Vision Mamba Network for Tuberculosis Detection from Chest X-Rays

Goli Harshavardhan[1], Prof. I. Kullayamma[2]
[1] *M. TECH Student, Department of Electronics and Communication Engineering, Sri Venkateswara University College of Engineering, Tirupati 517502, Andhra Pradesh, India*
[2]*Professor, Department of Electronics and Communication Engineering, Sri Venkateswara University College of Engineering, Tirupati 517502, Andhra Pradesh, India*
doi.org/10.64643/IJIRTV13I4-208677-459

***Abstract*—Detecting Tuberculosis (TB) in chest radiographs is not a simple problem for a model because it has to notice very small, localized lesions while also keeping the overall context of both lung fields, and the aim here is to address this local-plus-global challenge. In this paper, MSVMamba-TB is a hierarchically nested multi-scale Vision Mamba network classifier of 1 channel chest X-ray (CXR) into normal and TB defined in binary classification. We compared it with a deep learning technique, namely Vision Transformer (PaperViT), and two other approaches: ConvNeXtV2-Pico and EfficientNet-B3. Not only that, but all the models were also used in conjunction, fine-tuned, and verified on the same computer with the same specs and setup to keep the comparison for every model as similar as possible. In our proposed model, a multi-scale two-dimensional selective scan, a convolutional feed-forward block, squeeze-and-excitation channel weighting, global pooling, and a small binary head are put together in one architecture. Experiment A contained a total of 4, 200 patient-level records, of which 3, 500 were normal and 700 were tuberculosis. The data distribution was an 80/10/10 split, whereas Experiment B consisted of 7, 000 balanced records with a 70/15/15 data split. Both scenarios have the same preprocessing, augmentation, sampling, loss, optimizer, scheduler, early stopping, and other model features, and no model is given a separate advantage in terms of training. In Experiment A the model obtained accuracy of 99.05%, a sensitivity of 97.14%, and precision of 97.14%, with the F1-score of 97.14% being the experimental results. Experiment B was able to reach an accuracy of 98.38%, a precision of 99.42%, a sensitivity of 97.33%, an F1-score of 98.36%, and the highest score was on the AUC at 0.9992. The model proposed here ranked top in both accuracy and F1-score and made fewer mistakes. Therefore, our proposed model aims to have good prediction performance and a small size. Other than that, this is a very small model with only 7.012M trainable parameters, which, relative to PaperViT with 13.527 M, is only 48.17%. Detailed epoch records, confusion matrices, ROC curves, error analysis, activation maps, and cases where failures were made are presented for all four networks. Hence, the results are not only reported as a single accuracy number. Moreover, these results indicate that MSVMamba-TB is an encouraging candidate for fast tuberculosis screening, which is useful for computer-aided TB screening.**

***Index Terms*—Chest X-ray, Computer-aided detection, ConvNextV2, Deep learning, EfficientNet, Multi-Scale Scanning, Selective State-Space Model, Tuberculosis, Vision Mamba, Vision Transformer.**

I. INTRODUCTION

Tuberculosis is, even today, one of the most deadly infectious diseases around the world, and if we look at the WHO numbers for the year 2023, we can say that nearly 10.8 million people got sick with TB, and something like 1.25 million of them actually died from it, and what makes this even worse is that a big chunk of the problem comes from people who are simply never diagnosed at all; this is what is put forth in [1]. Now if we talk about chest radiograph, it is basically the cheapest and also the most portable tool that we have for screening pulmonary TB, but the thing is, when a human is reading a chest X-ray (CXR), the result depends quite a lot on who exactly is doing the reading, and in the places where the TB burden is highest, there simply are not enough people who can read these films properly, and this is more or less the reason why computer-aided detection (CAD) tools have become such an interesting direction for TB triage, as can be seen in [2], [11], [12]. If we look at what kind of models such a tool could be built on, there

are mainly three families that we can think of: convolutional neural networks (CNNs), things like EfficientNet [23] and ConvNeXt V2 [24], then Vision Transformers (ViTs) [5], and then, only very recently, the visual state-space models such as Vision Mamba [7], VMamba [8], and also the multi-scale version MSVMamba [9], in which a selective scan is what gives the global context, and this comes at a cost that is roughly linear, which is a nice property to have. It is important for us to underline that TB has the biggest epidemic load in India, and radiologists who are capable of interpreting these screening radiographs mostly work in the urban hospitals, whereas the films themselves go for screening in the remote rural outreach camps where TB incidence is the highest.

Also important is that since tuberculosis is the kind of disease which presents itself differently in various parts of the lungs, one can have on a single chest radiograph a small nodule of only 3 mm size, a cavity of a few millimetres' diameter, and also diffuse infiltration which is present in both lungs. Because of this, a plain CNN needs to go very deep before its receptive field is able to cover both lungs together, and meanwhile a ViT is able to see the whole image, but the price for that grows with the square of how many tokens there are, so it is not free either. What selective state-space models like Mamba [6] do is they offer what can be called a content-conditional recurrent architecture, and this is roughly of linear complexity; by doing this, it changes the trade-off we just talked about; and then the MSVMamba construction [9] adds, inside each block, at the very least, a coarse scanning path, which, in our opinion, is what actually captures the essence of how multi-scale the TB findings tend to be. That said, whether this really does lead to a better performance for TB detection specifically, or not, is still something that needs to be confirmed, and it needs to be confirmed under a protocol that is objective, which is why we, the authors, decided to freeze down one single pipeline, one optimization strategy, patient-disjoint subsets, and one evaluation framework, and this was applied the same way across all four models.

So, what each of the following points does is basically summarise the main contributions of this work.

• We put forth MSVMamba-TB, which combines multi-scale four-direction selective scanning, a convolutional feed-forward block, and squeeze-and-excitation channel recalibration; all of this is meant for one-channel 224×224 chest X-rays, while having only 7.012 million trainable parameters, which is not a lot.

• There is an imbalance experiment that was run, involving 4,200 images with a 5:1 ratio, and also a balanced experiment with 7,000 images at a 1:1 ratio, and both of these used a partitioning that was patient-level and leakage-free. A common protocol was applied in an identical way to a ViT baseline and also to two CNNs that are fairly recent.

• Finally, what we provide is a complete set of metrics, confusion matrices, epoch histories, an error analysis, activation maps, and also a film-by-film failure analysis, and the conclusions that get drawn are only the ones which turn out to be the same across both of the regimes we tested.

II. RELATED WORK

Goletti et al. [1] quantified, from the WHO global TB report of 2024, the diagnostic gap that is the public-health reason for automatic CXR screening tools. Rahman et al. [2] combined deep learning, lung segmentation and visualization over a large curated public collection, the group behind the repositories both our experiments are built on [2], [25], [26] set the precedent of showing class-activation maps next to the classification; Kotei and Thirunavukarasu [11] point to data imbalance and generalization as the recurring problems of TB detection, exactly what we probe with the imbalanced Experiment A and the balanced Experiment B, and Sharma et al. [12] also visualize the infected region. For convolutional networks. Das et al. [14] developed lightweight network architectures and used a softmax layer with orthogonal weights, while Nafisah and Muhammad [16] demonstrated 99.1% accuracy with an EfficientNet-based network [23] and explainable AI. That is a big factor that we retain EfficientNet-B3 as a reference. Further, Huy and Lin [29] and Xu and Yuan [30] enhanced the DenseNet model and integrated the coordinate attention technique into the CNN framework. Finally, from Woo et al. [24], came ConvNeXt V2. On the transformer side, the ViT goes back to Dosovitskiy et al. [5]. Venkatachalam et al. [3] proposed TB detection with a convolutional-stem ViT and CLAHE-based contrast enhancement, and it is their paper that our existing system is based on: their design motivated the PaperViT baseline rebuilt here and the CLAHE step of our common pipeline. Vanitha et al. [4] combined a

ViT with Grad-CAM, Thirunavukarasu et al. [15] studied the explainability of a ViT-based TB diagnosis on chest X-rays, Yulvina et al. [21] put a ViT together with a CNN for multi-label TB anomalies, El-Ghany et al. [22] made a ViT–PCA hybrid, and Rahman et al. [13] went further with TB-CXRNet toward drug-resistant TB, which marks the boundary of our binary task. On the state-space side of the development, first came Mamba from Gu and Dao [6], then Zhu et al. [7] introduced Vision Mamba. VMamba was proposed by Liu et al. [8], whose team also integrated the SS2D cross-scan module inside SS2D that they proposed to use at every stage of our network. Shi et al. [9] were the first to suggest the multi-scale VMamba, which is a combination of a high-resolution scan path and a downsampled one inside each block; our idea is exactly this one, which works for TB as well. We use it on TB and compare it with the transformer and CNN baselines that the original paper did not do, and Hedhoud et al. [10], who have been using plain Vision Mamba for TB, have neither looked at the multi-scale design nor have they tried a four-model protocol which shares the preprocessing steps. Liang et al. [17], Zhang et al. [18], Saurina et al. [19], and Zhu et al. [20] did their studies using other modalities, CT, sputum, and laboratory results, so their goals are entirely different from ours in terms of CXR screening; Grad-CAM[27] and Score-CAM[28] are the two explanation methods we have chosen. Table 1 lists the most closely related studies and where this work fits in.

Table 1. Comparison of the closely related studies and a summary of where the current work sits.

Study	Main method	Relation to this work
Rahman et al. (2020)	CNNs + segmentation + CAM	Source of Exp. A; CAM precedent
Venkatachalam et al. (2025)	Conv-stem ViT, CLAHE input	Base paper (ViT, CLAHE)
Vanitha et al. (2025)	ViT + Grad-CAM	Transformer + explainability
Das et al. (2024)	Orthogonal-softmax CNN	Case for compact classifiers
Nafisah & Muhammad (2024)	EfficientNet-B3 + XAI	Reason for EfficientNet-B3
Yulvina et al. (2024)	Hybrid ViT–CNN	Global + local modelling
El-Ghany et al. (2024)	ViT–PCA hybrid	Further transformer route
Hedhoud et al.(2025)	Plain Vision Mamba	Earlier state-space use for TB
Huy & Lin (2023)	Improved DenseNet	Classical CNN reference
Xu & Yuan (2022)	CNN + coordinate attention	Attention recalibration
This work	MSVMamba-TB, multi-scale scan	Four-model protocol, two regimes

III. MATERIALS AND METHODS

A. Datasets and patient-level partitions

The paper used two publicly published chest X-ray sets, one for each of the experiments, so that the suggested framework can be assessed both times: once when TB is the minority class and a second time when the two classes are balanced. For Experiment A, we took the Tuberculosis (TB) chest X-ray dataset hosted on Kaggle [2], [25]: 4,200 postero-anterior films stored as 1024×1024 PNGs, 3,500 normal and 700 TB-positive, a 5:1 ratio close to what a screening system sees in the field, and the file naming (Normal-<id>.png, Tuberculosis-<id>.png) carries a patient identity that allows a genuinely patient-level split. For Experiment B, we took the Tuberculosis (TB) Chest X-ray Database released on IEEE DataPort [26], from which 7,000 radiographs were used, exactly half normal and half TB, so that with 525 TB test images against 70 a single false negative shifts the sensitivity by only about 0.19 percentage points instead of 1.43. The partitions are stratified, 80/10/10 in Experiment A and 70/15/15 in Experiment B, with the patient identifiers grouped so that no patient lands in more than one subset (the cross-split overlap is zero); the validation part serves only to pick the checkpoint and

to stop early, and the test part is locked and touched only once. Table 1 gives the exact counts.

