

LungLens: chest X-ray tuberculosis screening, and what starting from chest X-rays changes about where a CNN looks

Zainab Hamid · BS Computer Science (7A), Superior University, Lahore · Deep Learning, instructor Dr. Munir Ahmad · November 2026

Live demo: <https://lunglens-tb.vercel.app> · Code & weights: <https://github.com/zyna-b/lunglens> · Video: <https://www.youtube.com/watch?v=caT1J9Lhu4A>

Abstract. Pakistan carries the world's fifth-largest tuberculosis (TB) burden, and people trained to read chest X-rays for TB are scarce outside large cities. We build LungLens, a three-class screening model (healthy / sick but not TB / TB) trained on TBX11K, a dataset of 8,399 labelled X-rays whose TB lesions were boxed by radiologists. We compare a CNN trained from scratch with two DenseNet-121 networks that differ only in their starting weights: ImageNet photographs or chest X-rays from seven public datasets. On 1,800 held-out test X-rays the deployed model (DenseNet-121, ImageNet start) reaches 99.2% accuracy (95% CI 98.8%–99.6%), 98.4% balanced accuracy and a TB-versus-rest AUC of 0.9996. At a screening threshold fixed on validation data it detects 100.0% of TB cases at 95.9% specificity. Because every model ends in global average pooling, its class activation map comes from the same forward pass; the hottest point of the TB map falls inside a radiologist's box in 36.0% of TB test X-rays, against 9.8% for a random point. The model runs entirely in the browser through ONNX Runtime Web, so no X-ray leaves the user's device.

1. Problem

The WHO estimates about 670,000 new TB cases in Pakistan in 2024, 6.3% of the world total and the fifth-highest national burden [1]. Many cases are never diagnosed. Chest radiography is cheap and widely available, and since 2021 the WHO has recommended computer-aided detection (CAD) software as an alternative to a human reader when screening people aged 15 and over [2]. A screening tool must above all not miss TB: the WHO's minimum target for a triage test is 90% sensitivity with 70% specificity [3].

Most student TB projects use the Kaggle "TB Chest X-ray" database [5], in which TB and normal images come from different hospitals and countries. A model can then learn which hospital an image came from instead of which disease it shows, a shortcut documented for pneumonia and COVID-19 models [13, 14]. We avoid this by using TBX11K [4], whose classes share their sources and which includes a class of patients who are *sick but do not have TB*: the realistic difficulty is separating TB from other lung disease, not from a healthy chest.

Contributions. (1) A controlled comparison of three starting points with identical data, augmentation and schedule. (2) A screening operating point chosen on validation data, and an account of why our first choice of rule failed. (3) A quantitative check of the heatmaps against radiologists' boxes (the pointing game [12]). (4) A deployed, private, in-browser tool with code, weights and data splits published.

2. Related work

CheXNet [6] showed that a DenseNet-121 [7] fine-tuned on ChestX-ray14 can match radiologists on pneumonia. Liu et al. [4] released TBX11K with bounding boxes and reported classification and detection baselines. Rahman et al. [5] report about 98% accuracy on their TB database, partly reflecting the source difference above. TorchXRayVision [8] provides DenseNet-121 weights trained on several large chest X-ray collections (NIH, CheXpert, MIMIC-CXR, PadChest and others), which we use as the domain-specific starting point. Class activation mapping (CAM) [10] and Grad-CAM [11] visualise evidence; the pointing game [12] turns such maps into a number.

3. Data

TBX11K (simplified Kaggle release, CC BY 4.0) contains 512×512 frontal X-rays labelled healthy, sick-but-not-TB, or TB (active or latent), with radiologist boxes around TB regions. The official test labels are not public, so we use TBX11K's labelled validation set as our untouched test set and split its training set 90/10 (stratified, seed 42) into training and validation. The exact file list is in `data/splits.csv`.

