Original paper

Resource-aware colorectal computational pathology using public haematoxylin-eosin datasets: uncertainty, calibration, stability, dataset bias and controlled resolution analyses

Fucai Xie[1,2], Haijia Zhong[3], Chengyang Zhang[4], Jiaoyang Hong[2], Fuming Lai[5], Xianwen Wu[6,2], Mey Sok[7]

[1]Jiangxi University of Software Professional Technology, Nanchang, China
[2]Shinawatra University, Pathum Thani, Thailand
[3]Guangdong ATV College of Performing Arts, Zhaoqing, China
[4]East China University of Political Science and Law, Shanghai, China
[5]Jiangxi University of Chinese Medicine, Nanchang, China
[6]Jinzhong College of Health, Jinzhong, China
[7]Institute of Technology of Cambodia, Phnom Penh, Cambodia

Digital pathology increasingly connects routine haematoxylin-eosin (HE) morphology with quantitative image analysis, but pathology-facing artificial intelligence (AI) studies often report accuracy without uncertainty, calibration, stability or computational cost boundaries.

We evaluated a resource-aware workflow for colorectal cancer (CRC) tissue classification using PathMNIST, a public benchmark derived from NCT-CRC-HE-100K (National Center for Tumor Diseases – NCT) and CRC-VAL-HE-7K, plus a public native-resolution 224 × 224 NCT-CRC subset. PathMNIST contained 89,996 training, 10,004 validation and 7,180 test patches across nine tissue classes. We compared transparent colour-texture baselines with a compact convolutional neural network (TinyCNN), reporting discrimination, paired patch-level bootstrap confidence intervals, calibration, repeated-seed stability and standardised compute cost.

On the CRC-VAL-derived MedMNIST test split, TinyCNN achieved 0.848 accuracy (95% CI: 0.841–0.856) and 0.794 macro-F1 using 94,857 parameters, against 0.907 for the official ResNet-18 baseline, which uses approximately 118 times more parameters. Across six seeds, accuracy was 0.860 ±0.007, but macro one-vs.-rest area under the receiver operating characteristic curve range was 0.950–0.979 and its advantage over the random forest was not reproducible across seeds. In a controlled experiment in which only input resolution changed, accuracy fell monotonically from 0.953 at 224 pixels to 0.861 at 28 pixels. Critically, colour-histogram descriptors alone reached 0.767 accuracy and greyscale conversion reduced TinyCNN to 0.188, indicating that performance on this dataset family is substantially colour-driven rather than morphology-driven.

The 224 × 224 analysis used an image-level validation subset from the same source family, not an independent patient-level test. These findings are retrospective public benchmark evidence, not clinical deployment validation.

Key words: colorectal cancer, digital pathology, computational pathology, resource-efficient artificial intelligence, haematoxylin-eosin.

This is an Open Access journal, all articles are distributed under the terms of the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0) License (http://creativecommons.org/licenses/by-nc-sa/4.0/)

© Polish Society of Pathologists

Introduction

Digital pathology has moved from slide digitisation toward quantitative workflows supporting diagnostic, prognostic and translational research. Whole-slide and patch-level analysis measure tissue patterns at a scale manual review cannot reach. Reproducibility, however, depends on hardware, image quality, metadata, staining and computational workflow control. These factors are now recognised as part of pathology quality assurance rather than only engineering details [1]. Reviews of artificial intelligence (AI) in digital and computational pathology have also emphasised that model outputs must be interpreted in the context of pathology practice, validation design and clinical boundaries [2, 3]. Recent work in this journal illustrates the breadth of the field, from prediction of immunohistochemical marker expression directly from HE morphology [4] to the generation of virtual stains, which is additionally motivated by the tissue and chemical-waste burden of conventional staining [5].

Deep learning performs strongly in pathology image analysis, including weakly supervised whole-slide classification and cancer outcome modelling [6–8]. In colorectal cancer (CRC), public haematoxylin-eosin (HE) datasets have enabled reproducible tissue-level benchmarks. The NCT-CRC-HE-100K and CRC-VAL-HE-7K family (National Center for Tumor Diseases – NCT) provides manually extracted CRC and normal tissue patches across nine classes, and earlier work established multiclass texture analysis in CRC histology [9, 10]. Compact and ensemble convolutional networks remain competitive for multi-class medical image classification more generally [11]. These datasets are now a common substrate for algorithm benchmarking. Yet many studies still prioritise single performance values, with less attention to model footprint, training time, inference latency, calibration, uncertainty and reproducibility boundaries.