Table 2. The two experimental datasets and their recorded split distributions.

Experiment	Subset	Normal	TB	Total	Share
A (4,200, 5:1; [2],[25])	Training	2,799	560	3,359	80 %
	Validation	351	70	421	10 %
	Test (locked)	350	70	420	10 %
B (7,000, 1:1; [26])	Training	2,449	2,450	4,899	70 %
	Validation	525	525	1,050	15 %
	Test (locked)	526	525	1,051	15 %

B. Common image preprocessing pipeline

Each radiograph follows an ordered and deterministic pipeline. All the statistics used in the pipeline are computed from the training data only. Since some files are stored with three channels, first map a colour image to one luminance channel using the standard luma weights.

$I_{gray} = 0.2989\, I_R + 0.5870\, I_G + 0.1140\, I_B$ (1)

The first step was to make sure that all four models used the same input. Contrast-limited adaptive histogram equalization (CLAHE) is used, where pixel intensities are redistributed over 8×8 blocks. The tile histogram is constrained by clipping the pixels at 2.0 level and the mapping is done using the local cumulative distribution function for each intensity r,

$s(r) = \frac{L-1}{N_{tile}} \sum_{k=0}^{r} H(k)$ (2)

where L is 256 for an eight-bit image, and H̃ is the clipped and normalized tile histogram, with bilinear interpolation between neighbouring tiles at the block boundaries; we kept CLAHE, which comes from the base paper [3], since it pulls the low-contrast apical and parenchymal patterns up into the range where the model works. Next comes a Gaussian smoothing with σ = 1.0,

$G(x,y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right), I_s = I_c * G$ (3)

which suppresses isolated high-frequency fluctuations (very fine structures may get a bit weaker, but the same operation goes to all models). Last, every image is brought to 224×224 by bilinear interpolation, scaled by 1/255 and standardized with the training-part mean and standard deviation,

$I_n = \frac{I\_s/255 - \mu}{\sigma + 10^{-5}}$ (4)

with μ = 0.5250, σ = 0.2556 in Experiment A and μ = 0.5327, σ = 0.2527 in Experiment B, so the fixed numerical contract for every model is a tensor in $\mathbb{R}1\times224\times224$.

C. Training augmentation, balanced sampling and batching

Random transformations are applied to the training batches only: a rotation inside ±15°, a horizontal flip and a random resized crop with scale in [0.8, 1.0] and aspect ratio in [0.9, 1.1]; validation and test tensors get only the deterministic steps of Section 3.2. For the 5:1 imbalance of Experiment A, the training loader picks each example j with a probability that goes with the inverse frequency of its class,

$w_j = \frac{1}{n_{y\_j}}$ (5)

where ny_j counts the training images of class y_j, so that in expectation the TB examples show up as often as the normal ones in the optimized batches; Experiment B runs with the same sampler, but there the weights are almost equal. Batches of 16 are used everywhere, which gives 209 training and 27 validation/test batches in Experiment A, and 306 and 66 in Experiment B.

D. Baseline architectures

Vision Transformer

The system we compare against is the transformer-based TB detector from the base paper [3] , which we faithfully rebuilt and call PaperViT. Four convolutional stem stages (stride-2 convolutions, batch normalization, GELU) bring the single-channel 224×224 input to a 14×14×384 grid, the channels growing 48 → 96 → 192 → 384; the grid is flattened to 196 tokens, a class token is put at the front, a positional-encoding generator puts back the location information, and six transformer encoder blocks with

eight heads and an MLP ratio of 4.0 are followed by an MLP head that gives one TB logit. Given a series of tokens $X \in \mathbb{R}^{N\times d}$, the attention calculation is made using all attention heads:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^{\top}}{\sqrt{d_k}}\right)V \quad (6)$$

The calculation is based on the matrix product of three linearly transformed versions of X: $Q = XW_Q$, $K = XW_K$, $V = XW_V$. d_k is the dimension of each head. The heads are then combined, and the final result goes through a new projection. Each block is equipped with a residual connection and a layer normalization as well as a position-wise multi-layer perceptron:

$$\text{MSA}(X) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)\, W_O \quad (7)$$

$$X' = X + \text{MSA}(\text{LN}(X)),\ X'' = X' + \text{MLP}(\text{LN}(X')) \quad (8)$$

For the model described by equation (8), a pre-normalization technique was used. The whole operation was performed six times. One of the drawbacks of that approach is that the computation and memory requirements increase with the square of the number of tokens in the input, N. The basic version that had been implemented has 13.527 million trainable parameters.

ConvNeXtV2-Pico and EfficientNet-B3 (CNN comparators)

Hence, to ensure that the evaluation of the transformer-only model will be fair, they also included two state-of-the-art convolutional networks. ConvNeXtV2-Pico [24] has four stages of depth [2, 2, 6, 2] and widths [64, 128, 256, 512], each block having a 7×7 depthwise convolution, GELU pointwise projections and global response normalization, and with our binary wrapper it comes to 9.328 million parameters; EfficientNet-B3 [23] comes from compound scaling and is built from mobile inverted bottleneck (MBConv) blocks with an SE gate, followed by a convolutional head, pooling, dropout and a one-output classifier, 10.697 million parameters in all. Both comparators were changed to accept a single-channel input, so the comparison is strictly fair on the input side.

Where our approach was better than the previous works: a transformer is a big model (13.527 million parameters). In fact, the quadratic attention in these transformers scales very poorly with resolution; a fixed-resolution token sequence gives a low-contrast image and wide asymmetry of the lung fields shares the bandwidth, and sensitivity is compromised by the imbalance, four of the seventy TB films are missed in Experiment A.

E. Proposed MSVMamba-TB network

A state-space model describes how a latent state h changes as a sequence is consumed; in continuous time the system reads

$$h'(t) = A\,h(t) + B\,x(t),\ y(t) = C\,h(t) \quad (9)$$

in which $x(t)$ is the input, $h(t)$ the hidden state and $y(t)$ the output, and A, B, C are the state, input and output matrices. Equation (9) is then discretized by the zero order hold with a step Δ as

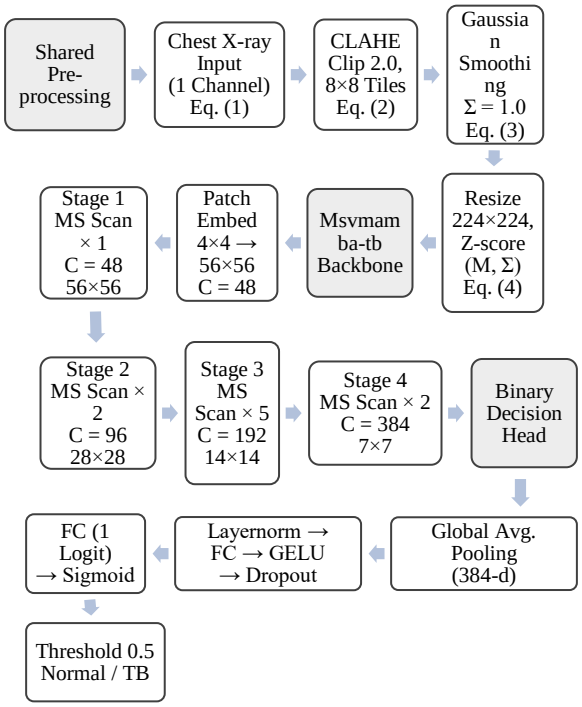
$$h_t = \overline{A}\,h_{t-1} + \overline{B}\,x_t,\ y_t = C\,h_t,\ \overline{A} = \exp(\Delta A),\ \overline{B} = (\Delta A)^{-1}(\exp(\Delta A) - I)\,\Delta B \quad (10)$$

The idea of Mamba [6] is to let (B, C, Δ) depend on the input itself, so that the transition becomes content-adaptive: the network works out for itself which radiographic features go into the latent state, which persist across positions and which go to the output, and since the recurrence is evaluated by a parallel scan the cost grows about linearly with the sequence length. For images, the SS2D cross-scan of VMamba [8] opens the feature map out into four directed sequences (rows, columns and the reverse of each), scans every one and merges what comes back, which takes out the bias of a single traversal direction.

Figure 1 can be read as the block diagram of our approach from shared pre-processing down to decision head; the pipeline was designed to have the same input processing across architectures, an architecture that integrated local textures and wide spatial context, a parameter budget smaller than what PaperViT has, and reproducibility from saved configuration and fixed split manifests. The radiograph normalized 1×224×224 is first patch-embedded with 4×4 gaps, producing a feature grid of size 56×56, then passes through four stages with channel width C = [48, 96, 192, 384] and depth [1, 2, 5, 2], let the grid transition from 56×56 to 28×28 to 14×14 to 7×7 across; in each stage the current convolution-only or attention-only block would be replaced with the partial state-space computation; after the fourth stage we average-pool the map into a 384-vector, add a layer normalizes, one fully-connected layer, a GELU, a dropout, and a final binary projector to derive the TB logit; the entire pipeline contains about 7.012M trainable parameters

("multiscale_4scan_12" variant, with ConvFFN and squeeze-and-excitation_enabled).

