UNDER REVIEW

RESCIENCE C

Replication / Machine Learning

[Re] TB-Net: A Tailored, Self-Attention Deep Convolutional Neural Network Design for Detection of Tuberculosis Cases from Chest X-Ray Images

Christopher Huang[1], **Wenhao Lu**[2], **Adithya Balakumar**[2], **Jonathan Liu**[2] and **Hamzah Rizvi**[3]

[1]Heritage High School, United States – [2]Millburn High School, Millburn, NJ, United States – [3]West Lafayette Jr./Sr. High School, West Lafayette, IN, United States

Edited by
(Editor)

Received
–

Published
–

DOI
–

Abstract We attempt to reproduce TB-Net (Wong et al., 2022), a self-attention convolutional network reported to reach 99.86% accuracy, 100% sensitivity and 99.7% specificity for tuberculosis detection from chest radiographs, and extend it with a compression and robustness study aimed at smartphone deployment in low-resource clinics. Two obstacles make a faithful replication impossible from published materials: the released TF1 code loads its architecture from a checkpoint whose download link is dead, and the public release of the training cohort has changed since publication (700 TB images today against roughly 3,500 referenced by the original split files). We therefore evaluate a capacity-matched 4.2M-parameter reconstruction and a 0.27M-parameter compact re-implementation on leak-checked splits of the current Rahman cohort, across five seeds with bootstrap confidence intervals. The compact model reaches 98.5±1.8% accuracy and 96.6±1.8% sensitivity, short of the original headline numbers but consistent with a near-saturated task; the larger reconstruction performs worse under the same budget. Compression is essentially free at moderate strength: 25% unstructured pruning, FP16 and dynamic INT8 preserve accuracy, iterative pruning removes the catastrophic sensitivity collapse that one-shot 75% pruning produces, and a distilled MobileNetV3-Small student shows no advantage over an identically trained from-scratch control. Simulated phone-capture distortions cut accuracy roughly in half, and external validation on the Montgomery and Shenzhen cohorts scores at or below chance for reasons we could not resolve; we report this openly rather than publish it as a domain-shift finding. We also document every defect found during the study, including a train/test leak in our own early pipeline whose removal changed the headline numbers by less than one point.

A replication of K. Wong 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 5 (2022).

1 Introduction

Tuberculosis remains among the deadliest infectious diseases worldwide, with the highest burden concentrated in regions where radiologists are scarce[1]. Deep learning systems for reading chest radiographs have repeatedly matched or exceeded human readers under controlled conditions[2], which makes an open, lightweight, locally runnable model an attractive proposition for clinics that lack both specialists and reliable connectivity. TB-Net[3] is one such proposal: a self-attention deep convolutional network, built from depthwise-separable convolutions and attention condensers via machine-driven design exploration, reported to reach 99.86% accuracy, 100% sensitivity and 99.7% specificity on a public tuberculosis chest X-ray cohort with a budget of roughly 4.24M parameters. This article reports an attempt to reproduce TB-Net in PyTorch and to extend it with the compression techniques – pruning, quantization and knowledge distillation[4,5,6] – that would be needed to run such a model on a mid-range smartphone, together with a

Correspondence should be addressed to Wenhao Lu (wenhao@nxthorizon.org)
The authors have declared that no competing interests exist.
Code is available at https://github.com/AIscend-Research/tb-repro..
Data is available at https://www.kaggle.com/datasets/tawsifurrahman/tuberculosis-tb-chest-xray-dataset.

simulated study of the phone-captured-film setting that motivates on-device deployment in the first place.
Two findings reshaped the study before any model was trained, and we state them up front because they bound what any replication of this work can claim.

1. **The original architecture is not recoverable from the released materials.** The published TF1 training script loads its computation graph with `tf.train.import_meta_graph`, so the network topology lives inside a checkpoint rather than in source code, and the checkpoint download link is dead. A layer-for-layer port is therefore impossible. What we evaluate as the *full* model is a capacity-matched reconstruction (∼4.32M parameters) built from the motifs the paper describes, not a faithful port, and we make no claim of architectural fidelity for it.
2. **The training cohort has changed since publication.** The original repository's own split files, preserved in our repository's history, reference 6,927 images with tuberculosis indices running to roughly 3,500, while the current public release of the Rahman dataset[7] contains 700 TB images. The cohort the original numbers were computed on no longer exists in the form it was used, so a like-for-like metric comparison is not available.

