

A LIGHTWEIGHT, RAPID AND EFFICIENT DEEP CONVOLUTIONAL NETWORK FOR CHEST X-RAY TUBERCULOSIS DETECTION

Daniel Capellán-Martín^{1,2}, Juan J. Gómez-Valverde^{1,2}, David Bermejo-Peláez^{1,2}, María J. Ledesma-Carbayo^{1,2}

¹ Biomedical Image Technologies, ETSI Telecomunicación, Universidad Politécnica de Madrid

² CIBER-BBN, Instituto Salud Carlos III, Madrid, Spain

ABSTRACT

Tuberculosis (TB) is still recognized as one of the leading causes of death worldwide. Recent advances in deep learning (DL) have shown to enhance radiologists' ability to interpret chest X-ray (CXR) images accurately and with fewer errors, leading to a better diagnosis of this disease. However, little work has been done to develop models capable of diagnosing TB that offer good performance while being efficient, fast and computationally inexpensive. In this work, we propose LightTBNNet, a novel lightweight, fast and efficient deep convolutional network specially customized to detect TB from CXR images. Using a total of 800 frontal CXR images from two publicly available datasets, our solution yielded an accuracy, F1 and area under the ROC curve (AUC) of 0.906, 0.907 and 0.961, respectively, on an independent test subset. The proposed model demonstrates outstanding performance while delivering a rapid prediction, with minimal computational and memory requirements, making it highly suitable for deployment in handheld devices that can be used in low-resource areas with high TB prevalence. Code publicly available at: <https://github.com/dani-capellan/LightTBNNet>.

Index Terms— Deep learning, Efficient ML, Deep convolutional neural network, Chest X-ray, Tuberculosis.

1. INTRODUCTION

Tuberculosis (TB) remains one of the leading causes of death worldwide [1]. About a quarter of the world's population is infected with *Mycobacterium tuberculosis*, the agent that causes TB, usually affecting the lungs, although it can also affect other parts of the body [2]. Prior to the coronavirus (COVID-19) outbreak, TB was the leading infectious disease-related cause of death due to a single infectious agent, surpassing HIV/AIDS [1, 3]. Since then, there has been a significant drop in the number of people newly diagnosed and treated that has led to an increase in TB deaths [1]. Early diagnosis of TB is essential to promote effective treatment and to reduce further transmission. Chest X-rays (CXRs) are recommended by the World Health Organization (WHO) as a TB screening and triage tool given their wide availability, rapid

execution, and acquisition with portable machines to reduce the risk of cross infection [4, 5]. Artificial intelligence (AI)-powered computer-aided detection (CAD) tools have shown to improve health outcomes, especially in under-resourced areas with a high TB burden [6, 7]. By using these solutions, radiologists are able to interpret images with greater accuracy and with fewer errors, allowing them to devote more time to patient care [8].

Several studies have explored the use of deep learning (DL) for TB detection in CXRs. Hwang et al. [9] and Islam et al. [10] proposed DL-based methods combining Convolutional Neural Networks (CNNs) and Transfer Learning techniques, and also explored the ensembling of predictions from multiple architectures to improve TB detection. However, efficiency is crucial when designing DL models for medical tasks, given the limited amounts of data that may lead to overfitting and poor generalization. Pasa et al. [11] proposed a novel Residual Neural Network (ResNet) for TB detection, emphasizing the importance of efficiency and the need for models that can generalize well with limited data. Rajpurkar et al. [12] combined clinical information with CXR images in a deep learning algorithm based on a DenseNet-121 architecture to diagnose TB. Wong et al. [13] introduced TBNet, a novel architecture specifically tailored to TB detection in CXRs, which included self-attention mechanisms and explored new approaches to CNN-based TB detection. Despite previous efforts, considering the WHO recommendation for TB screening and the increase in undiagnosed and untreated TB, there is an urgent need to create new solutions specifically tailored for deployment in low-resource settings, where efficiency and computational time could be a challenge.