A second and more fundamental concern has recently been raised about this specific dataset family. Ignatov and Malivenko [12] showed that NCT-CRC-HE-100K and CRC-VAL-HE-7K carry pronounced class-associated colour signatures that survive Macenko normalisation, class-dependent JPEG compression artefacts, and a subset of corrupted patches produced by incorrect dynamic-range handling. In their analysis, three mean red-green-blue (RGB) intensities alone gave 53.8% accuracy and a simple colour histogram gave 82.2%, while a compact ImageNet-pretrained EfficientNet-B0 reached 97.7% and outperformed foundation transformer models and cell-morphology-aware networks. Any benchmark built on this family, including the present study, must therefore establish how much of its reported performance reflects tissue morphology and how much reflects acquisition and preprocessing signatures.

This gap matters for pathology readers. A high accuracy value is difficult to interpret without knowing whether the model is calibrated, how wide its uncertainty is, whether the result persists across image scales, whether it survives perturbation of low-level image properties and what computational burden produced it. Not every pathology group has access to large graphics processing unit clusters, whole-slide repositories or annotation budgets. Lightweight benchmarks such as MedMNIST v2 provide standardised biomedical image-classification tasks, including PathMNIST derived from the CRC HE dataset family [13]. Uncertainty-aware modelling has also been shown to improve the trustworthiness of digital histopathology predictions by identifying low-confidence cases [14].

Green AI argues that model quality should be evaluated together with computational cost [15, 16]. Related work has examined how architectural choices affect convolutional-network behaviour when image quality is degraded, which is directly relevant to benchmarks whose classes differ in acquisition and compression characteristics [17].

Here we present a reproducible public-dataset benchmark study for CRC HE tissue classification. Relative to existing PathMNIST and NCT-CRC benchmarking, the contribution of this revised study is not a new architecture or a new accuracy record; it is a reporting protocol in which discrimination is reported jointly with uncertainty, calibration, run-to-run stability, standardised compute cost, an explicit test of colour and compression dependence, and a controlled test of image resolution. Its contributions are: first, a traceable public-data workflow with a complete file-level manifest; second, performance and compute cost reported side by side under a standardised measurement protocol; third, paired bootstrap confidence intervals and fully specified calibration metrics; fourth, six-seed stability assessment; fifth, a controlled resolution experiment in which only input resolution changes; and sixth, a dataset-bias and perturbation analysis quantifying how much of the achievable performance is attributable to colour rather than morphology.

Material and methods

Study design and reporting boundary

This was a retrospective public-dataset computational pathology benchmark. No new human tissue, clinical record or identifiable patient data was collected. The primary analysis used PathMNIST, a curated benchmark derived from the public NCT-CRC-HE-100K and CRC-VAL-HE-7K HE patch datasets [10, 13]. A secondary analysis used a public native resolution 224 × 224 NCT-CRC subset. The study claims no clinical-grade diagnostic performance, prognostic utility or deployment readiness. Reporting follows the

checklist for artificial intelligence in medical imaging (CLAIM) 2024 update, which explicitly covers pathology applications [18]; a completed CLAIM checklist is provided as Supplementary S1. STARD-AI [19] was consulted for dataset, evaluation and bias reporting, and incomplete reporting of AI diagnostic-accuracy studies has been documented specifically within the pathology literature [20]. Model-reporting practice follows the model card principle of documenting intended use, evaluation data and limitations alongside performance [21]. TRIPOD+AI [22] is directed at clinical prediction models and DECIDE-AI [23] at early-stage live clinical evaluation, so both are only partly applicable to a retrospective patch-classification benchmark and are cited with that qualification.

Datasets and provenance

PathMNIST was downloaded as a single NPZ file (205,615,438 bytes) from the public Kaggle mirror dongquan/pathmnist-colon-pathology-dataset (accessed 4 July 2026; re-verified 27 August 2026). The local file contained 89,996 training, 10,004 validation and 7,180 test images, each an RGB patch resized to 28 × 28 pixels, across nine classes: adipose tissue (ADI), background (BACK), debris (DEB), lymphocytes (LYM), mucus, smooth muscle, normal colon mucosa, cancer-associated stroma (STR) and colorectal adenocarcinoma epithelium (TUM). The test split corresponds to the CRC-VAL-derived component of the original dataset family. Class counts per split are given in Supplementary Table S1.

The native-resolution analysis used the public Kaggle mirror gpreda/nct-crc-he-100k-subset (accessed 4 July 2026; manifest regenerated 27 August 2026), containing 9,000 training and 900 validation 224 × 224 patches, balanced at 1,000 and 100 images per class. The train/ and valid/ folders were supplied by the mirror and used exactly as distributed; the authors did not repartition the data, and class labels were read from the filename prefix. A SHA-256 manifest of all 9,900 files is provided as Supplementary S2; screening found no exact-duplicate image pairs within the subset or across the two folders. Crucially, the mirror does not distribute patient or slide identifiers, so slide- or patient-level separation between folders cannot be verified, and this subset is drawn from the same source family from which PathMNIST was derived. It is therefore reported as image-level validation performance on a public subset, and explicitly not as independent patient-level external validation.