Within those constraints, we run a five-seed study with bootstrap confidence intervals covering: (i) baseline performance of a compact 0.27M-parameter re-implementation and the 4.32M capacity-matched reconstruction, under both the paper's preprocessing pipeline and a plain grayscale one; (ii) unstructured, structured and iterative pruning[8]; (iii) knowledge distillation into MobileNetV3-Small[9] against a from-scratch control; (iv) FP16 and dynamic INT8 quantization with ONNX CPU latency; (v) simulated phone-capture robustness with a per-distortion ablation; and (vi) external validation on the Montgomery and Shenzhen cohorts[10] that the original work reported on.
In the spirit of the NeurIPS reproducibility program[11] and of this journal, we also publish a complete defect log (`docs/BUGS.md` in the repository), including a train/test leak in our own early pipeline, and we report one result – below-chance external AUC – that we deliberately decline to interpret because we could not explain it.

2 Methods

2.1 Data and leak-checked splits

All models are trained and tested on the current Kaggle release of the Rahman et al. tuberculosis chest X-ray database[7]: 4,200 distinct PNG radiographs, 3,500 normal and 700 tuberculosis, a 5:1 class imbalance. We use a stratified 80/10/10 split (3,360 train, 420 validation, 420 test; the test set holds 350 normal and 70 TB images).
The 5:1 ratio has two consequences that run through the whole paper. First, a model that predicts "normal" for every image already scores 83.3% accuracy, so **sensitivity is the metric of record** throughout, and training selects checkpoints on validation sensitivity. Second, an early version of this study had a data-integrity defect worth recounting: the working directory contained each TB radiograph twice under two naming schemes, the split generator split over rows rather than distinct images, and 115 of the 140 TB images in the old test split were duplicates of training images. Our split generator now maps filenames onto a canonical image identity, deduplicates before splitting, and refuses to write splits in which any image crosses a split boundary. Notably, fixing the leak changed the compact baseline by less than a tenth of a point of accuracy (98.81 vs. 98.57) and left sensitivity unchanged – the contamination was real and had to go, but it was not what produced the high scores. Forensics against the original repository's split files showed the duplicates were introduced locally by our own pipeline, not inherited from the published TB-Net splits.

2.2 Preprocessing as an experimental axis

The original work describes a specific pipeline: blue-channel extraction, a padding-aware auto-crop, resizing, a fixed crop from the data interface, and corner masking to hide burnt-in metadata. We implement this as the *faithful* variant. Because an earlier version of this study had silently used a plain grayscale-and-resize pipeline instead, we kept that as the *simple* variant and promoted the choice to a first-class experimental axis rather than a background assumption.

One ambiguity is unresolvable from the released code: the TF1 reference calls `tf.image.crop_to_bounding_bo` `11, 11, 168, 202)`, whose last two arguments are a target height and width, while the repository's own Python preprocessing slices `[11:168, 11:202]` as if they were end indices. These are different crops. Our cache builder follows the TensorFlow semantics and exposes a flag for the other reading; which the original authors intended cannot be determined.

2.3 Models

Compact (0.27M). A small network built from the two motifs the paper describes: depthwise-separable convolution blocks interleaved with attention condensers (global average pooling into a two-layer bottleneck MLP producing per-channel sigmoid gates). 270,778 parameters, 113M MACs per 224×224 input, 1.07 MB on disk in FP32.

Full (4.32M). A capacity-matched reconstruction sized to the parameter budget reported for TB-Net, using the same motifs across five stages of increasing width. 4,320,598 parameters, 498M MACs, 16.6 MB in FP32. Again: this matches the reported budget and described building blocks, not the (unrecoverable) original topology.

Student (MobileNetV3-Small). For distillation, an ImageNet-initialized MobileNetV3-Small[9] with a two-class head (5.9 MB), trained either with KL-divergence soft targets from the compact teacher plus hard labels, or from scratch on hard labels alone as a control.

2.4 Training and evaluation protocol

Every experiment runs across five seeds (0–4), with all randomness routed through a single seeding function covering Python, NumPy, PyTorch, CUDA and DataLoader workers. Models train for ten epochs with Adam, checkpoint-selected on validation sensitivity. Every metric is reported as mean ± standard deviation across seeds, and each individual run carries a percentile-bootstrap 95% confidence interval. Per-sample predicted probabilities for every model and condition are written to the repository (`results/probs/`), so ROC analysis, operating-point selection and paired comparisons require no re-inference. Paired comparisons are computed on the shared test set with bootstrap CIs on the per-seed difference. The environment was a single Tesla T4 (torch 2.10, CUDA 12.8); the complete run takes about four hours.