In this context, we propose LightTBNNet, a novel lightweight, fast and efficient CNN specially customized to detect TB from CXRs. The main objectives of our design have been to outperform current state-of-the-art methods while reducing computational and memory requirements, through a careful design of the number of residual blocks and skip connections, and by proposing a specific architecture that reduces the number of operations and thus increase efficiency. This could allow our method to be deployed on handheld devices maximizing accuracy. Additionally, we present a thorough

assessment of the performance and efficiency of our method considering public datasets and state-of-the-art alternatives.

2. METHODS

2.1. Dataset and splits

For the development of this study, we considered a total of 800 frontal CXR images from two public datasets [14]:

- Montgomery County X-ray Set (MC): 138 frontal CXRs. 80 CXRs are non-TB and 58 are abnormal with manifestations of TB, including effusions and milary patterns.
- Shenzhen Hospital X-ray Set (SZ): 662 frontal CXRs. 326 CXRs are non-TB and 336 are abnormal showing diverse manifestations of TB, including some pediatric CXRs.

The mean age of the whole set (MC & SZ) is of 36.24 ± 15.65 y.o. (MC: 40.11 ± 18.72 y.o., SZ: 35.43 ± 14.80 y.o.). The dataset was randomly split following a training-validation-testing scheme. Twenty percent of the data was reserved for independent testing, with no patient overlap, and stratifying the data by cohort, TB class (positive/negative), sex and age, ensuring that the data were as balanced as possible. The remaining 80% was further split using a 5-fold cross-validation approach for hyperparameter tuning.

2.2. Image preprocessing and data augmentation

We applied Contrast Limited Adaptive Histogram Equalization (CLAHE) to all the data, as it has been shown to improve the detection of TB in CXRs [15, 16, 17]. The images were downscaled to 256×256 when inputted to the networks, ensuring that the tensors are not excessively large while keeping the spatial resolution required to distinguish TB findings, if present.

The training data were augmented with a random horizontal flip, a random rotation from -15° to $+15^\circ$, a random shift from -10% to $+10\%$ (both height and weight) and a random scaling from -10% to $+10\%$. The input images were normalized to zero mean and unit variance in order to train the network more efficiently.

2.3. Model architecture

The proposed architecture (see Figure 1) is an efficient, lightweight adaptation of the ResNet architecture composed of N residual convolutional blocks. These residual blocks consist of two 3×3 convolutions followed by ReLU activations and batch normalizations (BatchNorm). Parallel to these two convolutions, there are skip connections with a 1×1 convolution, which act as shortcuts when the tensors of both branches are concatenated. The residual block ends with a max-pooling layer (pooling size 2×2 , stride 2). These residual blocks are followed by a 1×1 convolutional layer, which reduces the dimensionality of the feature maps extracted

from the previous blocks. Then, an MLP-based classification module composed of two fully-connected layers is included.

There is a trade-off between inference time and model size that depends on the number of residual blocks (N). The more residual blocks we include to the architecture, the lighter the model will be, since the extracted feature maps will be smaller. However, this leads to an increase in the number of multiply-accumulate operations (MACs) and inference time. Moreover, the use of skip connections in the residual blocks allows the model to automatically decide where to pass the information through, following an optimal pathway for the given task. The number of convolutional layers impacts on the detection of TB findings in CXR images, since the more convolutional layers the model has, the more general features it will extract from the image.

The design of our architecture differs from traditional ResNets in the introduction of pooling layers between convolutions and in the use of convolutional steps within the skip connections. In addition, our proposal includes a fewer number of convolutional layers and a smaller number of parameters. The use of fewer convolutional layers, ReLU activation functions and max-pooling layers (which reduce tensor dimensionality) within the architecture leads to a fewer number of MACs and parameters, thus making it more efficient.

Other high-performing and efficient architectures including DenseNet-121 (used by Rajpurkar et al. [12]), EfficientNet (B0 and V2-s), MobileNet (MobileNetV3-small) and ResNet (ResNet-18, ResNet-34), were used in the experiments to compare with the performance and efficiency of our proposal.