Preprocessing

PathMNIST images were loaded as unsigned 8-bit RGB arrays. For neural-network training, pixel values were scaled to the 0–1 interval and arranged as channel-first tensors. For transparent baseline models, we extracted 91 low-cost colour and texture descriptors per image: channel means, standard deviations and five quantiles (21 features); per-channel 16-bin histograms (48 features); a 16-bin greyscale histogram; gradient magnitude mean, standard deviation and 90th percentile; and three colour ratios. These are low-level image descriptors. We do not claim that they correspond to pathologist-defined morphological features. No additional stain normalisation was performed; the original NCT-CRC-HE-100K dataset was reported by its creators as Macenko-normalised [10, 24].

Model architectures and optimisation

TinyCNN comprised three convolutional blocks. Each block used a 3 × 3 convolution with stride 1 and padding 1, followed by batch normalisation and ReLU; channel widths were w, 2w and 4w with w = 32 for the primary model. Blocks 1 and 2 were followed by 2 × 2 max pooling; block 3 by global adaptive average pooling, dropout (p = 0.15) and a linear classifier to nine outputs. Default PyTorch initialisation was used. Training used cross-entropy loss, AdamW (learning rate 2×10^{-3}, weight decay 10^{-4}), cosine-annealing schedule over 15 epochs, batch size 512, and augmentation by random horizontal and vertical flips and random 90° rotations. Validation was evaluated once per epoch and the checkpoint with the best validation macro-F1 was retained; no early stopping was applied. The final PathMNIST TinyCNN had 94,857 trainable parameters and a 0.371 MB checkpoint.

Lite224CNN comprised five convolutional blocks (16, 32, 64, 128, 128 channels; 3 × 3 kernels, stride 1, padding 1; batch normalisation and ReLU; 2 × 2 max pooling after the first four), followed by global average pooling, dropout (p = 0.20) and a linear classifier; 246,921 trainable parameters. It used AdamW (learning rate 10^{-3}, weight decay 10^{-4}), cosine annealing over 8 epochs and batch size 64. Because it ends in global adaptive pooling, the identical architecture accepts any input resolution, which is what makes the controlled resolution experiment below possible. Layer-by-layer specifications are given in Supplementary S3. No hyperparameter search was performed for either model: settings were fixed a priori from standard defaults, so the comparison reflects untuned architectures and no model received a larger tuning budget than another.

Transparent baselines were a stochastic gradient descent (SGD) logistic classifier (ColorTexture-SGD, standardised features, log loss) and a random forest (RF) (ColorTexture-RF, 260 trees, balanced subsample weighting) [25], implemented in scikit-learn [26]; TinyCNN and Lite224CNN in PyTorch [27].

Matched training-budget comparison

In the original submission the transparent baselines were fitted on a 45,000-image stratified subset while

TinyCNN used the full 89,996-image split, so model type and data volume varied together. We therefore added a matched comparison: TinyCNN was retrained on the identical 45,000-image stratified subset, and both transparent baselines were refitted on the full training split. We additionally computed learning curves for both model families at 5,625, 11,250, 22,500, 45,000 and 89,996 training images.

Controlled resolution experiment

To isolate the effect of image resolution, the same 224 × 224 source images with the same train/validation assignment were bilinearly downsampled to 224, 112, 56 and 28 pixels and used to train the identical Lite224CNN architecture with identical optimiser, epoch count, batch size, augmentation, random seed and checkpoint-selection rule. Input resolution is therefore the only variable that changes across these four runs. This replaces the earlier "larger-patch sensitivity analysis", which confounded resolution with image source, subset size, split, architecture and training settings.

Dataset-bias and perturbation analyses

Two analyses tested whether predictions depend on morphology or on dataset-specific image signatures. First, random forests were trained on isolated feature blocks: the three mean RGB intensities, the 48-bin colour histogram, the greyscale histogram, the gradient summary and the colour ratios. Second, the three trained primary models were re-evaluated after perturbations applied to the images only: greyscale conversion; hue shifts of ±10° and ±20°; saturation scaling by 0.5 and 1.5; brightness scaling by 0.9 and 1.1; JPEG re-compression at quality 80, 60, 40 and 20; and additive Gaussian noise ($\sigma = 5/255$). All models were scored through one probability path so the comparison is not confounded by differing decision rules. Per-class mean RGB intensities and the fraction of pixels clipped at the dynamic-range limits were also quantified as a corrupted-patch screen.

Evaluation metrics and statistical analysis

Performance was evaluated using accuracy, macro-F1, balanced accuracy, macro one-vs.-rest area under the receiver operating characteristic curve (AUROC) and multiclass log loss, with macro-F1 and balanced accuracy emphasised because class distributions differ across splits.