3 Results: baseline reproduction

Table 1 and Figures 1 summarize the baselines. The compact model lands within roughly 1.4 points of the reported accuracy and 3.4 points of the reported sensitivity, despite having $16\times$ fewer parameters than the reported budget and being trained on a different (smaller, more imbalanced) release of the cohort. We read this as evidence that the task, as posed by this dataset, is close to saturated and does not require the full parameter budget – but the gap to the published 100% sensitivity is real and survives every configuration we tried.

Arch	Preproc.	Accuracy	Sensitivity	Specificity	AUC	Size (MB)
compact	faithful	98.48 ± 1.83	96.57 ± 1.81	98.86 ± 2.28	0.9966 ± 0.0025	1.07
compact	simple	98.76 ± 0.54	96.29 ± 3.29	99.26 ± 0.56	0.9991 ± 0.0006	1.07
full	faithful	89.60 ± 7.55	91.00 ± 4.15	89.31 ± 9.52	0.9741 ± 0.0129	16.59
full	simple	96.33 ± 3.33	98.29 ± 1.20	95.94 ± 4.09	0.9976 ± 0.0022	16.59
TB-Net as reported[3]		99.86	100.0	99.70	–	∼17

Table 1. Baselines across five seeds (mean ± std), against the originally reported numbers. The comparison with the last row is indicative only: the original cohort is no longer available (Section 1) and the *full* row is a capacity-matched reconstruction, not a port.

Two paired comparisons are worth stating precisely:

- **Faithful preprocessing helps the full model, not the compact one.** Pooling both architectures, faithful beats simple by 7.8 ± 4.3 AUC points with all five seeds agreeing (95% CI $[2.9, 8.5]$), but the effect is driven almost entirely by the full model, which degrades badly under the faithful pipeline's harder optimization landscape at this training budget; for the compact model the two pipelines differ by -0.33 ± 0.29 AUC points, a gap whose CI includes zero.
- **Inverse-frequency class weighting hurts.** Weighted cross-entropy *reduces* AUC by 0.99 ± 1.19 points against plain cross-entropy, with zero of five seeds favouring weighting (CI $[-1.98, -0.03]$). An earlier draft of this study had asserted, without running it, that class weighting would push sensitivity toward 99–100%; the measured result contradicts the expectation, which is precisely why the control was worth running.

A surprise we flag but do not over-interpret: the 4.32M reconstruction underperforms the 0.27M compact model under the identical ten-epoch budget. That budget was tuned for the small model, and the larger network is plausibly undertrained, so no conclusion about capacity should be drawn. What can be said is that nothing about this dataset requires 4M parameters to reach the high nineties.

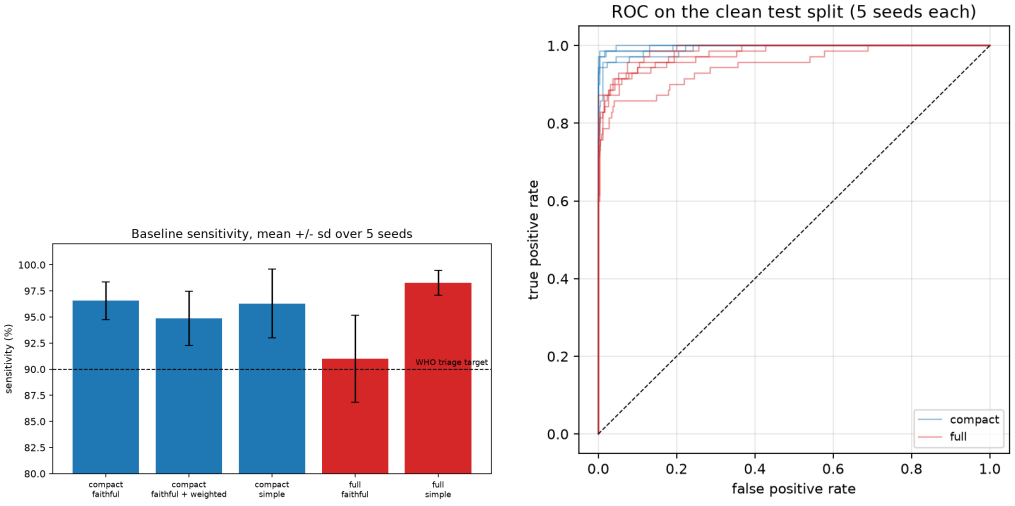

Figure 1. Left: per-seed test sensitivity for each architecture and preprocessing pipeline. Right: test-set ROC curves (per-seed), computed from the released per-sample probabilities.