3. EXPERIMENTS AND RESULTS

3.1. Implementation details

For all the experiments, we used Adam optimizer with default parameters ($\beta_1 = 0.9, \beta_2 = 0.999$), batch size of 16 images, learning rate $1 \times e^{-4}$, and focal loss [18], a variation of the cross entropy loss which introduces a modulating term $(1-p_t)^\gamma$ that downweights the loss assigned to well-classified examples (easy examples) and upweights the loss assigned to hard-to-classify examples, focusing learning on negative hard examples. The focal loss is formulated as follows:

$$FL(p_t) = -(1-p_t)^\gamma \log(p_t) \quad (1)$$

where $p \in [0, 1]$ is the model's estimated probability for target label t and $\gamma \geq 0$ is a tunable focusing parameter.

We also performed 5-fold cross-validation. On each fold, we evaluated performance on the validation subset and chose the best model checkpoint based on the highest validation AUC.

During the testing of the models, we used the best algorithm trained on each fold and averaged their predictions (TB

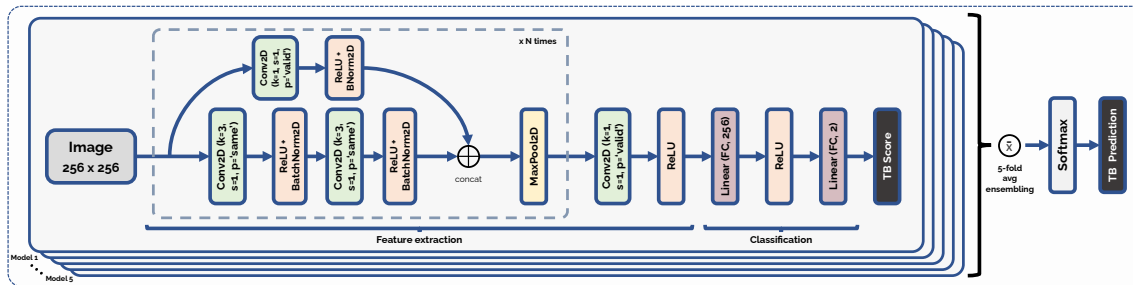


Fig. 1: Model architecture (LightTBNet) and pipeline for TB prediction. N corresponds to the hyperparameter that controls the number of residual blocks included in the model. Inference is done by ensembling five models obtained from five different cross-validation folds.

Model	Val ACC	Val F1	Val AUC	MACs (G)	Params (M)	Inference time (ms)	Size (MB)
LightTBNet (N=3)	0.869 ± 0.024	0.874 ± 0.027	0.948 ± 0.016	0.822	4.298	2.25 ± 0.63	150.02
LightTBNet (N=4)	0.867 ± 0.057	0.873 ± 0.043	0.952 ± 0.02	1.138	1.467	2.55 ± 0.21	148.44
LightTBNet (N=5)	0.836 ± 0.04	0.878 ± 0.019	0.947 ± 0.013	1.456	1.932	3.22 ± 0.27	154.9

Table 1: Results for model comparison in validation set (cross-validation). The inference time is expressed in milliseconds (ms) and the values correspond to the time taken to predict one image by a single fold model. The model size is expressed in megabytes (MB) and corresponds to the size of a single fold model (including architecture and weights). The best option in each metric is highlighted in bold.

scores) across an ensemble of the five models (see Figure 1). Previous experiments showed better performance when ensembling the five models extracted from each fold, ensuring a more robust final score.

To implement the networks, we used the PyTorch framework (v. 1.12). For the experiments, we used a workstation with NVIDIA A30 24GB GPU, 256GB RAM and 2 x Intel Xeon Silver 4216 @ 2.1 GHz CPUs.

3.2. Hyperparameter optimization

We compared the TB classification performance, efficiency and size among three different versions of the proposed architecture, in which we vary the the number of residual convolutional blocks (N). We considered different architectures ($N=3,4,5$) and selected the best performing configuration on the cross-validation subsets by analyzing their AUC scores. We compared these three versions since, if we took a smaller number of N , the network was not deep enough to correctly find TB-compatible findings, and if we took a higher value, the feature maps extracted after the convolutional steps were extremely small considering the input image size, and there was a significant loss of spatial information in the process.