Split	Healthy	Sick, not TB	TB	Total
Train	2,700	2,700	539	5,939
Validation	300	300	60	660
Test	800	800	200 (with 1–4 boxes each)	1,800

Images are converted to grayscale and resized to 384×384 : TB lesions are small and lose detail at the 224 px usual for ImageNet models. TB is only 9% of the training set, so the loss weights each class by inverse frequency. Augmentation runs on the GPU per image: small rotations, shifts and zooms, exposure, contrast and gamma changes, noise and blur. We use no horizontal flips: a mirrored X-ray puts the heart on the wrong side, which never occurs in practice.

4. Method

All three models take a grayscale image in $[0, 1]$, normalise it internally, and end in ReLU $\rightarrow$ global average pooling $\rightarrow$ one linear layer. For such a network the class activation map $M_c = \sum_k w_{ck} F_k$ (F the final feature maps, w the linear weights) averages exactly to the class logit [10], so the deployed ONNX model returns probabilities and 12×12 maps in one pass, and the browser can draw the heatmap without gradients.

- **CNN from scratch:** five blocks of two 3×3 convolutions with batch normalisation, $32 \rightarrow 256$ channels, 2.4M parameters.
- **DenseNet-121, ImageNet start** [7, 9]: the CheXNet architecture, 7.0M parameters.
- **DenseNet-121, chest-X-ray start:** the same architecture with a single-channel stem, initialised from TorchXRayVision's "all" weights [8].

Setting	Value
Input	grayscale, 384 × 384, values in [0, 1]
Optimiser	AdamW, weight decay 0.05
Learning rate	3e-4 (DenseNets), 2e-3 (scratch); 5% linear warm-up then cosine decay
Epochs	20 (DenseNets), 40 (scratch); best epoch by validation balanced accuracy
Batch size	32, bfloat16 mixed precision
Loss	cross-entropy, class weights $\propto 1 / \text{frequency}$ (TB ≈ 3.7 , others ≈ 0.73)
Augmentation (GPU)	rotation $\pm 7^\circ$, zoom 0.92–1.12, stretch $\pm 7\%$, shift $\pm 6\%$, gamma/contrast/brightness $\pm 15\text{--}25\%$, noise, blur ($p = 0.2$); no flips
Hardware	1 × NVIDIA L4 (Google Colab); about 51 s per DenseNet epoch
Seed	42 (splits and training)

Screening rule. A person is referred for a lab test (sputum smear or GeneXpert) when the TB probability is at least a threshold. We fix the threshold at 95% specificity on the 600 non-TB validation X-rays, i.e. about 5% unnecessary referrals, and then read off sensitivity on the test set. The deployed threshold is 0.0070.

A lesson on choosing thresholds. Our first rule picked the threshold that gave 90% sensitivity on the 60 validation TB X-rays. On the test set that threshold gave only 85.0% sensitivity (100.0% specificity): with so few positives the estimate is noisy, and the model is more confident on validation images than on test images. Anchoring the threshold on 600 negative images instead gives a far more stable estimate.

5. Results

Model	Params	Accuracy	95% CI	Bal. acc.	Macro F1	TB AUC	95% CI	Sens.	Spec.	Pointing game	CPU ms
CNN from scratch	2.35M	96.9%	96.1%–97.7%	97.0%	0.948	0.9952	0.9921–0.9979	98.0%	94.6%	34.0%	246
DenseNet-121, ImageNet start (deployed)	6.96M	99.2%	98.8%–99.6%	98.4%	0.987	0.9996	0.9991–0.9999	100.0%	95.9%	36.0%	313
DenseNet-121, chest-X-ray start	6.95M	98.3%	97.7%–98.9%	97.0%	0.972	0.9965	0.9936–0.9984	98.0%	95.8%	57.5%	294

Test set: 1,800 X-rays. Confidence intervals: 1,000 bootstrap resamples. Sensitivity and specificity for TB vs not-TB at each model's own validation threshold. Pointing game: the hottest point of the TB map lies inside a radiologist's box (random point: 9.8%). CPU ms: ONNX Runtime, one image, 2-core ARM server.

Starting from chest X-rays instead of ImageNet changed test balanced accuracy by -1.4 points, TB AUC by -0.0031 and the pointing-game hit rate by +21.5 points. The TB AUC confidence intervals do not overlap; the domain-pretrained network does, however, place its evidence inside the doctors' boxes more often. The CNN trained from scratch reaches 97.0% balanced

accuracy and a TB AUC of 0.9952, against 98.4% and 0.9996 for the deployed model: pretraining is worth 1.5 points of balanced accuracy here.