3.1 Operating points and calibration

WHO target product profiles for TB triage tools ask for 90% sensitivity at 70% specificity[12]. From the released probabilities, the compact model achieves 98.0 ± 1.6% sensitivity at a 95% specificity operating point and 98.3 ± 1.2% at 90% specificity, comfortably clearing the profile on this internal test set. The full model manages 89.1 ± 2.6% and 92.6 ± 4.0% respectively. Calibration separates the two models more sharply than ranking does: the compact model's Brier score is 0.013 ± 0.008 against 0.112 ± 0.089 for the full model, so only the compact model's scores could responsibly be read as probabilities.
Of the 70 TB images in the test split, exactly one (`Tuberculosis-492.png`) is missed by all five seeds, and seven are missed by at least one; the consistently missed case is the kind worth inspecting by eye, being either genuinely hard or mislabelled.

4 Results: compression

4.1 Pruning

Arch	Kind / schedule	Target	Sensitivity	Accuracy	AUC
compact	unstructured, one-shot	25%	96.00 ± 0.64	99.19 ± 0.27	0.9979 ± 0.0013
compact	unstructured, one-shot	50%	93.71 ± 3.59	98.81 ± 0.48	0.9978 ± 0.0011
compact	unstructured, one-shot	75%	$\mathbf{7.14 \pm 15.97}$	84.52 ± 2.66	0.9901 ± 0.0073
compact	unstructured, iterative	75%	$\mathbf{94.86 \pm 0.78}$	99.10 ± 0.11	0.9985 ± 0.0003
compact	structured, one-shot	25%	95.14 ± 3.29	96.19 ± 5.49	0.9946 ± 0.0040
compact	structured, one-shot	50%	86.29 ± 8.84	95.76 ± 2.18	0.9791 ± 0.0235
compact	structured, one-shot	75%	86.57 ± 12.68	67.95 ± 37.92	0.9119 ± 0.0901
full	unstructured, one-shot	75%	0.00 ± 0.00	83.33 ± 0.00	0.8627 ± 0.2035
full	unstructured, iterative	75%	84.86 ± 7.19	96.76 ± 0.96	0.9816 ± 0.0098

Table 2. Pruning with five fine-tuning epochs after (or interleaved with) pruning. The one-shot 75% rows are the sensitivity cliff; the iterative rows reach the same final sparsity gradually and avoid it. Full-model one-shot rows at 25/50% are omitted for space and follow the same pattern (sensitivity 85–91%); complete tables are in the repository.

The headline finding of the compression study is a **sensitivity cliff** (Table 2, Figure 2): one-shot unstructured pruning at 75% sparsity collapses the compact model to 7.1 ± 16.0% sensitivity – and the full model to exactly 0% – while accuracy remains a deceptively respectable 84%, because the pruned network simply answers "normal" for everything. Accuracy on an imbalanced screening task can hide a model that has stopped detecting disease entirely. Note also that AUC stays at 0.99 across the cliff: the damage is to the decision threshold's placement rather than the ranking, which is exactly the failure mode that per-class metrics exist to catch.
The same final sparsity reached gradually (five prune–finetune cycles) recovers 94.86 ± 0.78% sensitivity at 99.10 ± 0.11% accuracy – statistically indistinguishable from the unpruned baseline, and the paired comparison against one-shot pruning at equal sparsity is an 87.7 ± 16.3 point sensitivity difference. Mild pruning is free or mildly beneficial: 25% unstructured pruning beats the unpruned baseline by 0.21 ± 0.27 AUC points (four of five seeds, CI touching zero). Structured (channel) pruning[8] degrades faster than unstructured at every strength, consistent with the compact model having little redundant channel capacity to remove.
One honesty note on file sizes: our structured pruning zeroes whole filters but no compaction pass physically removes them, so the reported FLOP savings are real while the on-disk size is unchanged. Size reduction in this study comes from quantization, not pruning.