We also calculated the multiply-accumulate operations (MACs)¹, the number of parameters and inference time for each of the configurations in order to compare their efficiency and computational requirements. These calculations were performed by running 300 repetitions in a row and then averaging the timings, thus preventing the GPU from going into power-saving mode when measuring time.

The results obtained for each of the configurations in the validation set are included in Table 1. All three versions

demonstrated high efficiency and low inference times. LightTBNet with 4 residual convolutional blocks ($N = 4$) shows better performance (val AUC) with low computational requirements when comparing with the other configurations ($N = 3, 5$) tested. As a result, this configuration was selected for testing.

3.3. Results on test set

In this section, we test our best configuration ($N=4$) along with other high-performing and efficient architectures, detailed in section 2.3, on an independent test subset of 160 images. The Table 2 shows that our proposal outperforms the results of other efficient approaches both in terms of performance (highest AUC score) and efficiency (lowest number of parameters and inference time). The inference times of all the architectures were obtained following the same procedure as detailed in section 3.2. Moreover, a graphical comparison taking into account performance (AUC score) and efficiency (MACs & number of parameters) of each of the architectures, is shown in Figure 2.

Next, we compared as fairly as possible our proposal with other approaches in the literature that report results on the same TB datasets. To do so, we considered different deep-learning proposals and selected the best results from each contribution which are comparable to our results. As shown in Table 3, our proposal outperforms the results of other contributions, both when combining and splitting separately the two TB datasets. However, our work could not be directly compared to other efficient TB methodologies, such as those proposed by Wong et al. (TB-Net) [13] and Lakhani et al. [16], as their results were not reported independently for the two public datasets used in this work. When testing on the

¹Code used: <https://github.com/Lyken17/pytorch-OpCounter>

Model	Test ACC	Test F1	Test AUC	MACs (G)	Params (M)	Inference time (ms)	Size (MB)
DenseNet-121	0.906	0.908	0.952	3.639	6.95	26.09 ± 0.9	411.01
EfficientNetB0	0.844	0.839	0.937	0.513	4.01	13.23 ± 0.33	213.51
EfficientNetV2-s	0.9	0.893	0.955	3.749	20.18	27.7 ± 0.69	444.82
MobileNetV3-small	0.812	0.803	0.892	0.072	1.52	6.59 ± 0.43	46.93
ResNet-18	0.875	0.868	0.959	2.274	11.176	3.79 ± 0.76	118.64
ResNet-34	0.9	0.896	0.951	4.695	21.288	7.56 ± 0.32	207.21
LightTBNNet (N=4)	0.906	0.907	0.961	1.138	1.467	2.55 ± 0.21	148.44

Table 2: Results for model comparison in test set. The inference time is expressed in milliseconds (ms) and the values correspond to the time taken to predict one image by a single fold model. The model size is expressed in megabytes (MB) and corresponds to the size of a single fold model (including architecture and weights). The best option in each metric is highlighted in bold.

Model / Implementation	ACC (MC + SZ)	AUC (MC + SZ)	ACC (MC)	AUC (MC)	ACC (SZ)	AUC (SZ)
Hwang et al. [9]	-	-	0.67	0.88	0.84	0.93
Islam et al. [10]	-	-	-	-	0.88	0.91
Pasa et al. [11]	0.862	0.925	0.79	0.811	0.844	0.9
LightTBNNet (N=4)	0.906	0.961	0.889	0.943	0.91	0.963

Table 3: Accuracy (ACC) and AUC results of our model on the Montgomery County (MC) and Shenzhen (SZ) test subsets in comparison to the results reported in other publications. To provide a better comparison with our results, figures from other publications where pre-training is performed on ImageNet and/or ensembles are used have been discarded. Only deep learning-based approaches with results on MC & SZ were considered. The best option in each metric is highlighted in bold.