Which model to deploy? Our pre-declared rule (best validation balanced accuracy) selects the DenseNet-121, ImageNet start model, while the DenseNet-121, chest-X-ray start model places its evidence inside the radiologists' boxes more often (57.5% vs 36.0%). For a clinical tool, evidence that lines up with radiologists' marks can be worth a small loss in accuracy. We keep the rule rather than re-selecting after seeing test results, and the website ships both models with a switch, so users can compare the two heatmaps on the same X-ray.

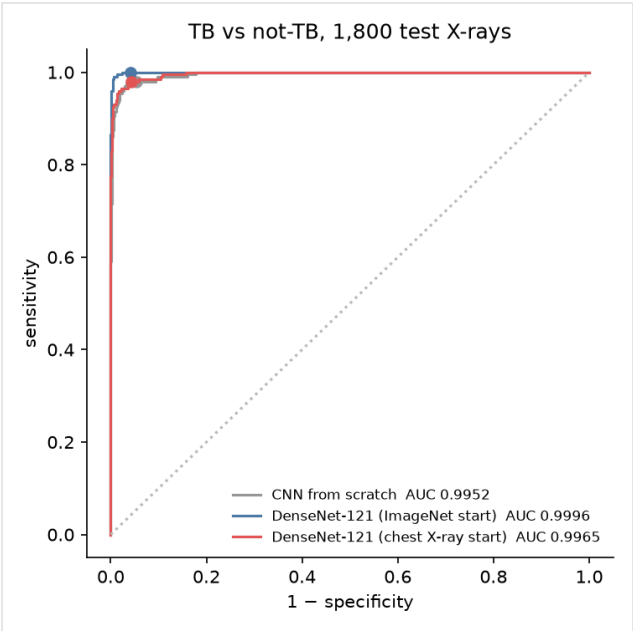


Figure 1. TB-vs-rest ROC curves on the test set; dots mark each model's operating point.

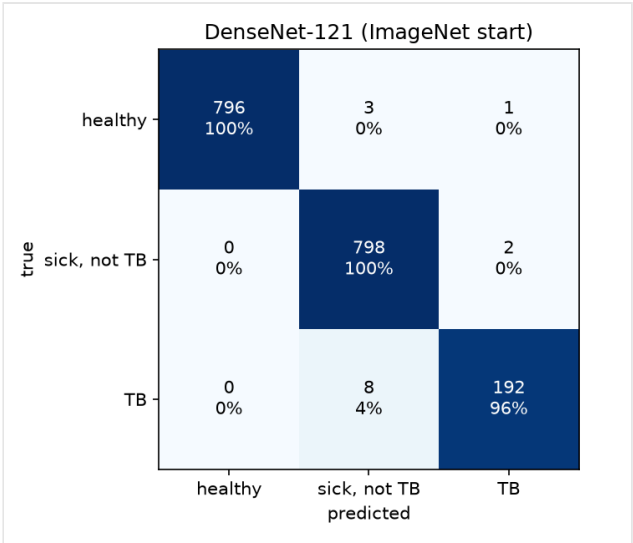


Figure 2. Confusion matrix of the deployed model (rows: true class).

Class	Precision	Recall	F1	Test images
healthy	1.000	0.995	0.997	800
sick but not tb	0.986	0.998	0.992	800
tb	0.985	0.960	0.972	200