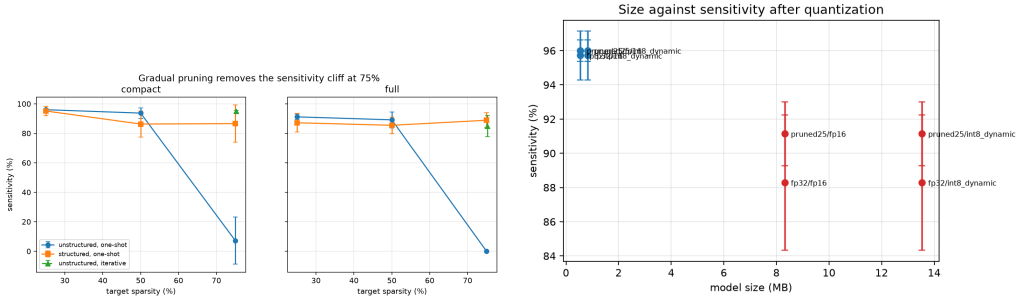

Figure 2. Left: sensitivity against sparsity for each pruning kind and schedule; the one-shot curves fall off a cliff at 75% that the iterative schedule avoids. Right: the size–sensitivity plane across every model variant in the study.

4.2 Distillation, with the control that matters

A MobileNetV3-Small student distilled from the compact teacher (KL-divergence soft targets[6] mixed with hard labels) reaches $97.71 \pm 1.63\%$ sensitivity at $93.48 \pm 4.88\%$ accuracy. Reported alone, that would look like a distillation success. The control – an identical student trained from scratch on hard labels only – reaches $97.71 \pm 1.92\%$ sensitivity at $93.38 \pm 6.66\%$ accuracy: sensitivity difference 0.00 ± 3.19 points, AUC difference -0.34 ± 0.31 points with zero of five seeds favouring distillation and a CI spanning zero (Figure 3). On this task, at this data scale, with an ImageNet-pretrained student, **distillation adds nothing over ordinary supervised training**, and any pipeline that skipped the control would have claimed otherwise.

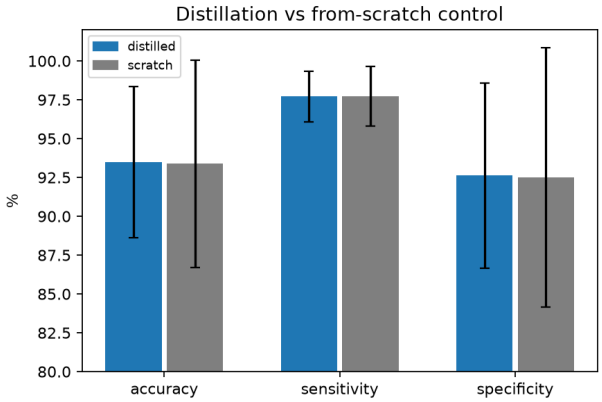

Figure 3. Distilled student against the from-scratch control, per seed. The two are statistically indistinguishable.

4.3 Quantization and measured latency

FP16 halves model size and dynamic INT8 leaves metrics untouched (Table 3); the INT8 file is larger than FP16 here because dynamic quantization converts only linear layers while convolutions stay in FP32. The deployment-relevant configuration – compact, 25% pruned, FP16 – is a **0.55 MB** model at $96.0 \pm 0.6\%$ sensitivity with a $\sim$2.5 ms CPU forward pass, three orders of magnitude under the 50 MB budget that motivated the study. An honest process note: the first complete run produced *no* latency numbers at all, because the ONNX export was wrapped in a broad exception handler that silently converted a missing `onnxruntime` dependency into what looked like a torch-version incompatibility. The defect log records the episode; the lesson – that `except Exception` handlers which

Arch	Base	Quant.	Sensitivity	Size (MB)	ONNX CPU latency (ms)
compact	fp32	–	95.71 ± 1.43	1.07	2.97 ± 0.95
compact	fp32	fp16	95.71 ± 1.43	0.55	–
compact	fp32	int8 dynamic	95.71 ± 1.43	0.82	–
compact	pruned25	fp16	96.00 ± 0.64	0.55	2.47 ± 0.06
full	fp32	–	88.29 ± 3.96	16.59	8.11 ± 2.85
full	fp32	fp16	88.29 ± 3.96	8.31	–

Table 3. Quantization is lossless at these strengths: accuracy, sensitivity and AUC are unchanged to reported precision under both FP16 and dynamic INT8, for both architectures, pruned or not. Latency is the median of 100 single-image CPU runs (mean ± std across seeds), measured on shared cloud vCPUs and therefore indicative rather than device latency.

print only the message cost more than they save – seems general.