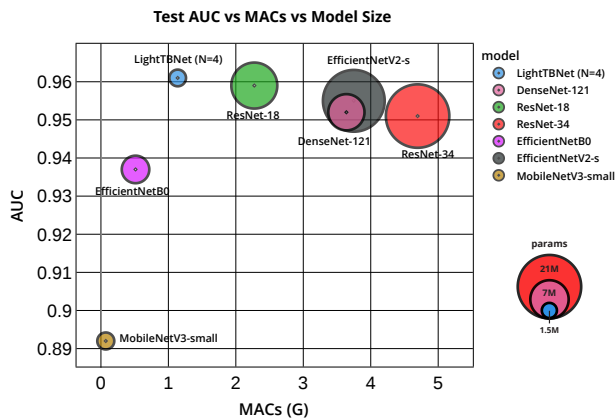


Fig. 2: Performance (AUC score) on the test set vs. efficiency (MACs & number of parameters). Comparison between our proposal and other high-performing and efficient architectures.

independent test set, the proposed model provides an ACC, F1 and AUC of 0.906, 0.907 and 0.961, respectively. With a default prediction threshold of 0.5, we achieve a sensitivity (SN) and specificity (SP) of 0.924 and 0.889, respectively, thus fulfilling the requirements of the WHO’s Target Product Profile (TPP) of triage tests ($\geq 90\%$ SN and $\geq 70\%$ SP). Finally, Figure 3 displays the saliency maps and grad-CAMs of two true positive (TP) cases, i.e., truly classified as positive for TB. Saliency maps can be used to assess an image’s overall spatial support for a given class. On the other hand, grad-CAMs are coarse location maps that highlight the key areas of the image that the model focuses on for prediction. As shown in Figure 3, the proposed model correctly identifies a TB-compatible cavity in the left lung of patient 1 and shows a generalized activation in the image of patient 2 due to the miliary pattern present in the CXR.

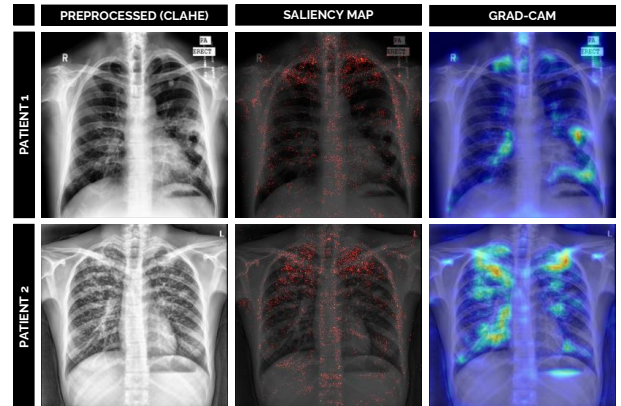


Fig. 3: Preprocessed image (CLAHE), saliency map and grad-CAM of two TB positive cases truly predicted as positive. Patient 1 belongs to MC and patient 2 to SZ. The output TB scores provided by the model were 0.8215 and 1.0 for patient 1 and 2, respectively.

4. CONCLUSION

In this study, we have proposed a lightweight, fast and efficient DL-based residual network for detecting TB from CXRs, which has shown outstanding performance and low computational requirements compared to other state-of-the-art architectures and implementations. Given these low computational requirements and good performance, this solution demonstrates its potential use in computer-aided diagnosis systems and could be deployed for use in smartphones or other handheld devices, serving to gain accessibility to diagnosis and to reduce the high clinical burden when screening for TB, which remains a challenging task, especially in low-resource settings with high TB prevalence.

5. COMPLIANCE WITH ETHICAL STANDARDS

This research study was conducted using retrospective data from open access sources (details in text). Ethical approval was not required as confirmed by the license of the public databases used.

6. ACKNOWLEDGEMENTS

This work was supported by H2020-MSCA-RISE-2018 INNOVA4TB (EU) project (ID 823854) and ADVANCE-TB Cost Action (CA21164 (EU)). DCM's PhD fellowship was supported by Universidad Polit3cnica de Madrid.