Uncertainty was estimated with 2,000 class-stratified bootstrap resamples (increased from 500), preserving each class count; reported intervals are 2.5th to 97.5th percentiles. These are conditional patch-level intervals. PathMNIST does not distribute patient or slide identifiers, so patches were resampled as independent units even though CRC-VAL-HE-7K originates from 50 patients and multiple patches may share a patient or slide. Patch-level resampling ignores within-patient and within-slide correlation and therefore yields narrower intervals than a patient- or slide-clustered bootstrap would. These intervals quantify sampling variability of the labelled patch set only; they do not quantify generalisation to new patients or slides and must not be read as evidence of clinical generalisability. Because separate per-model intervals do not test whether two models differ, we additionally computed paired bootstrap intervals for between-model metric differences, with both models evaluated on identical resampled patches.

Calibration was assessed on top-label confidence, defined as the maximum predicted class probability. Expected calibration error (ECE) was computed as the sample-weighted mean absolute gap between empirical accuracy and mean confidence across 10 bins, under both equal-width and equal-frequency binning; maximum calibration error is the largest such gap. The multiclass Brier score is the sum-of-squares form, the mean over images of the squared difference between the predicted probability vector and the one-hot label, with a range of 0–2. Per-bin counts are reported in Supplementary S4. No *post hoc* calibration was fitted: all probabilities are raw model outputs, and the validation split was not used to fit any calibration transformation. One-vs.-rest class-wise calibration was additionally computed. Expected calibration error is sensitive to binning and can appear favourable despite weak discrimination, as the ColorTexture-SGD result below illustrates, so it is interpreted alongside the Brier score rather than alone.

Stability was assessed by repeating TinyCNN training across six seeds. Seed 42 was fixed as the primary run before any repeated-seed result was generated, and the three additional seeds (2024, 7777, 31337) added at revision were specified before their results were inspected. All across-run summaries use the sample standard deviation (ddof = 1).

Standardised computational-cost measurement

All timings were re-measured on an idle machine: Apple M5 (10 logical cores), 16 GB unified memory, macOS 26.5.1, Python 3.9.6, PyTorch 2.8.0, scikit-learn 1.6.1, NumPy 2.0.2, float32 precision, 4 PyTorch threads, 0 data-loader workers, batch size 512. Each measurement used two discarded warm-up passes then five timed repetitions, reported as mean and sample SD, with metal performance shaders (MPS) synchronisation before stopping the timer. Transparent models ran on the central processing unit (CPU) and neural models on both CPU and MPS; the pathway is stated for every value. Per-image values are throughput-normalised batched times, not single-image latencies. Component-level and end-to-end times are both reported, because feature extraction dominates the transparent workflow while representation learning

is internal to neural training. Model size is the uncompressed on-disk size of the serialised artefact. The large validation-vs.-test inference-time difference in the original submission was an unwarmed-measurement artefact and does not persist here.

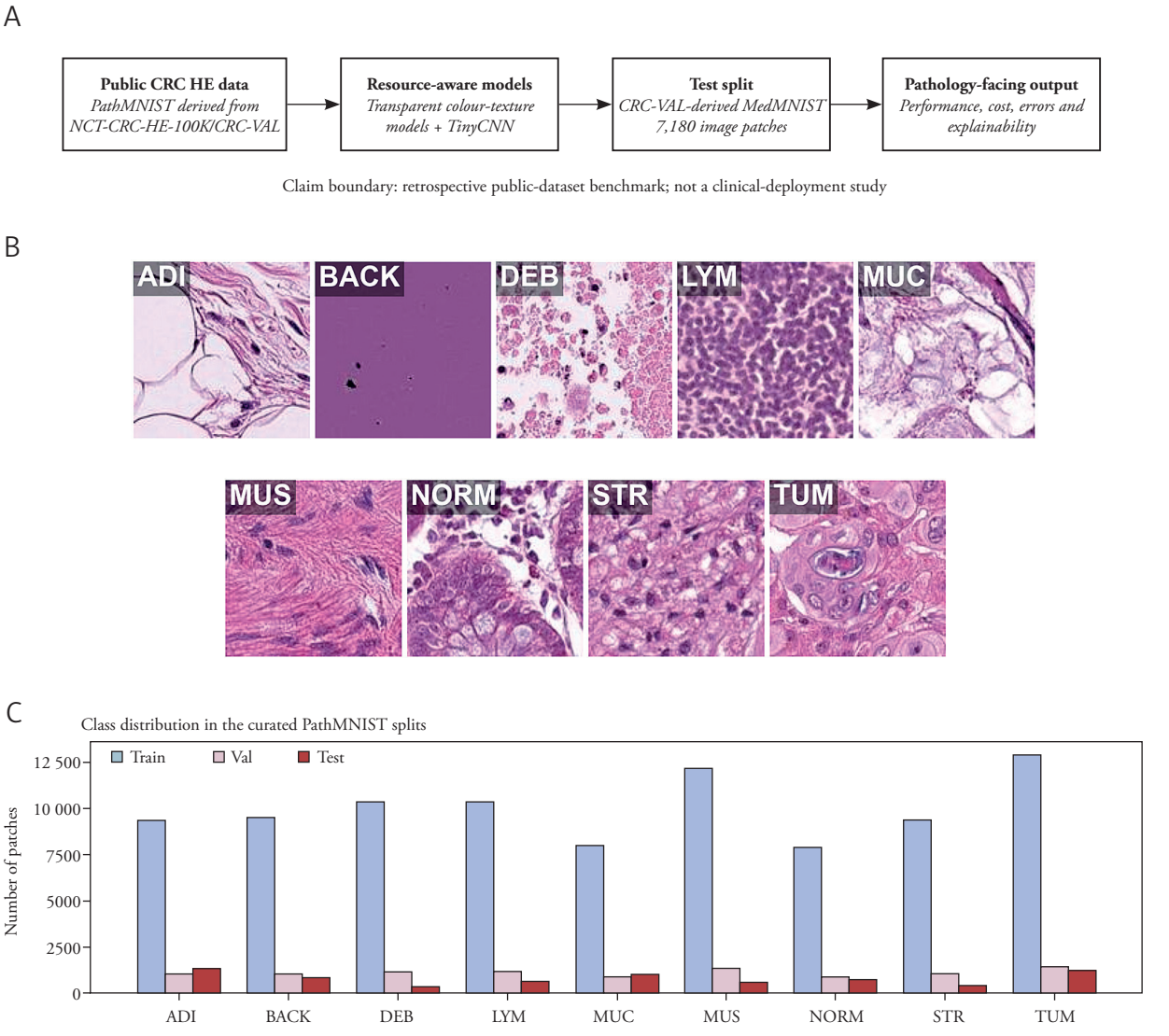
Software, reproducibility and independent verification