Fig. 1. Block diagram of the proposed method: shared pre-processing (top row), the four-stage MSVMamba-TB backbone with its multi-scale (MS) selective-scan stages (middle row) and the binary decision head (bottom



F. Multi-scale selective scan block

Figure 2 opens up what goes on inside every "MS scan" stage of Figure 1. From the stage tensor $F \in \mathbb{R}^{B\times C\times H\times W}$ two paths are made, one projection at the original scale and one at a coarse scale,

$$F_o = \varphi_o(F),\ F_d = \varphi_d\big(\text{Downsample}(F)\big) \quad (11)$$

The original-scale path is scanned along four directions, which keeps the small structures and the directional continuity, whereas in the coarse path F is first downsampled, so a single state transition covers a larger receptive field:

$$S_o = \text{SS2D}(F_o),\ S_d = \text{Upsample}\big(\text{SS2D}(F_d)\big) \quad (12)$$

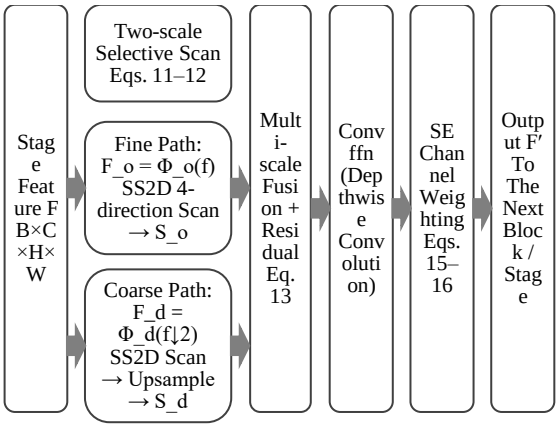
$$F' = \mathcal{F}(S_o, S_d) \quad (13)$$

in which φ_o and φ_d are learned projections, SS2D the directional selective scan and $\mathcal{F}$ the multi-scale fusion with a residual connection. Inside every directional sequence, the discretized update reads

$$h_t = \overline{A}_t\, h_{t-1} + \overline{B}_t\, x_t,\ y_t = C_t\, h_t \quad (14)$$

with the input-dependent ($\bar{A}_t$, $\bar{B}_t$, C_t) derived from x_t. With this hierarchy-in-hierarchy arrangement the local convolution, the global state propagation and the multi-scale fusion all work inside one stage, so a tiny nodule at the lung periphery and a big bilateral infiltrate each get represented at the scale that is natural for them.

Fig. 2. Flow chart of one MSVMamba-TB stage: fine and coarse selective-scan paths, their fusion with a residual connection, and the ConvFFN and squeeze-and-excitation refinement.



G. ConvFFN refinement & SE channel weighting

After the fusion, the representation goes through a convolutional feed-forward network (ConvFFN) with a depthwise spatial convolution inside, which holds on to the local spatial structure that pure token mixers tend to wash out, and then a squeeze-and-excitation (SE) module takes care of the channel attention. If $Z \in \mathbb{R}^{B\times C\times H\times W}$ is the stage tensor, a global average pooling first gives one descriptor per channel,

$$g_c = \frac{1}{HW}\sum_{u=1}^{H}\sum_{v=1}^{W} Z_c(u, v) \quad (15)$$

and two learned projections give the normalized channel weights

$$a = \sigma\big(W_b\, \delta(W_a\, g)\big),\ \tilde{Z}_c = a_c\, Z_c \quad (16)$$

where δ is ReLU, σ the sigmoid and $a_c \in (0, 1)$ the learned importance of channel c: a channel that reacts strongly to a useful pattern gets a bigger weight and a weak or redundant one is pushed down, so the selective scan controls the information across positions and the SE across channels, without any all-to-all attention matrix.

H. Classification head

In the head, the last 7×7×384 tensor is average-pooled globally, a layer normalization follows, one fully connected layer widens to a hidden size and goes

through GELU and dropout, and a last linear layer gives a single output; the sigmoid of this logit is the TB probability, and a film is referred as TB-positive when it is 0.5 or more.

I. Common training protocol and experimental workflow

For all the models, the loss is the binary cross-entropy computed on the logits,

$$\mathcal{L}_{\mathrm{BCE}} = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log\sigma(\ell_i) + (1 - y_i)\log\left(1 - \sigma(\ell_i)\right)\right] \quad (17)$$

and the optimizer is AdamW with decoupled weight decay, a starting learning rate of 10^{-4}, weight decay 5×10^{-4} , and the gradient norm clipped to 1.0; over the epochs the learning rate follows a cosine schedule, so at epoch t it is

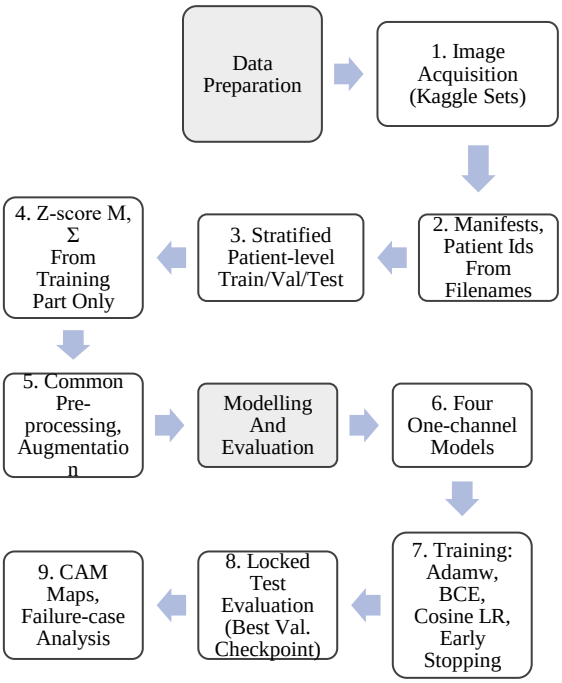
$$\eta_t = \eta_{\min} + \frac{1}{2}(\eta_{\max} - \eta_{\min})\left[1 + \cos\left(\frac{\pi t}{T}\right)\right] \quad (18)$$

A maximum of 100 epochs are permitted, but training terminates if there has been no decrease in the validation loss over the previous 10 epochs, mixed precision autocasting is enabled; checkpoint selection is based on the lowest validation loss, and the decision threshold is maintained permanently at 0.5 for all subjects. Table 2 collects every setting shared by the four models, and Figure 3 lays out the whole experimental workflow in nine fixed steps; because the z-score statistics, the augmentation and the checkpoint choice all sit before the locked evaluation, no test information can leak into preprocessing, checkpoint selection or optimization.

Fig. 3. Flow chart of the experimental workflow applied identically in Experiments A and B.

Table 2. Hyperparameters and settings shared by all four networks.

Setting	Value	Setting	Value
Input size / channels	224 × 224 / 1	Loss	BCE with logits
Batch size	16	Optimizer	AdamW
Maximum epochs	100	Learning rate	10−4
Early-stop patience	10 epochs	Weight decay	5 × 10−4
Scheduler	Cosine annealing	Gradient clipping	1.0



Decision threshold	0.5	Mixed precision	Enabled on CUDA
Random seed	42	CLAHE clip / grid	2.0 / 8 × 8
Gaussian σ	1.0	Rotation	±15°
Crop scale / ratio	[0.8, 1.0] / [0.9, 1.1]	Horizontal flip	Yes (training only)
Drop-path (MSVMamba-TB)	0.2	Dropout (PaperViT head)	0.1

J. Evaluation metrics

All the evaluation starts from the four counts of the confusion matrix, TN, FP, FN and TP, with TB as the positive class. Accuracy is the share of all test films classified correctly,

$$\text{Accuracy} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \quad (19)$$

but since accuracy can look high when one class dominates, the class-specific measures go along with it: precision tells how many of the TB predictions are really TB,

$$\text{Precision} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \quad (20)$$

sensitivity (recall) tells what share of the actual TB films were caught, the screening-critical one, because behind every false negative there may be an infectious patient who was not flagged,

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (21)$$

specificity tells what share of the normal films were correctly rejected, which limits the false referrals,

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \quad (22)$$

the F1-score is the harmonic mean of precision and sensitivity,

$$F_1 = \frac{2 \cdot \text{Precision} \times \text{Sensitivity}}{\text{Precision} + \text{Sensitivity}} \quad (23)$$

and the AUC, the area below the receiver-operating-characteristic curve, tells how well the output probabilities put the TB films above the normal ones without depending on any threshold,

$$\text{AUC} = \int_0^1 \text{TPR}(u)\, du \quad (24)$$

where u is the false-positive rate; an AUC close to 1 means strong separation and 0.5 is chance level. Every metric is computed once on the locked test part of each experiment.

K. Experimental environment and reproducibility

Both experiments were run in Kaggle GPU notebook sessions on a pair of NVIDIA Tesla T4 cards (16 GB each), with Python 3.12, PyTorch 2.4.0 and CUDA 12.1, the selective scan running through its dedicated CUDA kernel (mamba-ssm); one configuration file fixes the random seed (42), the dataset, the split ratios and every preprocessing parameter and hyperparameter, and the evaluation stage reloads each best checkpoint and regenerates every number and figure reported below.

IV. RESULTS AND DISCUSSION

A. Experiment A: training and validation behaviour

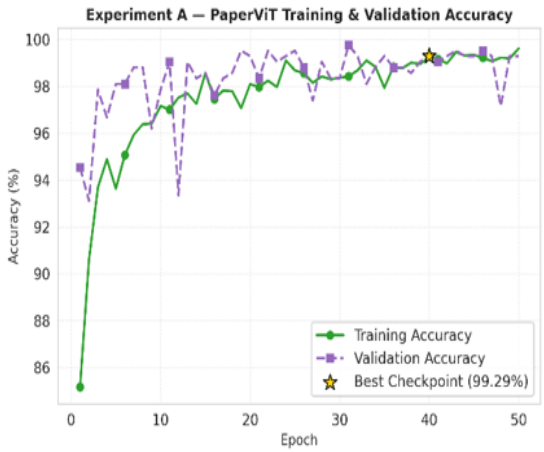
For every model the history of every completed epoch was saved; Table 3 gives the epoch where each run stopped, the best epoch and the validation figures at that point, and Figures 4 and 5 give the accuracy and the loss curves, with the best checkpoint marked with a star. PaperViT went on for 50 epochs with its lowest validation loss, 0.0120, at epoch 40; MSVMamba-TB was the quickest to converge and stopped after 35 epochs with 0.0454 at epoch 25; ConvNeXtV2-Pico went for 48 epochs with 0.0256 at epoch 38, and EfficientNet-B3 was done after 22 epochs with 0.0349 at epoch 12, the different stopping points coming only from the patience rule.