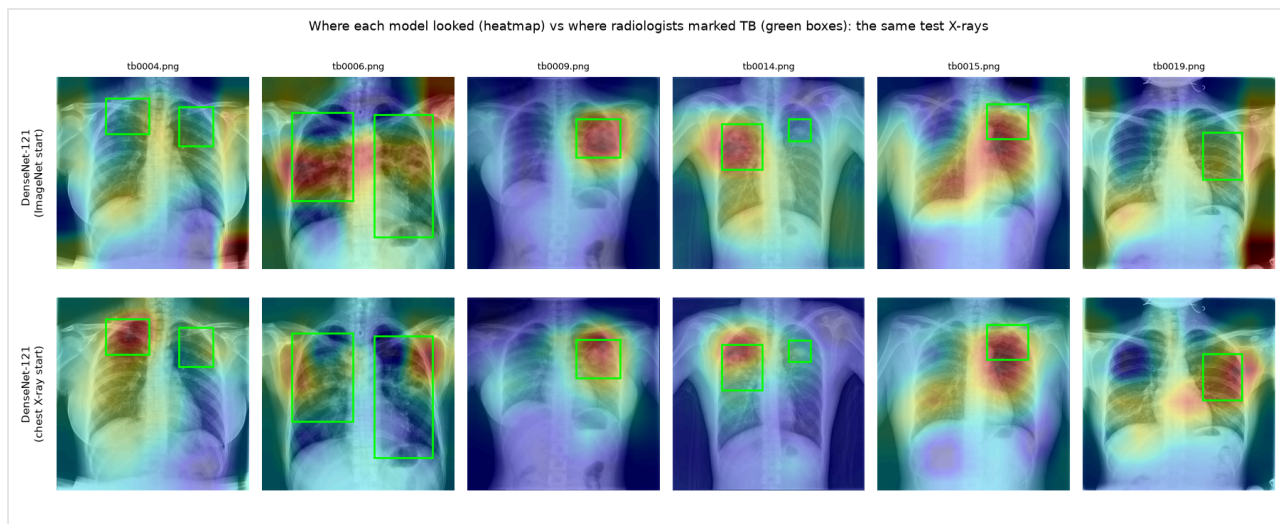


Figure 3. Class activation maps of the deployed model on TB test X-rays it classified as TB, with the radiologists' boxes in green.

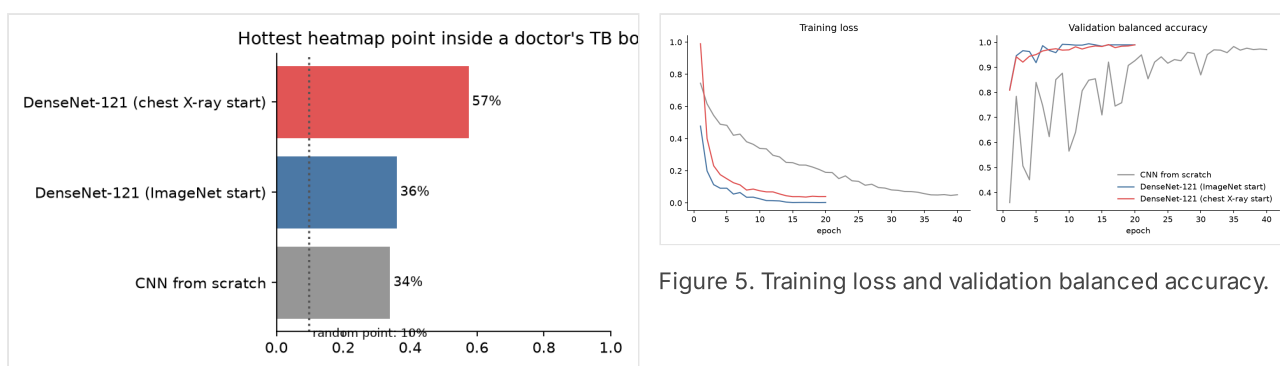


Figure 4. Pointing game by model.

Figure 5. Training loss and validation balanced accuracy.

6. Error analysis

At the screening threshold the deployed model misses 0 of 200 TB X-rays. It detects 160 of 160 **active** TB cases (100.0%) and 40 of 40 **latent** TB cases (100.0%); latent TB leaves subtler, often calcified traces, so the gap is clinically plausible. It refers 16 of 800 healthy and 49 of 800 sick-but-not-TB patients. In the three-class decision the most common confusion is between healthy and sick-but-not-TB (3 healthy called sick, 0 sick called healthy), which matters less for TB screening than confusions involving TB (8 TB called non-TB by the top class, 3 non-TB called TB).