All code, configuration, environment lock file, dataset manifests with checksums, trained weights, raw predicted probabilities, bootstrap and calibration outputs, timing logs and figure source data are openly available at https://github.com/fufu1209/pjp-crc-resource-aware-pathology. To test dependence on one machine and one accelerator backend, the dataset-bias, repeated-seed, perturbation and controlled-resolution analyses were additionally executed under the identical protocol on free Kaggle Notebook hardware using an NVIDIA CUDA backend, a different operating system, Python version and PyTorch build. The full cross-platform comparison is given in Supplementary S5.

Results

Public-data workflow and comparison with published results

The analysis workflow is summarised in Figure 1. Table I places our results alongside published results on the same dataset family. The official MedMNIST v2 baselines reach 0.907 accuracy (ResNet-18, 28 px) and 0.911 (ResNet-50, 28 px), and at native resolution Ignatov and Malivenko [12] report 0.977 for a fine-tuned



ADI – adipose tissue, BACK – background, DEB – debris, LYM – lymphocytes, MUC – mucus, MUS – smooth muscle, NCT – National Center for Tumor Diseases, NORM – normal colon mucosa, STR – cancer-associated stroma, TUM – colorectal adenocarcinoma epithelium

Figure 1. Public-data workflow and PathMNIST tissue classes. **A**) Study workflow linking public colorectal cancer (CRC) HE data, resource-aware models, bias probing and pathology-facing outputs. **B**) Representative native-resolution HE patches from the public National Center for Tumor Diseases CRC subset, shown for visual context. **C**) Class distribution across the PathMNIST training, validation and test splits

Table I. Comparison with published results on the PathMNIST/NCT-CRC-HE dataset family

Study	Input [px]	Train/test data	Split level	Model	Pretrained	Parameters	Accuracy	AUROC
Yang *et al.* [13]	28	PathMNIST	Image	ResNet-18	ImageNet	~11.2 M	0.907	0.983
Yang *et al.* [13]	28	PathMNIST	Image	ResNet-50	ImageNet	~23.5 M	0.911	0.990
Yang *et al.* [13]	224	PathMNIST	Image	ResNet-18	ImageNet	~11.2 M	0.909	0.989
Yang *et al.* [13]	224	PathMNIST	Image	Auto-sklearn	No	NR	0.716	0.934
Ignatov *et al.* [12]	224	NCT-CRC-HE-100K/ CRC-VAL-HE-7K	Patient (test)	Colour histogram + RF	No	NA	0.822	NR
Ignatov *et al.* [12]	224	NCT-CRC-HE-100K/ CRC-VAL-HE-7K	Patient (test)	EfficientNet-B0	ImageNet	~4.0 M	0.977	NR
This study	28	PathMNIST	Image	ColorTexture-RF	No	91 features	0.770	0.963
This study	28	PathMNIST	Image	TinyCNN (w32)	No	94,857	0.848	0.979
This study	224	Public NCT-CRC subset	Image	Lite224CNN	No	246,921	0.953	0.998

AUROC – one-vs.-rest area under the receiver operating characteristic curve, NA– not applicable, NCT – National Center for Tumor Diseases, NR – not reported

Rows differ in input resolution, training data volume and pretraining and are shown for context, not as a direct ranking. All values are as reported by the cited source; hardware differs across studies.