Table 3. Experiment A: course of training and the checkpoint that was kept (lowest validation loss).

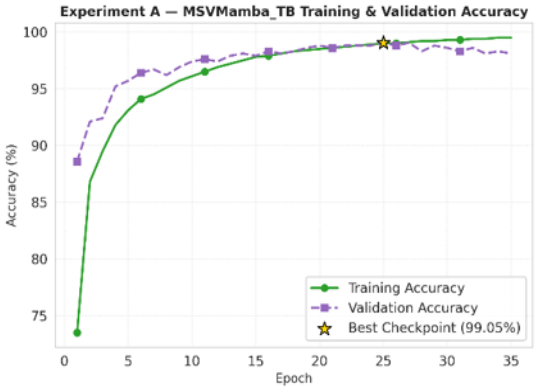
Model	Epochs run	Best epoch	Min. validation loss	Validation accuracy at best
PaperViT	50	40	0.0120	99.29%
MSVMamba-TB	35	25	0.0454	99.05%
ConvNeXtV2-Pico	48	38	0.0256	99.29%
EfficientNet-B3	22	12	0.0349	99.05%

In Figure 4 the training accuracy of all four models climbs steeply in the first few epochs and then flattens above 98%, with the validation accuracy sitting close to it, so the generalization gap stays small for everybody; the two CNNs and the proposed model give smooth curves, whereas PaperViT is visibly more jumpy, its validation accuracy swinging between roughly 93% and 99.8% before it settles in the last ten or fifteen epochs.

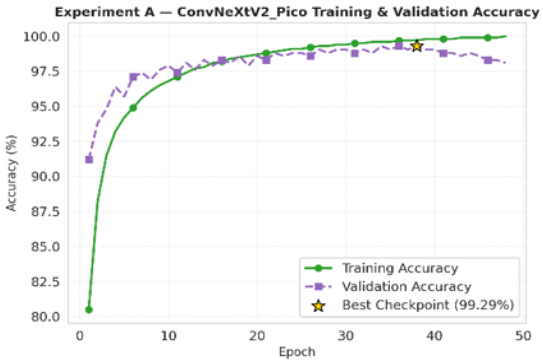
Fig. 4. Experiment A: training and validation accuracy of (a) PaperViT, (b) MSVMamba-TB, (c) ConvNeXtV2-Pico and (d) EfficientNet-B3; the star marks the best checkpoint (minimum validation loss).



(a)



(b)

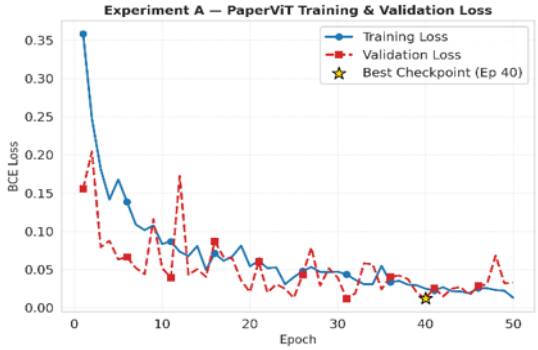


(c)

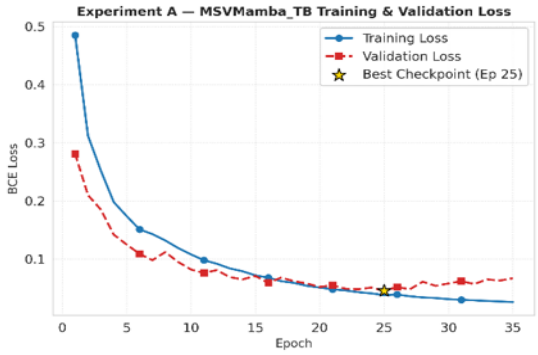
(d)

In Figure 5 the training loss goes down monotonically for all of them, the validation loss of the proposed model and of the two CNNs drifts slightly upward after the starred epoch, exactly where early stopping takes over, and the PaperViT validation loss has several spikes (near epochs 2, 9, 12 and 48).

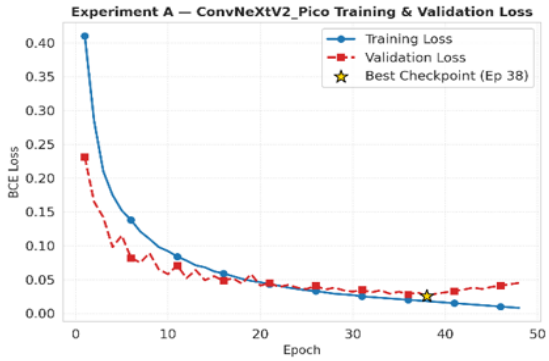
Fig. 5. Experiment A: training and validation binary cross-entropy loss of (a) PaperViT, (b) MSVMamba-TB, (c) ConvNeXtV2-Pico and (d) EfficientNet-B3; the star marks the best-checkpoint epoch.



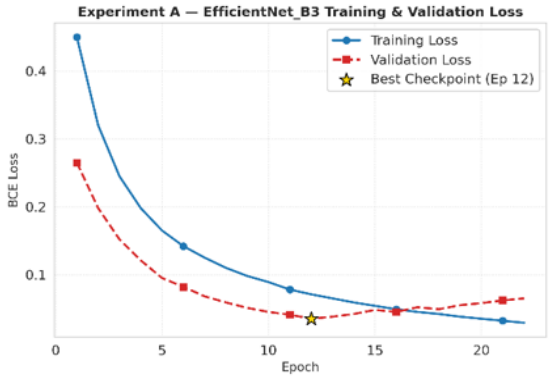
(a)



(b)



(c)



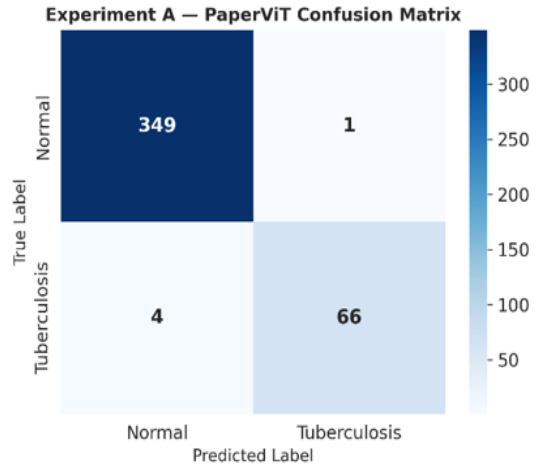
(d)

B. Experiment A: test performance, confusion matrices and ROC analysis

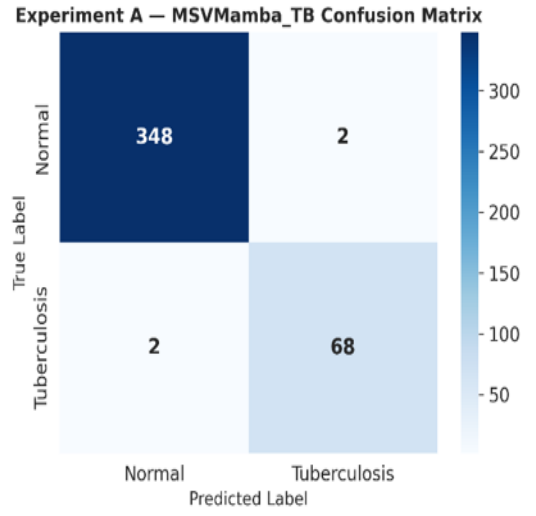
The Experiment A test set has 420 images (350 normal, 70 TB). Table 4 gives the locked test metrics with the counts; Figure 6 shows the confusion matrices, and Figure 7 depicts the ROC curves. For MSVMamba-TB, 416 was obtained, an accuracy of 99.05%, with four errors (two false positives and two false negatives); thus, 97.14% sensitivity, 97.14% precision, and 99.43% specificity. PaperViT and ConvNeXtV2-Pico each got 415 questions right but in different ways: PaperViT missed the four from the total of the seventy TB films (sensitivity was 94.29% as it detected) with one false positive.PaperViT and ConvNeXtV2-Pico matched the sensitivity of the proposed model but gave three false positives; PaperViT EfficientNet-B3 got the other 413 questions correct but was least sensitive, as it missed six out of the seventy films (91.43%).

Table 4. Experiment A: The metrics of the locked test done with the 420-image partition (errors = FP + FN).

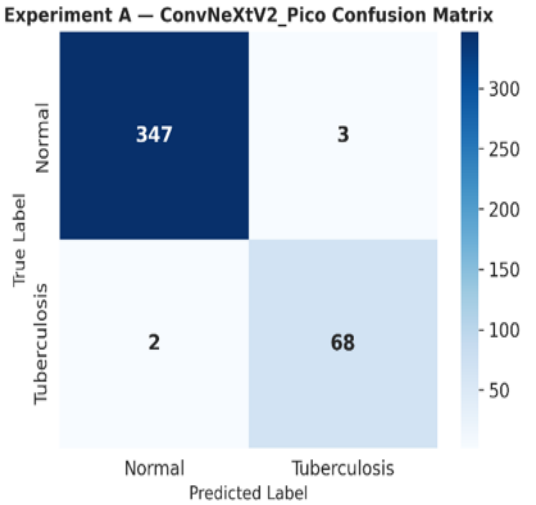
Model	Acc.	Prec.	Sens.	Spec.	F1	AUC	TN	FP	FN	TP	Errors
PaperViT	0.98811	0.98511	0.94291	0.99715	0.96635	0.99993	349	1	4	66	5
MSVMamba-TB	0.99057	0.97144	0.97143	0.99434	0.97144	0.99889	348	2	2	68	4
ConvNeXtV2-Pico	0.98811	0.95777	0.97144	0.99144	0.96455	0.99926	347	3	2	68	5
EfficientNet-B3	0.98333	0.98466	0.91433	0.99715	0.94811	0.99906	349	1	6	64	7



(a)

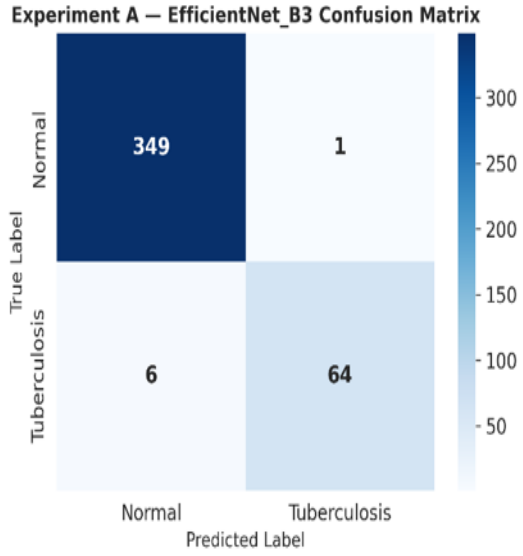


(b)



(c)

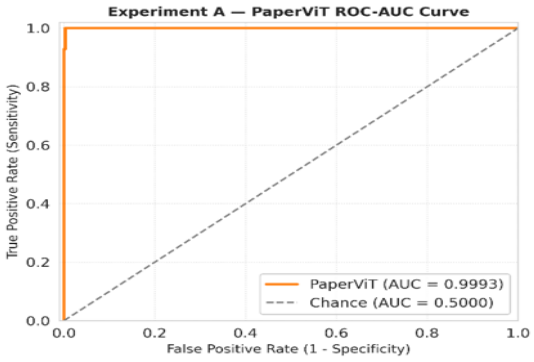
Fig. 6. Experiment A: test-set confusion matrices (420 images) of (a) PaperViT, (b) MSVMamba-TB, (c) ConvNeXtV2-Pico and (d) EfficientNet-B3.