5 Results: phone capture and external validation

5.1 Simulated phone-capture robustness

The deployment scenario motivating this work is a clinician photographing an X-ray film with a phone. We simulate that capture path with five distortions – defocus blur, brightness/contrast shift, specular glare, moiré interference and small rotations – applied singly and together.

Applied together, the distortions are devastating to models trained on clean images: the compact model falls from 98.8% to $46.5 \pm 5.1\%$ accuracy ($57.1 \pm 7.7\%$ sensitivity), and the full model to $40.6 \pm 3.4\%$. The per-distortion ablation (Figure 4) shows the two architectures fail differently: the compact model is destroyed by blur (a 78.6-point accuracy drop, collapsing to predicting "TB" for nearly everything) and rotation (46.3 points), while the full model's worst enemies are glare (48.1) and brightness (27.1).

Fine-tuning with phone-style augmentation recovers aggregate accuracy (compact: $86.1 \pm 0.7\%$ under full distortion) but at a cost that the aggregate hides: the compact model's sensitivity drops to $32.0 \pm 6.7\%$ under distortion and to $38.0 \pm 8.4\%$ *even on clean images*. The augmented model learned to survive the corruptions by becoming conservative, which is the wrong trade for a screening tool. The full model handles the trade better ($70.9 \pm 4.1\%$ distorted sensitivity) but remains far from clinical utility. We conclude that simulated augmentation alone does not solve the phone-capture problem for these models, and that real phone-photographed films – which we did not have – are the necessary next step.

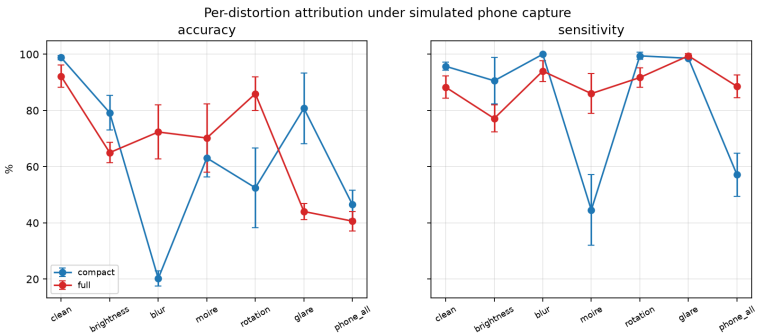

Figure 4. Per-distortion accuracy damage for each architecture. The compact model fails under blur and rotation; the full model under glare and brightness.

5.2 External validation: an unresolved below-chance result

The Montgomery (138 radiographs) and Shenzhen (662) cohorts[10] are the setting in which the original paper's numbers were reported, and the only place a delta against the original is even nominally meaningful. Our result is not a domain-shift finding but a puzzle: the compact model scores AUC 0.293 ± 0.036 pooled and 0.253 ± 0.050 on Shenzhen – *below* chance – with 1.1% sensitivity, while the full model sits at chance (0.533 ± 0.055). A systematically inverted ranking usually indicates a defect rather than a generalization failure, so we treated it as one and eliminated the two obvious causes:

1. **Labels are read correctly.** The parsed class balance exactly matches the published one (Montgomery 80 normal / 58 TB; Shenzhen 326 / 336), ruling out an inverted filename convention, and a separate defect in which lung segmentation masks were counted as radiographs (inflating Montgomery to 414 "images") was found and fixed before these numbers were produced.
2. **Preprocessing mismatch is not the cause.** Evaluating simple-pipeline models on simple-processed external images does not rescue the AUC; it lowers it (0.264 vs. 0.323 for compact). The failure is cohort-specific – Montgomery sits near chance under both pipelines while Shenzhen is strongly inverted – and survives a complete change of preprocessing.

A third possibility, that some Montgomery/Shenzhen images are inside the Rahman training cohort (which lists the NLM sets among its sources, making the evaluation optimistic rather than external), could not be settled: nearest-neighbour perceptual-hash distances between the two cohorts form a single smooth distribution with no duplicate spike, meaning the hash measures shared thoracic anatomy rather than image identity (Figure 5, right). Overlap therefore remains untested, and settling it needs embedding-space or pixel-level comparison.