EfficientNet-B0 [13]. TinyCNN reached 0.848 with 94,857 parameters, approximately 118 times fewer than ResNet-18, for 5.9 percentage points less accuracy. The comparison makes the exchange explicit: this study buys a very small footprint at a measurable accuracy cost, and does not compete for state-of-the-art performance. Results obtained at 28 × 28 and at 224 × 224 are shown together for context only; because input resolution, training data volume and pretraining differ, they are not a direct performance ranking.

PathMNIST test-split performance and matched training budget

TinyCNN achieved the highest accuracy and macro-F1 among the models evaluated in this study (Table II). On the CRC-VAL-derived MedMNIST test split, TinyCNN reached 0.848 accuracy, 0.794 macro-F1 and 0.979 macro-AUROC, against 0.770 and 0.714 for ColorTexture-RF and 0.695 and 0.656 for ColorTexture-SGD. Validation performance exceeded test performance for all models (TinyCNN 0.969 vs. 0.848), indicating that the CRC-VAL-derived split remains a meaningful domain-shift challenge.

Equalising the training budget did not change this ordering. TinyCNN retrained on the same 45,000-image stratified subset used by the transparent baselines reached 0.862 accuracy, slightly above its full-split result, while ColorTexture-RF refitted on the full 89,996-image split reached 0.771, essentially unchanged from 0.776 on the subset. Learning curves confirmed that both families plateau early: the random forest rose from 0.734 at 5,625 images to 0.771 at 89,996, and TinyCNN 0.828–0.848, with its best value (0.862) at 45,000 images. The performance gap therefore reflects model family rather than data volume, and additional training data would not close it.

Performance-cost relationship and model-size sweep

Under the standardised protocol, TinyCNN required 366.2 s of training, occupied 0.371 MB and processed the test split at 0.033 ms per image on the MPS pathway and 0.118 ms per image on CPU. ColorTexture-SGD fitted in 1.8 s and occupied 0.0066 MB but reached only 0.695 accuracy; ColorTexture-RF gave intermediate accuracy with a 255.0 MB model file. Feature extraction dominates the transparent workflow: end-to-end test-set processing took 1.11 s for ColorTexture-SGD and 1.16 s for ColorTexture-RF, of which approximately 1.11 s was feature extraction. Because transparent models ran on CPU and neural models on MPS, the two hardware pathways are distinguished in Figure 2 rather than plotted as a single frontier.

A width sweep showed that larger is not automatically better within this architecture. TinyCNN widths of 8, 16, 24, 32 and 48 gave 0.827, 0.828, 0.872, 0.848 and 0.868 test accuracy with 6,441, 24,393, 53,865, 94,857 and 211,401 parameters, respectively. Width 24 was the most accurate of the five, using 43% fewer parameters than width 32 and roughly a quarter of those of width 48, so within-family capacity gains are neither monotone nor reliable at this scale (Figure 2, Table II).

Table II. PathMNIST model performance, matched training budget and compute cost

A Discrimination

Model	Split	Train, *n*	Accuracy	Macro-F1	Balanced *accuracy*	Macro-AUROC
ColorTexture-SGD	Test	45,000	0.695	0.656	0.679	0.925
ColorTexture-RF	Test	45,000	0.770	0.714	0.727	0.963
TinyCNN (w32)	Validation	89,996	0.969	0.969	0.968	0.999
TinyCNN (w32)	Test	89,996	0.848	0.794	0.794	0.979
ColorTexture-SGD	Test	89,996	0.701	0.666	–	–
ColorTexture-RF	Test	89,996	0.771	0.719	–	–
TinyCNN (w32)	Test	45,000	0.862	0.806	–	–
TinyCNN (w8)	Test	89,996	0.827	0.776	–	–
TinyCNN (w16)	Test	89,996	0.828	0.757	–	–
TinyCNN (w24)	Test	89,996	0.872	0.829	–	–
TinyCNN (w48)	Test	89,996	0.868	0.830	–	–

Rows 5-11 are the matched-training-budget and width-sweep conditions added at revision. Equalising the training budget does not change the ranking.