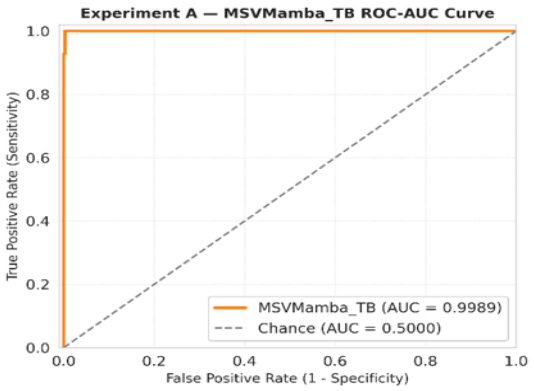
(d)

The numbers are tightly packed because all four models are doing well, so the real distinction is in the sensitivity and in how the errors are split. The proposed model has the fewest errors overall of total with four, with the most balanced sensitivity and precision; hence, its F1-score of 0.9714 is the best among all. In a screening reading, every false negative represents an infectious patient possibly going home unflagged and every false positive represents one confirmatory test, which makes the proposed model keeps the combined burden at the least. In Figure 6, the PaperViT errors sit in the missed-TB cell (4 FN / 1 FP), the ConvNeXtV2-Pico errors lean toward unnecessary referrals (3 FP / 2 FN), and EfficientNet-B3 puts nearly all of its errors into the false-negative cell. In Figure 7 every ROC curve is hugging the top-left corner, which fits with the AUC column of Table 4 (0.9993 for PaperViT, 0.9989 for MSVMamba-TB, 0.9926 for ConvNeXtV2-Pico and 0.9906 for EfficientNet-B3): PaperViT keeps the numerically highest AUC here with the proposed model just behind it, and the differences at the fixed 0.5 threshold have to be read from Table.4 and Figure.6.

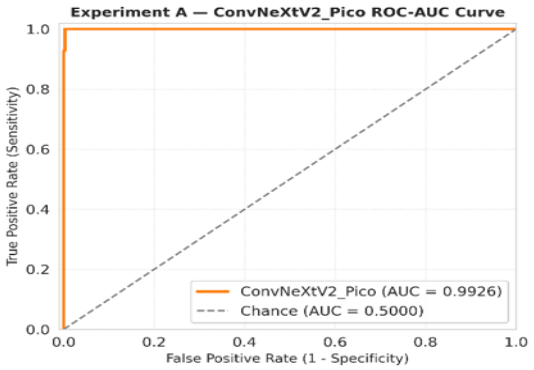
Fig. 7. ROC curves obtained on the 420-image test partition for the four models, namely (a) PaperViT, (b) MSVMamba-TB, (c) ConvNeXtV2-Pico, and (d) EfficientNet-B3, along with the chance diagonal for comparison.



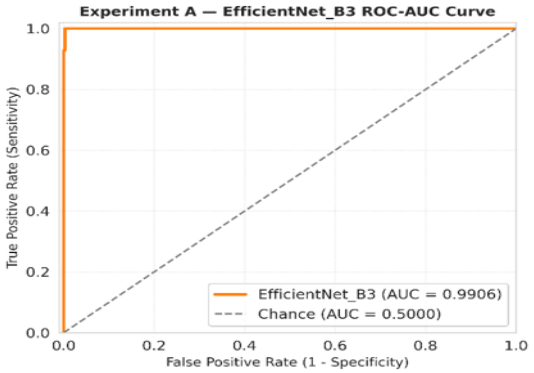
(a)



(b)



(c)

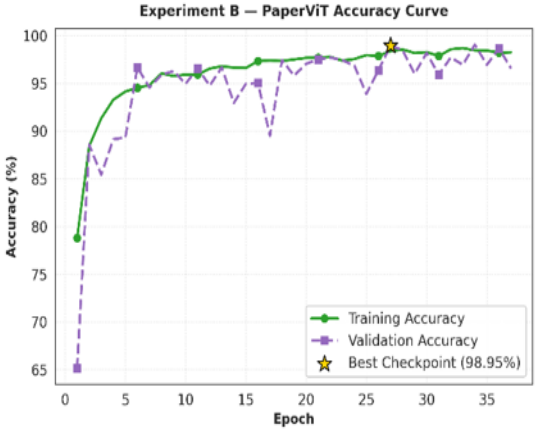


(d)

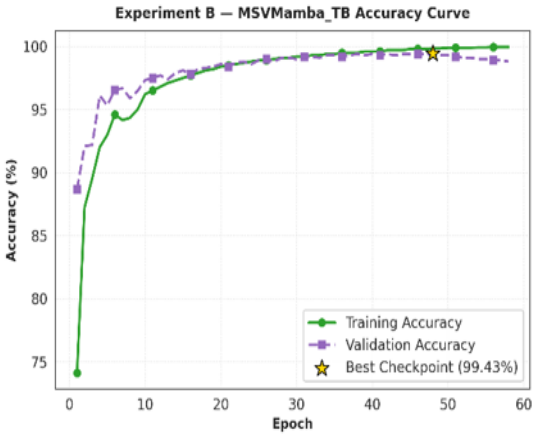
C. Experiment B: Training and Validation Behaviour

In Experiment B, the training corpus is roughly doubled compared with the earlier experiment, while at the same time the classes are made equal. The main checkpoint information is given in Table 5, and the complete training histories can be seen in Figures 8 and 9. MSVMamba-TB was further trained 58 times and produced the best model at epoch 48, where validation error was 0.0225 and accuracy was 99.43%. PaperViTT got completed at epoch 37, whereas its maximum performing one turned out to be at epoch 27, where the validation error was 0.0293. ConvNeXtV2-Pico continued training until 65 epochs, and its best checkpoint was at epoch 55 with a validation loss of 0.0267. EfficientNet-B3 was stopped at epoch 31, and its best-performing checkpoint was at epoch 21 with a validation loss of 0.0298.

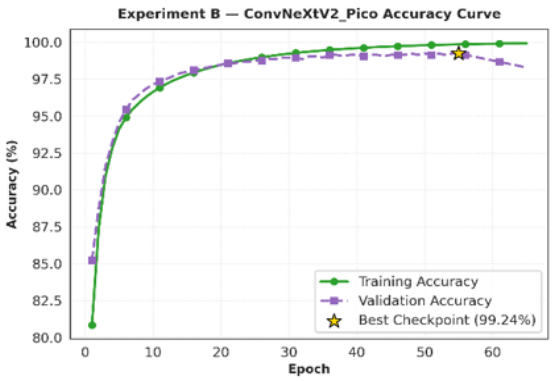
Fig. 8. Experiment B: training and validation accuracy of (a) PaperViT, (b) MSVMamba-TB, (c) ConvNeXtV2-Pico and (d) EfficientNet-B3; the star marks the best checkpoint.



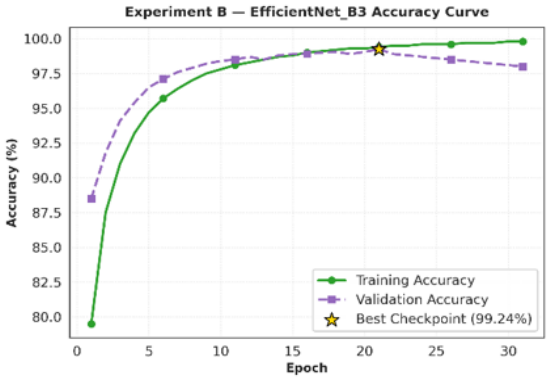
(a)



(b)



(c)



(d)

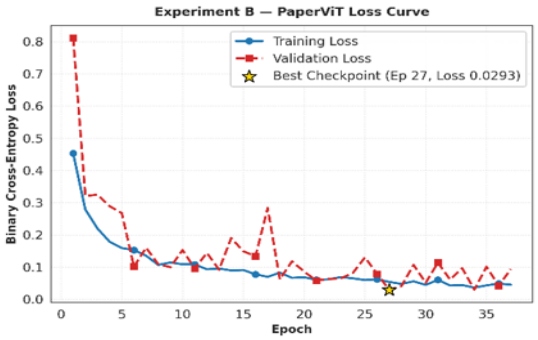
Table 5. Experiment B: course of training and the checkpoint that was kept (lowest validation loss).

Model	Epochs run	Best epoch	Min. validation loss	Validation accuracy at best
PaperViT	37	27	0.0293	98.95%
MSVMamba-TB	58	48	0.0225	99.43%
ConvNeXtV2-Pico	65	55	0.0267	99.24%
EfficientNet-B3	31	21	0.0298	99.24%

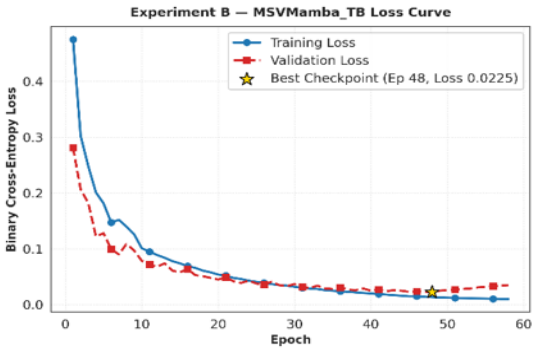
In Figure 8, the accuracy curves of MSVMamba-TB and ConvNeXtV2-Pico are almost lying on top of each other for training and validation throughout the whole run, a nice sign that the bigger balanced corpus is reducing overfitting, while EfficientNet-B3 shows its validation accuracy starting to slide after about epoch 21, when its checkpoint was taken, and PaperViT again is the noisy one, starting near 65% and dropping to roughly 90% around epoch 17 before recovering.