7. Deployment

The deployed model is exported to ONNX (opset 17, 28.2 MB) with softmax and CAM inside, and to TorchScript. The website loads it once and runs it in the browser with ONNX Runtime Web (WebAssembly), about a second per X-ray on a laptop; the X-ray never leaves the device, which matters for medical images. The GitHub repository contains the code, the exact splits, all weights with SHA-256 hashes in `model_card.json`, the executed Colab notebook, and a GitHub Actions workflow that runs `predict.py` and `evaluate.py` on 12 bundled labelled X-rays at every push.

```
git clone https://github.com/zyna-b/lunglens && pip install -r requirements.txt && python evaluate.py
```

8. Limitations and ethics

- TBX11K was collected in Chinese hospitals; performance on Pakistani X-ray machines and populations is unknown and must be measured before any use.
- The model is for adults; WHO's CAD recommendation covers ages 15 and over [2].
- Our test set is TBX11K's validation set, so numbers are not directly comparable with the official (hidden-label) test leaderboard.
- A heatmap shows where evidence is, not a lesion outline, and a high pointing-game score does not make the model a diagnostician.
- LungLens is a teaching and research demo, not a medical device. A positive screen needs a lab test; a negative one does not rule TB out.

9. Conclusion

A DenseNet-121 fine-tuned on TBX11K screens chest X-rays for TB with a TB AUC of 0.9996, separates TB from other lung disease, and points at radiologist-marked lesions far more often than chance, while running privately in a browser. The next steps are a test on X-rays from Pakistani hospitals, calibration of the threshold on that population, and lung segmentation to keep the model's attention inside the lungs.

References

1. World Health Organization. *Global Tuberculosis Report 2025*. Geneva: WHO; 2025.
2. World Health Organization. *WHO consolidated guidelines on tuberculosis. Module 2: screening – systematic screening for tuberculosis disease*. Geneva: WHO; 2021.
3. World Health Organization. *High-priority target product profiles for new tuberculosis diagnostics: report of a consensus meeting*. Geneva: WHO; 2014.
4. Liu Y, Wu Y-H, Ban Y, Wang H, Cheng M-M. Rethinking computer-aided tuberculosis diagnosis. *CVPR*, 2020.
5. Rahman T, Khandakar A, Kadir MA, et al. Reliable tuberculosis detection using chest X-ray with deep learning, segmentation and visualization. *IEEE Access* 8, 2020.
6. Rajpurkar P, Irvin J, Zhu K, et al. CheXNet: radiologist-level pneumonia detection on chest X-rays with deep learning. arXiv:1711.05225, 2017.
7. Huang G, Liu Z, van der Maaten L, Weinberger KQ. Densely connected convolutional networks. *CVPR*, 2017.
8. Cohen JP, Viviano JD, Bertin P, et al. TorchXRayVision: a library of chest X-ray datasets and models. *MIDL*, PMLR 172, 2022.
9. Deng J, Dong W, Socher R, Li L-J, Li K, Fei-Fei L. ImageNet: a large-scale hierarchical image database. *CVPR*, 2009.
10. Zhou B, Khosla A, Lapedriza A, Oliva A, Torralba A. Learning deep features for discriminative localization. *CVPR*, 2016.
11. Selvaraju RR, Cogswell M, Das A, et al. Grad-CAM: visual explanations from deep networks via gradient-based localization. *ICCV*, 2017.
12. Zhang J, Bargal SA, Lin Z, Brandt J, Shen X, Sclaroff S. Top-down neural attention by excitation backprop. *IJCV* 126, 2018.
13. Zech JR, Badgeley MA, Liu M, et al. Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs. *PLOS Medicine* 15(11), 2018.
14. DeGrave AJ, Janizek JD, Lee S-I. AI for radiographic COVID-19 detection selects shortcuts over signal. *Nature Machine Intelligence* 3, 2021.

15. Loshchilov I, Hutter F. Decoupled weight decay regularization. *ICLR*, 2019.
16. ONNX Runtime contributors. ONNX Runtime Web. <https://onnxruntime.ai>, 2024.

This report was drafted with the help of an AI assistant (Claude) from the project's code and results, as the course requires, and checked against them.