We flag this stage as **unresolved and unpublishable as a finding**: the numbers are reported for completeness and to invite diagnosis, not as evidence about domain shift. What they do establish, at minimum, is that neither of our models transfers usefully to the original cohorts as processed by our pipeline – a sharp contrast with the near-perfect cross-cohort numbers of the original paper, and a reminder that internal test metrics on this dataset say little about deployment.

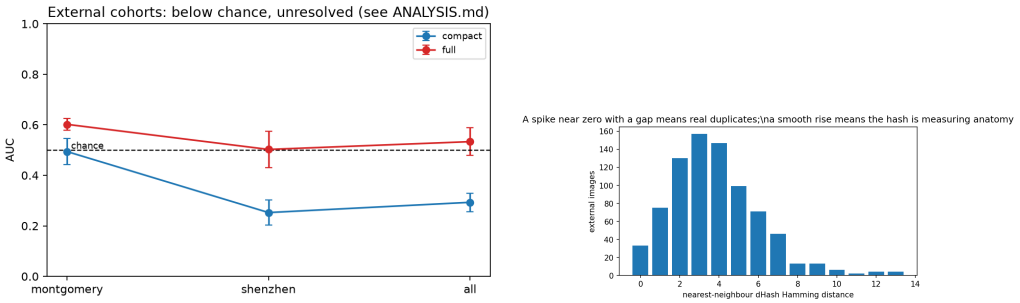

Figure 5. Left: external AUC by cohort and architecture; the dashed line is chance. Right: nearest-neighbour dHash distance from each external image to the training cohort; real duplicates would form a spike near zero separated by a gap, and none exists, so cohort overlap is untestable by this method.

6 Defects found along the way

ReScience exists partly so that the failures surrounding a result are recorded next to the result. Our repository's `docs/BUGS.md` logs eighteen defects with status and measured

impact; the ones that changed numbers or nearly did are summarized here.

- *Train/test leakage from duplicate files* (fixed; impact under 0.3 accuracy points – Section 2.1).
- *CSV rows written under the wrong headers*: iterative-pruning rows carried two fewer fields than the file header, shifting every column and making iterative pruning appear to score an arithmetically impossible 100% sensitivity at 99.94% accuracy. Reading the columns correctly gives 94.86 ± 0.78 – still the strongest compression result in the study, but a real number rather than an artifact.
- *The paper's preprocessing was never wired in*: the described pipeline existed in the codebase but nothing imported it, so early results came from a pipeline the write-up did not describe. Promoted to the faithful/simple experimental axis.
- *A forced re-run silently dropped an arm*: regenerating the baseline CSV deleted the class-weighting rows, and the finding they support (weighting hurts, five of five seeds) would have vanished without error had the rows not been recovered and merged.
- *The quantization stage silently processed only the first architecture* passed to it; the missing full-model rows were recovered by a rerun.
- *An earlier draft cited scripts that did not exist and presented an expectation as a result*; both practices were removed, and the expectation, once tested, turned out to be wrong (Section 3).

The pattern across these defects is that almost none of them crashed. Silent misalignment, silent globbing of the wrong files, silent exception swallowing and silent arm-dropping all produced plausible numbers, and each was caught only by cross-checking results against arithmetic possibility, published cohort counts, or the repository's own history.

7 Limitations

1. The *full* model is a capacity-matched reconstruction; no statement here is a claim about the original TB-Net weights or exact topology, which are unrecoverable.
2. The training cohort differs from the original study's, so the comparison in Table 1 is indicative, not a replication delta.
3. The ten-epoch budget was tuned on the compact model; the full model is plausibly undertrained, and no capacity conclusion should be drawn from its underperformance.
4. Phone capture is simulated; no real phone-photographed films or on-device Android measurements are included, and CPU latency was measured on shared cloud vCPUs.
5. External validation is unresolved (Section 5.2) and possible overlap between the training cohort and the external cohorts is untested.
6. Structured pruning's file-size savings are not realized because no compaction pass removes zeroed filters.