Fig. 9. Experiment B: training and validation binary cross-entropy loss of (a) PaperViT, (b) MSVMamba-

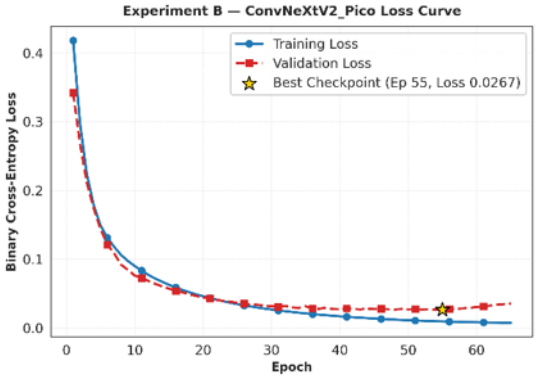
TB, (c) ConvNeXtV2-Pico and (d) EfficientNet-B3; the star marks the best-checkpoint epoch.



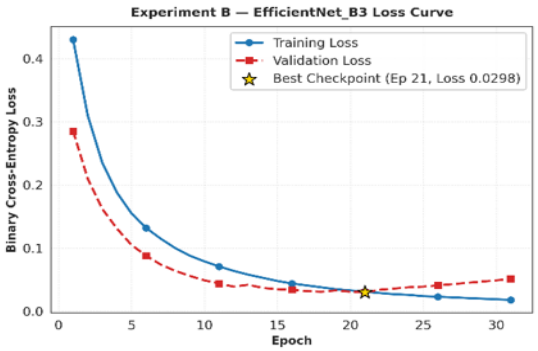
(a)



(b)



(c)



(d)

In Figure 9, the validation loss of the proposed model and the CNNs follows the training loss down to a floor near 0.02–0.03 and then turns up slowly, whereas the PaperViT validation loss begins above 0.8 and keeps oscillating until the end.

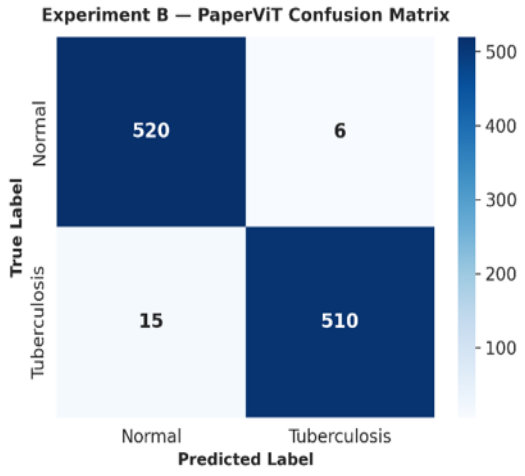
D. Experiment B: test performance, confusion matrices and ROC analysis

The Experiment B test set has 1,051 images (526 normal, 525 TB), so the positive support is 7.5 times what it was in Experiment A; Table 6 gives the locked results, Figure 10 the confusion matrices and Figure 11 the ROC curves. MSVMamba-TB comes to an accuracy of 98.38% and keeps the class-specific numbers in the best balance: precision 99.42% (three false positives only), sensitivity 97.33% (fourteen false negatives), specificity 99.43%, F1-score 98.36% and AUC 0.9992. Next is PaperViT with 98.00% accuracy (6 false positives, 15 false negatives); EfficientNet-B3 is the best on precision (99.80%) and specificity (99.81%) but the worst on sensitivity (96.38%); and ConvNeXtV2-Pico matches the sensitivity of the proposed model but makes five more false positives (97.91% accuracy).

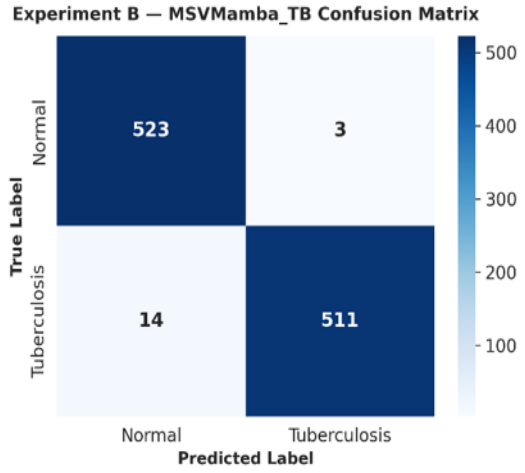
Table 6. Experiment B: locked test metrics, on the 1,051-image partition (errors = FP + FN).

Model	Acc.	Prec.	Sens.	Spec.	F1	AUC	TN	FP	FN	TP	Errors
PaperViT	0.9800	0.9884	0.9714	0.9886	0.9798	0.9975	520	6	15	510	21
MSVMamba-TB	0.9838	0.9942	0.9733	0.9943	0.9836	0.9992	523	3	14	511	17
ConvNeXtV2-Pico	0.9791	0.9846	0.9733	0.9848	0.9789	0.9984	518	8	14	511	22
EfficientNet-B3	0.9810	0.9980	0.9638	0.9981	0.9806	0.9975	525	1	19	506	20

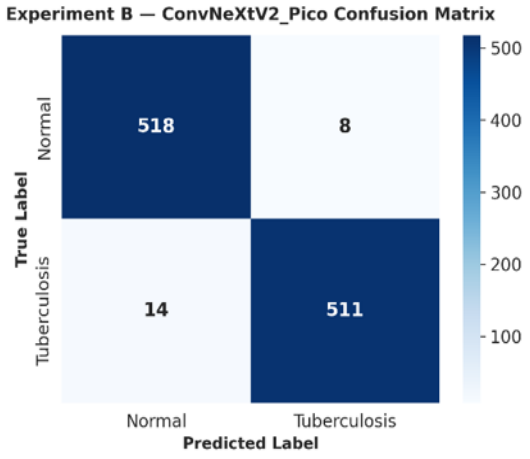
Fig. 10. Experiment B: test-set confusion matrices (1,051 images) of (a) PaperViT, (b) MSVMamba-TB, (c) ConvNeXtV2-Pico and (d) EfficientNet-B3.



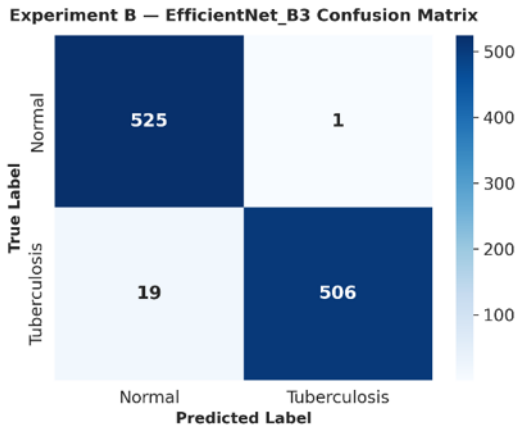
(a)



(b)

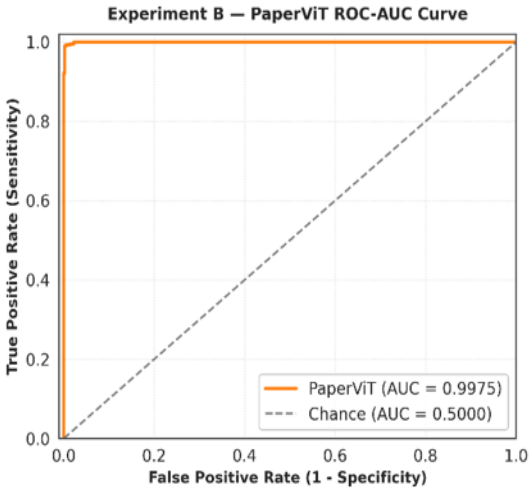


(c)

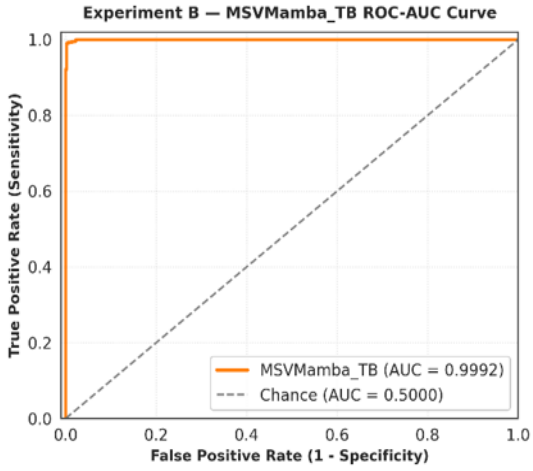


(d)

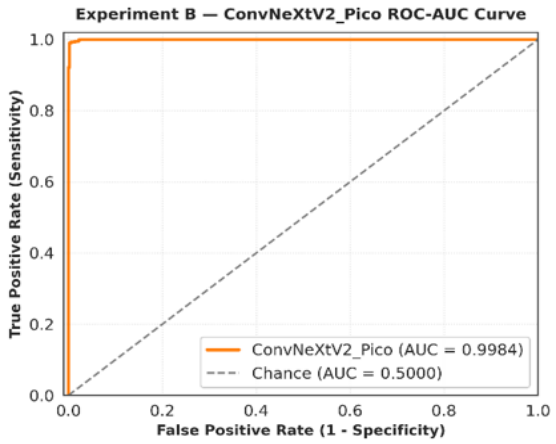
Fig. 11. Experiment B: ROC curves on the 1,051-image test partition of (a) PaperViT, (b) MSVMamba-TB, (c) ConvNeXtV2-Pico and (d) EfficientNet-B3, with the chance diagonal.