B Computational cost

Model	Parameters	Fit/train (s)	Feature extraction [s]	End-to-end [s]	Model size (MB)	Inference [ms/image]	Pathway
ColorTexture-SGD	91 features	1.8	1.11	1.11	0.0066	0.0003	CPU
ColorTexture-RF	260 trees	14.3	1.11	1.16	255.0	0.0063	CPU
TinyCNN (w32), validation	94,857	366.2	–	–	0.371	0.0204	MPS
TinyCNN (w32), test	94,857	366.2	–	–	0.371	0.0329	MPS
TinyCNN (w8)	6,441	68.5	–	–	0.033	–	MPS
TinyCNN (w16)	24,393	89.4	–	–	0.102	–	MPS
TinyCNN (w24)	53,865	141.6	–	–	0.214	–	MPS
TinyCNN (w48)	211,401	306.3	–	–	0.817	–	MPS

AUROC – one-vs.-rest area under the receiver operating characteristic curve, CPU – central processing unit, MPS – metal performance shaders, SGD – stochastic gradient descent, RF – random forest

Timings follow the standardised protocol (two warm-up passes, five timed repetitions) on Apple M5/16 GB/macOS 26.5.1. Inference values are throughput-normalised batched times per image, not single-image latencies. Feature extraction dominates the transparent-model workflow.

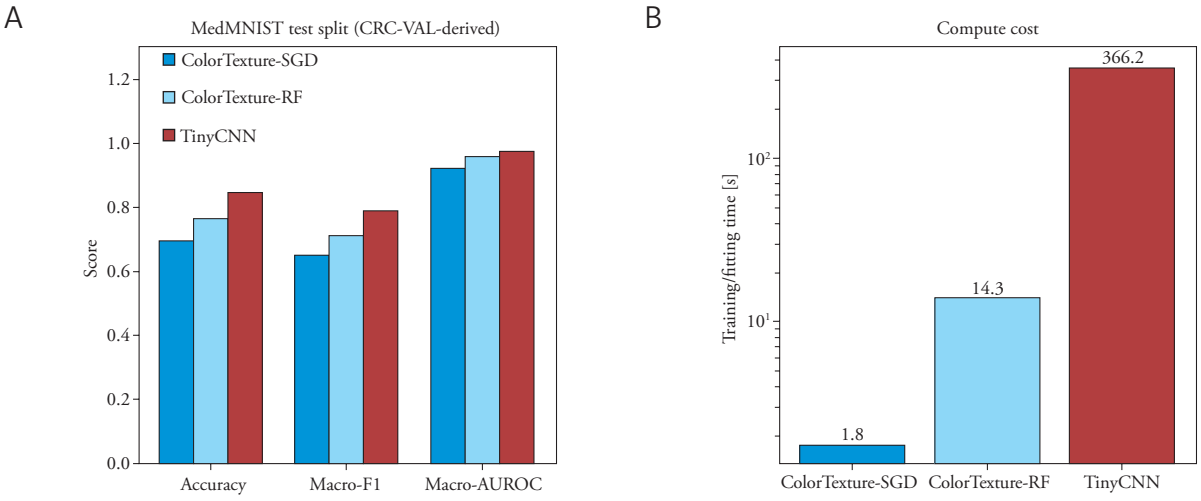
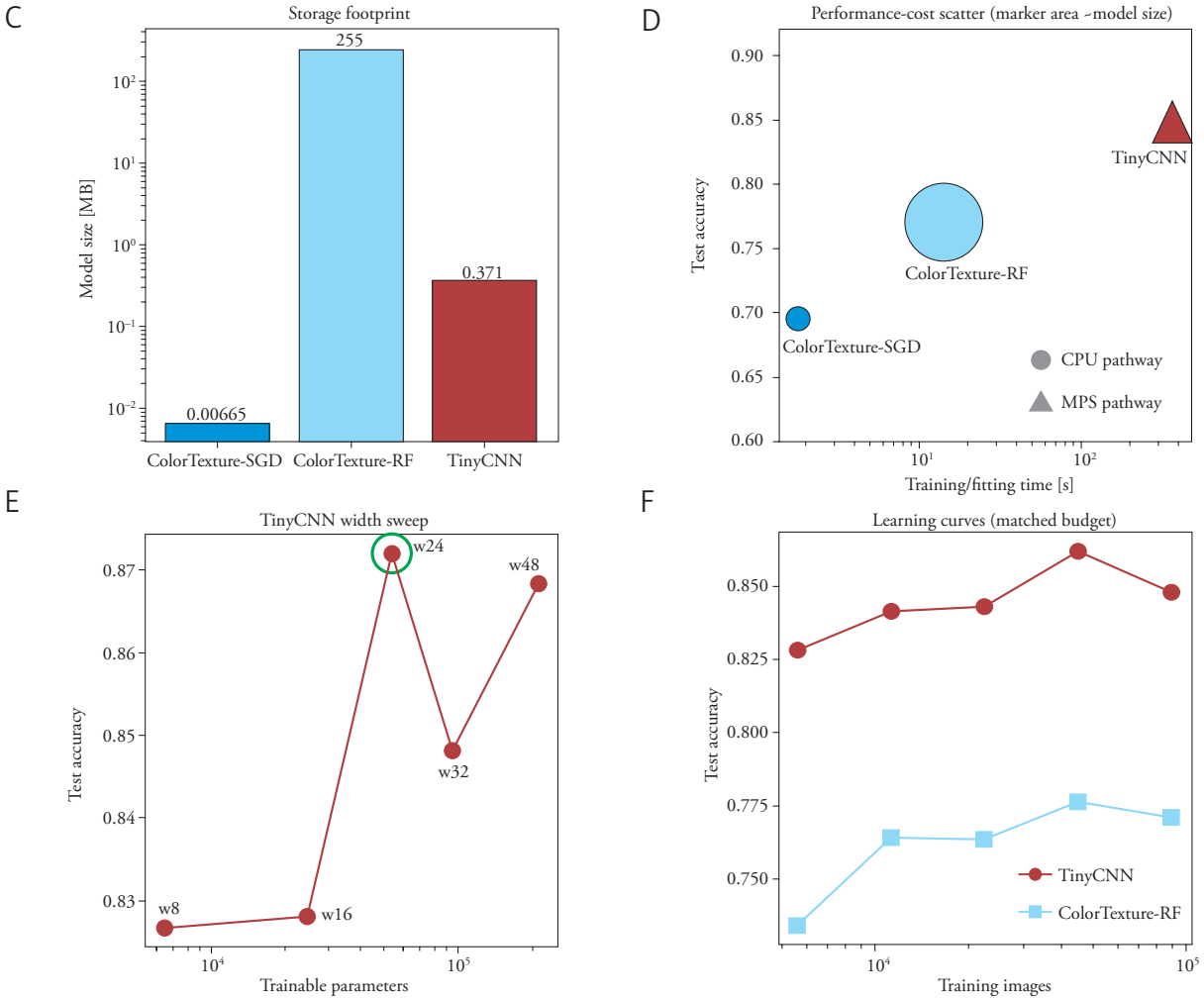