8 Conclusion

Judged narrowly as a replication, the outcome is mixed and mostly outside our control: the original architecture cannot be rebuilt from released materials and the original cohort no longer exists in the form it was used, so the published 99.86/100/99.7 operating point can be neither confirmed nor refuted. Judged as a test of the paper's substantive claim – that a small attention-condenser CNN can screen for tuberculosis on this cohort at very high sensitivity – the claim broadly survives: a 0.27M-parameter model built from the described motifs reaches $96.6 \pm 1.8\%$ sensitivity at $98.5 \pm 1.8\%$ accuracy on leak-checked splits, and compresses to 0.55 MB with no measurable loss.
The extensions carry the more practical lessons. Compression at moderate strength is free, but one-shot high-sparsity pruning fails in the most dangerous possible way for a screening tool – sensitivity collapse behind a healthy-looking accuracy – and the fix (iterative pruning) is cheap. Distillation, once a from-scratch control is run, contributes nothing here. And the deployment scenario that motivates lightweight TB models is exactly where they break: simulated phone capture halves accuracy, augmentation buys aggregate robustness by sacrificing the sensitivity that justifies the tool, and transfer to the original external cohorts fails in a way we cannot yet explain. A sub-megabyte, 2.5-millisecond model is achievable; a deployable one, on this evidence, is not yet.

Acknowledgements

We thank the maintainers of the public Rahman, Montgomery and Shenzhen datasets, and the authors of TB-Net for releasing their training and evaluation code, without which even a partial reproduction would have been impossible.

References

1. World Health Organization. **Global Tuberculosis Report 2024**. 2024. URL: https://www.who.int/teams/global-tuberculosis-programme/tb-reports.
2. Z. Z. Qin et al. "Using artificial intelligence to read chest radiographs for tuberculosis detection: A multi-site evaluation of the diagnostic accuracy of three deep learning systems." In: **Scientific Reports** 9 (2019), p. 15000. DOI: 10.1038/s41598-019-51503-3.
3. A. Wong, J. R. H. Lee, H. Rahmat-Khah, A. Sabri, A. Alaref, and H. Liu. "TB-Net: A Tailored, Self-Attention Deep Convolutional Neural Network Design for Detection of Tuberculosis Cases From Chest X-Ray Images." In: **Frontiers in Artificial Intelligence** 5 (2022). DOI: 10.3389/frai.2022.827299.
4. S. Han, J. Pool, J. Tran, and W. J. Dally. "Learning both Weights and Connections for Efficient Neural Networks." In: **Advances in Neural Information Processing Systems**. Vol. 28. 2015.
5. B. Jacob, S. Kligys, B. Chen, M. Zhu, M. Tang, A. Howard, H. Adam, and D. Kalenichenko. "Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference." In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. 2018, pp. 2704–2713.
6. G. Hinton, O. Vinyals, and J. Dean. "Distilling the Knowledge in a Neural Network." In: **arXiv preprint arXiv:1503.02531** (2015).
7. T. Rahman et al. "Reliable Tuberculosis Detection Using Chest X-ray With Deep Learning, Segmentation and Visualization." In: **IEEE Access** 8 (2020), pp. 191586–191601. DOI: 10.1109/ACCESS.2020.3031384.
8. H. Li, A. Kadav, I. Durdanovic, H. Samet, and H. P. Graf. "Pruning Filters for Efficient ConvNets." In: **International Conference on Learning Representations (ICLR)**. 2017.
9. A. Howard et al. "Searching for MobileNetV3." In: **Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)**. 2019, pp. 1314–1324. DOI: 10.1109/ICCV.2019.00140.
10. S. Jaeger, S. Candemir, S. Antani, Y.-X. J. Wáng, P.-X. Lu, and G. Thoma. "Two public chest X-ray datasets for computer-aided screening of pulmonary diseases." In: **Quantitative Imaging in Medicine and Surgery** 4.6 (2014), pp. 475–477. DOI: 10.3978/j.issn.2223-4292.2014.11.20.
11. J. Pineau, P. Vincent-Lamarre, K. Sinha, V. Larivière, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and H. Larochelle. "Improving Reproducibility in Machine Learning Research (A Report from the NeurIPS 2019 Reproducibility Program)." In: **Journal of Machine Learning Research** 22.164 (2021), pp. 1–20.

12. World Health Organization. **WHO consolidated guidelines on tuberculosis. Module 2: Screening – systematic screening for tuberculosis disease**. 2021. URL: https://www.who.int/publications/i/item/9789240022676.