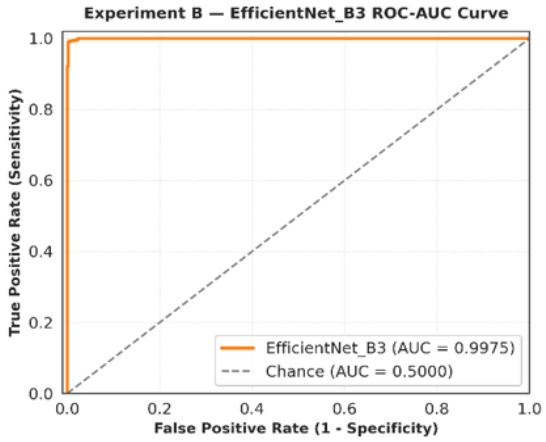
(a)



(b)



(c)



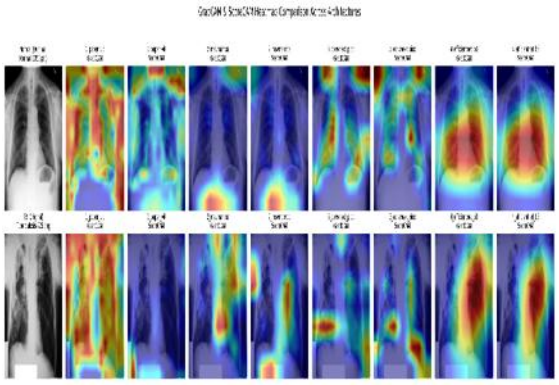
(d)

E. Qualitative observations: activation maps

Apart from the numbers, we also looked at the trained models with Grad-CAM [27], where every channel of the feature map gets a weight equal to the spatially averaged gradient of the class score, and Score-CAM [28], which avoids the gradients and measures the change in the score produced by activation-derived masks; warm colours point to the regions which contribute more to the output. Figure 12 puts the two methods next to each other for one normal test radiograph (Normal-1359) and one TB radiograph (Tuberculosis-225) of Experiment A, for all four models. PaperViT gives a broad activation that stretches over both lungs and up to the image border, so the transformer is not really pointing at one place.

In our proposed model, the maps are much tighter: on the TB film, both Grad-CAM and Score-CAM put the warm region on the same para-hilar and mid-lung zone on the right-hand side of the image, where the film shows coarse shadowing, and on the normal film the only warm response sits low, under the diaphragm and away from the lung fields. ConvNeXtV2-Pico gives a few localized but scattered peaks, and EfficientNet-B3 gives one broad blob over the thorax in both films, TB or not, more a fixed template than a case-specific explanation; a radiologist's assessment is still to be done, but the compact activation of the proposed model goes together with the multi-scale design, where the coarse path gives the context and the fine path keeps the position of the abnormality.

Fig. 12. Grad-CAM and Score-CAM maps for one normal (Normal-1359, top row) and one TB test radiograph (Tuberculosis-225, bottom row) of Experiment A; first column the original film, then the Grad-CAM / Score-CAM pairs of PaperViT (1_paper_vit), MSVMamba-TB (2_msvmamba), ConvNeXtV2-Pico (3_convnext_pico) and EfficientNet-B3 (4_efficientnet_b3). Warm colors mark the regions with the larger contribution to the output.



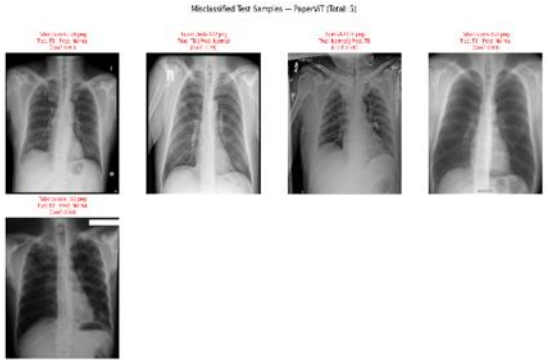
F. Failure-case analysis

We went through every wrongly classified test film one by one; Figure 13 shows the 80 errors of Experiment B and Figure 14 shows the 21 errors of Experiment A, one panel per model, each film labelled with its name, its true and predicted class and the TB probability (Conf.) the model gave it, and Table 8 lists every one of these films with its probability. In Experiment B, the errors of all four models are dominated by TB films called normal (15 of 21 for PaperViT, 14 of 17 for MSVMamba-TB, 14 of 22 for ConvNeXtV2-Pico and 19 of 20 for EfficientNet-B3), so the models differ mostly in how many normal films they over-call. Of the 48 films involved, 28 are missed by only one model, 12 by two, 4 by three and 4 by all four (Tuberculosis-120, -1485, -2241 and -3344), and those four get a very low TB probability from almost

every model, so at 224×224 the input resolution rather than the backbone is the limit there; the three false positives of MSVMamba-TB are all close to the threshold (0.54–0.65).

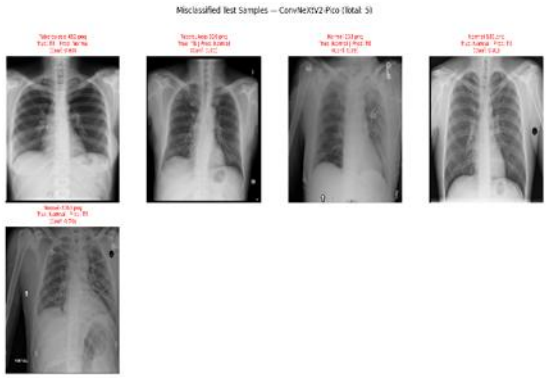
Experiment A repeats the pattern at a smaller scale: its 21 errors fall on 13 films, eight missed by one model only, three by two, Tuberculosis-492 by three and Tuberculosis-120 by all four. None of the four errors of the proposed model is its own, and its two false positives, Normal-233 and Normal-3063, like the only false positives of PaperViT (Normal-2014) and EfficientNet-B3 (Normal-1913), are portable anteroposterior films with side markers, arrows and monitoring lines over the thorax, so we suspect the acquisition style itself is being read as a sign of disease. Tuberculosis-192, -122, -173 and -492 are near-threshold misses (0.37–0.50) that a validation-based operating point could recover, whereas Tuberculosis-120 (at most 0.02 for all) is a confident mistake of every model; since most hard films are missed by just one model, ensemble or reader-plus-model workflows are a natural next step.

Fig. 13. Experiment A: every misclassified test film of (a) PaperViT (5), (b) MSVMamba-TB (4), (c) ConvNeXtV2-Pico (5) and (d) EfficientNet-B3 (7), labelled as in Figure 13

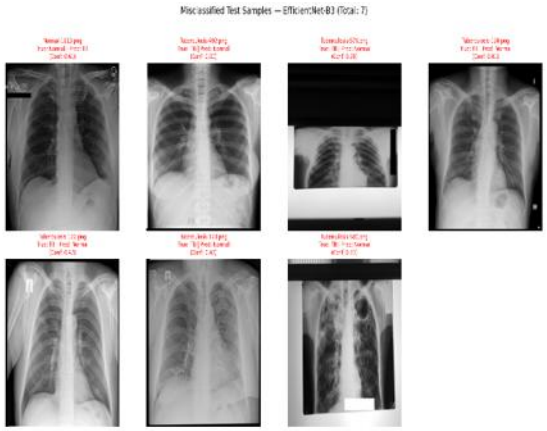


(a)

(b)



(c)



(d)

Table 7. Every misclassified test film of both experiments with the TB probability given by the model (TB-n = Tuberculosis-n.png, a TB film called normal; N-n = Normal-n.png, a normal film called TB).

Experiment / model (errors)	Misclassified films (TB probability given by the model)
A / PaperViT (5)	TB-120 (0.01), TB-122 (0.24), N-2014 (0.98), TB-624 (0.03), TB-192 (0.50)
A / MSVMamba-TB (4)	TB-492 (0.37), TB-120 (0.02), N-233 (1.00), N-3063 (0.98)
A / ConvNeXtV2-Pico (5)	TB-492 (0.00), TB-120 (0.00), N-233 (0.99), N-186 (0.81), N-3063 (0.70)
A / EfficientNet-B3 (7)	N-1913 (0.61), TB-492 (0.00), TB-579 (0.20), TB-120 (0.01), TB-122

	(0.47), TB-173 (0.42), TB-580 (0.33)
B / PaperViT (21)	TB-820 (0.10), TB-624 (0.12), N-3084 (0.91), TB-120 (0.01), TB-3468 (0.49), TB-463 (0.11), TB-895 (0.06), TB-1781 (0.34), N-2048 (0.51), N-1012 (0.63), TB-1136 (0.50), TB-3344 (0.10), TB-2241 (0.01), N-3063 (0.58),TB-1519 (0.39), TB-1362 (0.03), TB-1485 (0.33), TB-1495 (0.01), N-1788 (0.84), N-2014 (0.98), TB-1800 (0.00)
B / MSVMamba-TB (17)	TB-820 (0.49), TB-2274 (0.30), TB-2844 (0.32), TB-120 (0.01), TB-753 (0.06), TB-1438 (0.11), TB-747 (0.41),TB-3344 (0.01), TB-1356 (0.12), TB-2241 (0.00), N-2759 (0.55), TB-650 (0.41), TB-1362 (0.10), N-233 (0.65),TB-1485 (0.01), TB-1495 (0.02), N-1788 (0.54)
B / ConvNeXtV2-Pico (22)	TB-1715 (0.09), TB-2844 (0.00), TB-120 (0.06), TB-753 (0.05), N-2048 (0.81), TB-1409 (0.03), N-1012 (0.89),TB-3344 (0.08), TB-1244 (0.17), TB-948 (0.12), TB-2241 (0.15), N-2759 (0.79), TB-859 (0.40), N-23 (0.70), N-1913 (0.73), TB-650 (0.04), TB-3048 (0.13), N-233 (0.77),TB-1485 (0.00), N-2014 (0.60), TB-1201 (0.00), N-1561 (0.56)
B / EfficientNet-B3 (20)	TB-820 (0.00), TB-1715 (0.36), TB-2886 (0.48), TB-1167 (0.02), TB-120 (0.00), TB-753 (0.29), TB-2270 (0.11), N-2048 (0.56), TB-1136 (0.27), TB-3344 (0.48),TB-1244 (0.46), TB-1478 (0.00), TB-1356 (0.50), TB-2241 (0.12),TB-1485 (0.00), TB-1495 (0.23), TB-1484 (0.18), TB-1218 (0.05), TB-42 (0.07), TB-666 (0.02)

V CONCLUSION

In this work, we built MSVMamba-TB, a multi-scale Vision Mamba network of the hierarchy-in-hierarchy kind, for the two-class pulmonary TB decision on single-channel chest radiographs, and tested it against a ViT baseline (PaperViT) and two recent CNNs (ConvNeXtV2-Pico and EfficientNet-B3) under one strictly common protocol, on an imbalanced set of 4,200 images and on a balanced set of 7,000 images with patient-disjoint splits. In Experiment A, the proposed model got 99.05% accuracy, 97.14% sensitivity, 97.14% F1-score, and an AUC of 0.9989; in Experiment B, 98.38% accuracy, 99.42% precision, 97.33% sensitivity, 98.36% F1-score, and an AUC of 0.9992. It had the top accuracy and F1-score in both experiments and the fewest test errors, with about 7.012 million trainable parameters, 48.17% below PaperViT, so it gives the best trade-off between accuracy and compactness and a credible foundation for a deployable, explainable computer-aided TB triage tool.