7. REFERENCES

- [1] World Health Organization, "Global Tuberculosis Report," 2022.
- [2] S. Ravimohan, H. Kornfeld, D. Weissman, and G. P. Bisson, "Tuberculosis and lung damage: from epidemiology to pathophysiology," *European Respiratory Review*, vol. 27, no. 147, Mar 2018.
- [3] A. Koch and V. Mizrahi, "Mycobacterium tuberculosis," *Trends in Microbiology*, vol. 26, no. 6, pp. 555–556, Jun 2018.
- [4] World Health Organization, "Chest Radiography in Tuberculosis Detection: Summary of current WHO recommendations and guidance on programmatic approaches," 2016.
- [5] M. A. Orsi, G. Oliva, T. Toluian, et al., "Feasibility, Reproducibility, and Clinical Validity of a Quantitative Chest X-Ray Assessment for COVID-19," *The American Journal of Tropical Medicine and Hygiene*, vol. 103, no. 2, pp. 822, Aug 2020.
- [6] S. Kulkarni and S. Jha, "Artificial Intelligence, Radiology, and Tuberculosis: A Review," *Academic Radiology*, vol. 27, no. 1, pp. 71–75, Jan 2020.
- [7] Z. Z. Qin, T. Naheyan, M. Ruhwald, et al., "A new resource on artificial intelligence powered computer automated detection software products for tuberculosis programmes and implementers," *Tuberculosis*, vol. 127, March 2021.
- [8] M. P. McBee, O. A. Awan, A. T. Colucci, et al., "Deep Learning in Radiology," *Academic Radiology*, vol. 25, no. 11, pp. 1472–1480, Nov 2018.
- [9] S. Hwang, H.-E. Kim, J. Jeong, and H.-J. Kim, "A novel approach for tuberculosis screening based on deep convolutional neural networks," in *Medical imaging 2016: computer-aided diagnosis*. SPIE, 2016, vol. 9785, pp. 750–757.
- [10] M. T. Islam, M. A. Aowal, A. T. Minhaz, and K. Ashraf, "Abnormality Detection and Localization in Chest X-Rays using Deep Convolutional Neural Networks," *arXiv*, May 2017.
- [11] F. Pasa, V. Golkov, F. Pfeiffer, et al., "Efficient Deep Network Architectures for Fast Chest X-Ray Tuberculosis Screening and Visualization," *Scientific Reports* 2019 9:1, vol. 9, no. 1, pp. 1–9, Apr 2019.
- [12] P. Rajpurkar, C. O'Connell, A. Schechter, et al., "CheX-aid: deep learning assistance for physician diagnosis of tuberculosis using chest x-rays in patients with HIV," *npj Digital Medicine* 2020 3:1, vol. 3, no. 1, pp. 1–8, Sep 2020.
- [13] A. Wong, J. R. H. Lee, H. Rahmat-Khah, et al., "TB-Net: A Tailored, Self-Attention Deep Convolutional Neural Network Design for Detection of Tuberculosis Cases From Chest X-Ray Images," *Frontiers in Artificial Intelligence*, vol. 5, pp. 827299, Apr 2022.
- [14] S. Jaeger, S. Candemir, S. Antani, et al., "Two public chest X-ray datasets for computer-aided screening of pulmonary diseases," *Quantitative Imaging in Medicine and Surgery*, vol. 4, no. 6, pp. 475, Dec 2014.
- [15] K. Zuiderveld, "Contrast Limited Adaptive Histogram Equalization," *Graphics Gems*, pp. 474–485, Jan 1994.
- [16] P. Lakhani and B. Sundaram, "Deep learning at chest radiography: Automated classification of pulmonary tuberculosis by using convolutional neural networks," *Radiology*, vol. 284, no. 2, pp. 574–582, 2017.
- [17] G. Siracusano, A. L. Corte, M. Gaeta, et al., "Pipeline for Advanced Contrast Enhancement (PACE) of Chest X-ray in Evaluating COVID-19 Patients by Combining Bidimensional Empirical Mode Decomposition and Contrast Limited Adaptive Histogram Equalization (CLAHE)," *Sustainability (Switzerland)*, vol. 12, no. 20, pp. 1–18, Oct 2020.
- [18] T. Y. Lin, P. Goyal, R. Girshick, et al., "Focal Loss for Dense Object Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 2, pp. 318–327, Aug 2020.