03-14.
8. World Health Organization. *Chest radiography in tuberculosis detection: Summary of current WHO recommendations and guidance on programmatic approaches*; World Health Organization: Geneva, Switzerland, 2016; p. 7. WHO Reference Number: WHO/HTM/TB/2016.20.
9. Organization, W.H.; et al. High priority target product profiles for new tuberculosis diagnostics: report of a consensus meeting, 28-29 April 2014, Geneva, Switzerland. In Proceedings of the High priority target product profiles for new tuberculosis diagnostics: report of a consensus meeting, 28-29 April 2014, Geneva, Switzerland, 2014.
10. Bosman, S.; Ayakaka, I.; Muhairwe, J.; Kamele, M.; van Heerden, A.; Madonsela, T.; Labhardt, N.D.; Sommer, G.; Bremerich, J.; Zoller, T.; et al. Evaluation of C-Reactive Protein and Computer-Aided Analysis of Chest X-rays as Tuberculosis Triage Tests at Health Facilities in Lesotho and South Africa. *Clinical Infectious Diseases* **2024**, *79*, 1293–1302.
11. Rahman, T.; Khandakar, A.; Kadir, M.A.; Islam, K.R.; Islam, K.F.; Mazhar, R.; Hamid, T.; Islam, M.T.; Kashem, S.; Mahbub, Z.B.; et al. Reliable Tuberculosis Detection Using Chest X-Ray With Deep Learning, Segmentation and Visualization. *IEEE Access* **2020**, *8*, 191586–191601. https://doi.org/10.1109/ACCESS.2020.3031384.
12. Huang, G.; Liu, Z.; Weinberger, K.Q. Densely Connected Convolutional Networks. *CoRR* **2016**, *abs/1608.06993*, [1608.06993].
13. Alshmrani, G.M.M.; Ni, Q.; Jiang, R.; Pervaiz, H.; Elshennawy, N.M. A deep learning architecture for multi-class lung diseases classification using chest X-ray (CXR) images. *Alexandria Engineering Journal* **2023**, *64*, 923–935.
14. Ahmed, M.S.; Rahman, A.; AlGhamdi, F.; AlDakheel, S.; Hakami, H.; AlJumah, A.; AlIbrahim, Z.; Youldash, M.; Alam Khan, M.A.; Basheer Ahmed, M.I. Joint diagnosis of pneumonia, COVID-19, and tuberculosis from chest X-ray images: A deep learning approach. *Diagnostics* **2023**, *13*, 2562.
15. Venkataramana, L.; Prasad, D.V.V.; Saraswathi, S.; Mithumary, C.; Karthikeyan, R.; Monika, N. Classification of COVID-19 from tuberculosis and pneumonia using deep learning techniques. *Medical & Biological Engineering & Computing* **2022**, *60*, 2681–2691.
16. Goswami, K.K.; Kumar, R.; Kumar, R.; Reddy, A.J.; Goswami, S.K. Deep learning classification of tuberculosis chest X-rays. *Cureus* **2023**, *15*.
17. Kaur, G.; Sharma, N.; Chauhan, R.; Thapliyal, S.; Gupta, R. Tuberculosis Classification Using EfficientNet B3 Deep Learning Architecture. In Proceedings of the 2023 Global Conference on Information Technologies and Communications (GCITC). IEEE, 2023, pp. 1–6.
18. Rahman, T.; Khandakar, A.; Kadir, M.A.; Islam, K.R.; Islam, K.F.; Mahbub, Z.B.; Ayari, M.A.; Chowdhury, M.E.H. Reliable Tuberculosis Detection using Chest X-ray with Deep Learning, Segmentation and Visualization. *IEEE Access* **2020**, *8*, 191586–191601. https://doi.org/10.1109/ACCESS.2020.3031384.
19. Tan, M.; Le, Q.V. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. *CoRR* **2019**, *abs/1905.11946*, [1905.11946].
20. Bhosale, R.D.; Yadav, D.M. Analysis of EfficientNet Family Models by Retraining for Tuberculosis Detection from Chest X-Ray Images. In Proceedings of the 2023 Second International Conference on Informatics (ICI), 2023, pp. 1–6. https://doi.org/10.1109/ICI60088.2023.10420853.
21. Montalbo, F.J. Truncating fined-tuned vision-based models to lightweight deployable diagnostic tools for SARS-CoV-2 infected chest X-rays and CT-scans. *Multimedia Tools and Applications* **2022**, *81*, 16411–16439.
22. Ke, A.; Ellsworth, W.; Banerjee, O.; Ng, A.Y.; Rajpurkar, P. CheXtransfer: performance and parameter efficiency of ImageNet models for chest X-Ray interpretation. In Proceedings of the Proceedings of the conference on health, inference, and learning, 2021, pp. 116–124.
23. Tan, M.; Chen, B.; Pang, R.; Vasudevan, V.; Sandler, M.; Howard, A.; Le, Q.V. Mnasnet: Platform-aware neural architecture search for mobile. In Proceedings of the Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 2820–2828.

 doi:10.20944/preprints202605.0663.v1




24. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition, 2015, [arXiv:cs.CV/1512.03385].
25. Vásquez-Venegas, C.; Wu, C.; Sundar, S.; Prôa, R.; Beloy, F.J.; Medina, J.R.; McNichol, M.; Parvataneni, K.; Kurtzman, N.; Mirshawka, F.; et al. Detecting and Mitigating the Clever Hans Effect in Medical Imaging: A Scoping Review. *Journal of Imaging Informatics in Medicine* **2024**, pp. 1–17.
26. DeGrave, A.J.; Janizek, J.D.; Lee, S.I. AI for radiographic COVID-19 detection selects shortcuts over signal. *Nature Machine Intelligence* **2021**, *3*, 610–619.
27. Pedrosa, J.; Aresta, G.; Ferreira, C.A.; Mendonça, A.M.; Campilho, A. Automatic Label Detection in Chest Radiography Images. In Proceedings of the BIOIMAGING, 2022, pp. 63–69.
28. Kiran, S.; Jabeen, I. Dataset of Tuberculosis Chest X-rays Images, 2024. https://doi.org/10.17632/8j2g3csprk.2.
29. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 4510–4520.
30. Tran, H.N.; Nguyen, N.V.; Le, N.Q.; Nguyen, N.N.; Le, T.A.; Nguyen, V.D. Enhancing semantic scene segmentation for indoor autonomous systems using advanced attention-supported improved UNet. *Signal, Image and Video Processing* **2025**, *19*, 190.
31. of Allergy, N.I.; (NIAID), I.D. NIAID TB portal program dataset. Online.
32. for Health, N.I.; Excellence, C. BELARUS TUBERCULOSIS PORTAL. Online, 2020. Accessed: 2024-09-28.
33. Kaggle. RSNA Pneumonia Detection Challenge. Online, 2020. Accessed: 2024-09-28.
34. Jaeger, S.; Candemir, S.; Antani, S.; Wáng, Y.X.J.; Lu, P.X.; Thoma, G. Two public chest X-ray datasets for computer-aided screening of pulmonary diseases. *Quantitative Imaging in Medicine and Surgery* **2014**, *4*, 475.
35. Kumar, S. Covid19-Pneumonia-Normal Chest X-Ray Images, 2022. https://doi.org/10.17632/dvntn9yhd2.1.