VI FUTURE SCOPE

This paper mainly tests the method properly here, What we aim to do next is run MSVMamba-TB on datasets that are bigger and multi-hospital, with patient-level splits done properly and also duplicate-checking, so that a model doesn't end up getting evaluated on basically the same image twice over, and beyond just having bigger data, what we also want to do is throw in other lung conditions too, pneumonia,

lung cancer, fibrosis, this kind of thing, so that we actually get to know whether the model is picking out TB specifically as such, it is just doing the somewhat easier job of "normal versus not TB."

The other big piece of it is figuring out what exactly inside of the model is the thing that is actually doing the work. Right now, what happens is the whole network gets tested as one single block, so what is needed is component-level ablations, where we are pulling out, or maybe swapping, the coarse-scale branch, the scanning direction, the ConvFFN, and so on, one thing at a time, and then also comparing this against plain Vision Mamba and VMamba as well. Along with this, the numbers that are being reported need to be stress-tested a little bit more: different random seeds, confidence intervals, calibration and also precision-recall curves rather than only accuracy and AUC, and the decision threshold should be picked from the validation set for a screening target, instead of it just defaulting to 0.5 automatically.

The last part is more of a practical matter: how well this thing actually runs somewhere in reality, not only on paper. What this means is latency, memory, and energy numbers, and this on both GPUs and also the weaker devices, since a parameter count that is smaller only really matters if it is translating into something that is real. And then, only once everything has gotten locked down and checked independently, this is when radiologists come in to go over the Grad-CAM outputs and also the misclassified cases, and after that, followed by reader-assistance and clinical workflow studies, and this being done in that particular order, so the evaluation does not end up getting quietly reshaped after having already peeked at the test results.

VII. ACKNOWLEDGMENT

The authors would like to express their sincere gratitude to their guide, Prof. I. Kullayamma, for providing valuable guidance, continuous support, constructive suggestions, and encouragement throughout this research work. The authors also acknowledge the support provided by the Department of Electronics and Communication Engineering for offering the necessary academic facilities, resources, and research environment required to complete this study. The authors are thankful to all faculty members, colleagues, and others who contributed directly or indirectly to the successful completion of this work.

VIII. DATASET AVAILABILITY

The chest X-ray images used in this study were obtained from the publicly available Tuberculosis (TB) Chest X-ray Database [2],[25],[26]. The dataset includes normal and tuberculosis-positive chest radiograph images and was used for training, validation, and testing of the proposed MSVMamba-TB model. The dataset is available through the sources reported in and , subject to their respective access, licensing, and usage conditions. No private patient data were newly collected for this study.

REFERENCES

[1] Goletti, D., Meintjes, G., Andrade, B. B., Zumla, A., & Lee, S. S. (2025). Insights from the 2024 WHO global tuberculosis report–more comprehensive action, innovation, and investments required for achieving WHO end TB goals. *International Journal of Infectious Diseases*, *150*.

[2] Rahman, T., Khandakar, A., Kadir, M. A., Islam, K. R., Islam, K. F., Mazhar, R., ... & Chowdhury, M. E. (2020). Reliable tuberculosis detection using chest X-ray with deep learning, segmentation and visualization. *Ieee Access*, *8*, 191586-191601.

[3] Venkatachalam, S., Venkatachalam, C., & Shah, P. (2025). Tuberculosis detection in chest X-rays using vision transformers with convolutional stems and CLAHE-based contrast enhancement. *Biomedical Materials & Devices*, 1-16.

[4] Vanitha, K., Mahesh, T. R., Kumar, V. V., & Guluwadi, S. (2025). Enhanced tuberculosis detection using Vision Transformers and explainable AI with a Grad-CAM approach on chest X-rays. *BMC Medical Imaging*, *25*(1), 96.

[5] Dosovitskiy, A. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

[6] Gu, A., & Dao, T. (2023). Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*.

[7] Zhu, L., Liao, B., Zhang, Q., Wang, X., Liu, W., & Wang, X. (2024). Vision mamba: Efficient visual representation learning with bidirectional state space model. *arXiv preprint arXiv:2401.09417*.

[8] Liu, Y., Tian, Y., Zhao, Y., Yu, H., Xie, L., Wang, Y., ... & Liu, Y. (2024). Vmamba: Visual state space model. *Advances in neural information processing systems*, *37*, 103031-103063.

[9] Shi, Y., Dong, M., & Xu, C. (2024). Multi-scale vmamba: Hierarchy in hierarchy visual state space model. *Advances in Neural Information Processing Systems*, *37*, 25687-25708.

[10] Hedhoud, Y., Mekhaznia, T., & Amroune, M. (2025, April). Vision Mamba for efficient Tuberculosis Detection based on Chest X-Rays: A comparative study with CNN and Vision transformers. In *2025 7th International Conference on Pattern Analysis and Intelligent Systems (PAIS)* (pp. 1-6). IEEE.

[11] Kotei, E., & Thirunavukarasu, R. (2024). A Comprehensive Review on Advancement in Deep Learning Techniques for Automatic Detection of Tuberculosis from Chest X-ray Images: E. Kotei, R. Thirunavukarasu. *Archives of computational methods in engineering*, *31*(1), 455-474.

[12] Sharma, V., Nillmani, Gupta, S. K., & Shukla, K. K. (2024). Deep learning models for tuberculosis detection and infected region visualization in chest X-ray images. *Intelligent Medicine*, *4*(02), 104-113.

[13] Rahman, T., Khandakar, A., Rahman, A., Zughaier, S. M., Al Maslamani, M., Chowdhury, M. H., ... & Chowdhury, M. E. (2024). TB-CXRNet: Tuberculosis and drug-resistant tuberculosis detection technique using chest X-ray images. *Cognitive Computation*, *16*(3), 1393-1412.

[14] Das, P. K., Sreevatsav, S., & Abraham, A. (2024). An efficient deep learning network with orthogonal softmax layer for automatic detection of tuberculosis. *Engineering Applications of Artificial Intelligence*, *133*, 108116.

[15] Thirunavukarasu, R., Kotei, E., & Venkatesan, T. (2026). Explainability of Tuberculosis Diagnosis based on Chest X-Ray Images with Vision Transformer. *Journal of Scientific & Industrial Research (JSIR)*, *85*(1), 36-43.

[16] Nafisah, S. I., & Muhammad, G. (2024). Tuberculosis detection in chest radiograph using convolutional neural network architecture and explainable artificial intelligence. *Neural Computing and Applications*, *36*(1), 111-131.

[17] Liang, S., Xu, X., Yang, Z., Du, Q., Zhou, L., Shao, J., ... & Wang, C. (2024). Deep learning for precise diagnosis and subtype triage of drug-resistant tuberculosis on chest computed tomography. *MedComm*, *5*(3), e487.

[18] Zhang, F., Han, H., Li, M., Tian, T., Zhang, G., Yang, Z., ... & Liu, Y. (2025). Revolutionizing diagnosis of pulmonary Mycobacterium tuberculosis based on CT: a systematic review of imaging analysis through deep learning. *Frontiers in Microbiology*, *15*, 1510026.

[19] Saurina, N., Chamidah, N., Rulaningtyas, R., & Aryati, A. (2025). Multimodal deep learning from sputum image segmentation to classify Mycobacterium tuberculosis using IUATLD assessment. *Bulletin of Electrical Engineering and Informatics*, *14*(2), 1579-1590.

[20] Zhu, J., Zhao, Y., Huang, C., Zhou, C., Wu, S., Chen, T., & Zhan, X. (2025). Two-centers machine learning analysis for predicting acid-fast bacilli results in tuberculosis sputum tests. *Journal of Clinical Tuberculosis and Other Mycobacterial Diseases*, *38*, 100511.

[21] Yulvina, R., Putra, S. A., Rizkinia, M., Pujitresnani, A., Tenda, E. D., Yunus, R. E., ... & Valindria, V. (2024). Hybrid vision transformer and convolutional neural network for multi-class and multi-label classification of tuberculosis anomalies on chest X-ray. *Computers*, *13*(12), 343.

[22] El-Ghany, S. A., Elmogy, M., A. Mahmood, M., & Abd El-Aziz, A. A. (2024). A robust tuberculosis diagnosis using chest X-rays based on a hybrid vision transformer and principal component analysis. *Diagnostics*, *14*(23), 2736.

[23] Tan, M., & Le, Q. (2019, May). Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning* (pp. 6105-6114). PmLR.

[24] Woo, S., Debnath, S., Hu, R., Chen, X., Liu, Z., Kweon, I. S., & Xie, S. (2023, June). Convnext v2: Co-designing and scaling convnets with masked autoencoders. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 16133-16142). Ieee.

[25] Rahman, T. (2020). Tuberculosis (TB) chest X-ray dataset. *Kaggle*.

[26] Tawsifur Rahman, Amith Khandakar, Muhammad Enamul Hoque Chowdhury (2020). Tuberculosis

(TB) Chest X-ray Database. IEEE Dataport. https://dx.doi.org/10.21227/mps8-kb56

[27] Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2017). Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision* (pp. 618-626).

[28] Wang, H., Wang, Z., Du, M., Yang, F., Zhang, Z., Ding, S., ... & Hu, X. (2020, June). Score-CAM: Score-weighted visual explanations for convolutional neural networks. In *2020 IEEE/CVF conference on computer vision and pattern recognition workshops (CVPRW)* (pp. 111-119). IEEE.

[29] Huy, V. T. Q., & Lin, C. M. (2023). An improved densenet deep neural network model for tuberculosis detection using chest X-ray images. *Ieee Access*, *11*, 42839-42849.

[30] Xu, T., & Yuan, Z. (2022). Convolution neural network with coordinate attention for the automatic detection of pulmonary tuberculosis images on chest X-rays. *Ieee Access*, *10*, 86710-86717.