Figure 2. Performance and computational cost. **A**) Accuracy, macro-F1 and macro one-vs.-rest area under the receiver operating characteristic curve on the colorectal cancer-VAL-derived MedMNIST test split. **B**) Training or fitting time



AUROC – one-vs.-rest area under the receiver operating characteristic curve, CPU – central processing unit, MPS – metal performance shaders, SGD – stochastic gradient descent, RF – random forest

These are scatter plots over the evaluated configurations, not an exhaustive performance-cost frontier.

Figure 2. **C**) Model storage footprint. **D**) Performance-cost scatter plot; marker size reflects model size and marker shape distinguishes the CPU and metal performance shaders hardware pathways. **E**) TinyCNN width sweep, showing that width 24 outperforms width 32 with 43% fewer parameters. **F**) Learning curves for both model families as a function of training-set size

Bootstrap uncertainty and paired model comparison

With 2,000 replicates (Figure 3A), TinyCNN achieved test accuracy 0.848 (95% CI: 0.841–0.856), macro-F1 0.794 (0.784–0.804) and macro-AUROC 0.979 (0.976–0.981); ColorTexture-RF 0.770 (0.761–0.779), 0.714 (0.703–0.724) and 0.963 (0.960–0.966); ColorTexture-SGD 0.695 (0.685–0.705), 0.656 (0.645–0.667) and 0.925 (0.920–0.930).

Paired analysis was more informative than these separate intervals. TinyCNN's accuracy advantage over ColorTexture-RF was +0.078 (95% CI: 0.068–0.087) and its macro-F1 advantage +0.080 (0.068–0.091); both excluded zero for every seed tested. Its macro-AUROC advantage did not. For seed 7 the paired difference against ColorTexture-RF was −0.001 (95% CI: −0.005 to 0.004), and for seed 123 the random forest was significantly better by 0.013 (0.008–0.018). Only for seed 42 did TinyCNN hold a significant macro-AUROC advantage. Paired differences between TinyCNN seeds also excluded zero for accuracy and macro-AUROC, showing that run-to-run variation exceeds the patch-level resampling variability captured by the bootstrap. Conclusions about ranking are therefore secure for accuracy and macro-F1 but not for macro-AUROC.

Calibration

On the test split (Figure 3B), TinyCNN had ECE 0.052 under equal-width and 0.051 under equal-frequency binning, Brier 0.222 and negative log likelihood 0.467. ColorTexture-RF had ECE 0.088 and Brier 0.335; ColorTexture-SGD had ECE 0.050 but Brier 0.482 and negative log likelihood 2.168, confirming that a favourable ECE can coexist with weak discrimination and that ECE must not be read alone. Agreement between the two binning schemes indicates the estimates are not artefacts of one binning choice. Class-wise calibration identified DEB as the worst-calibrat-

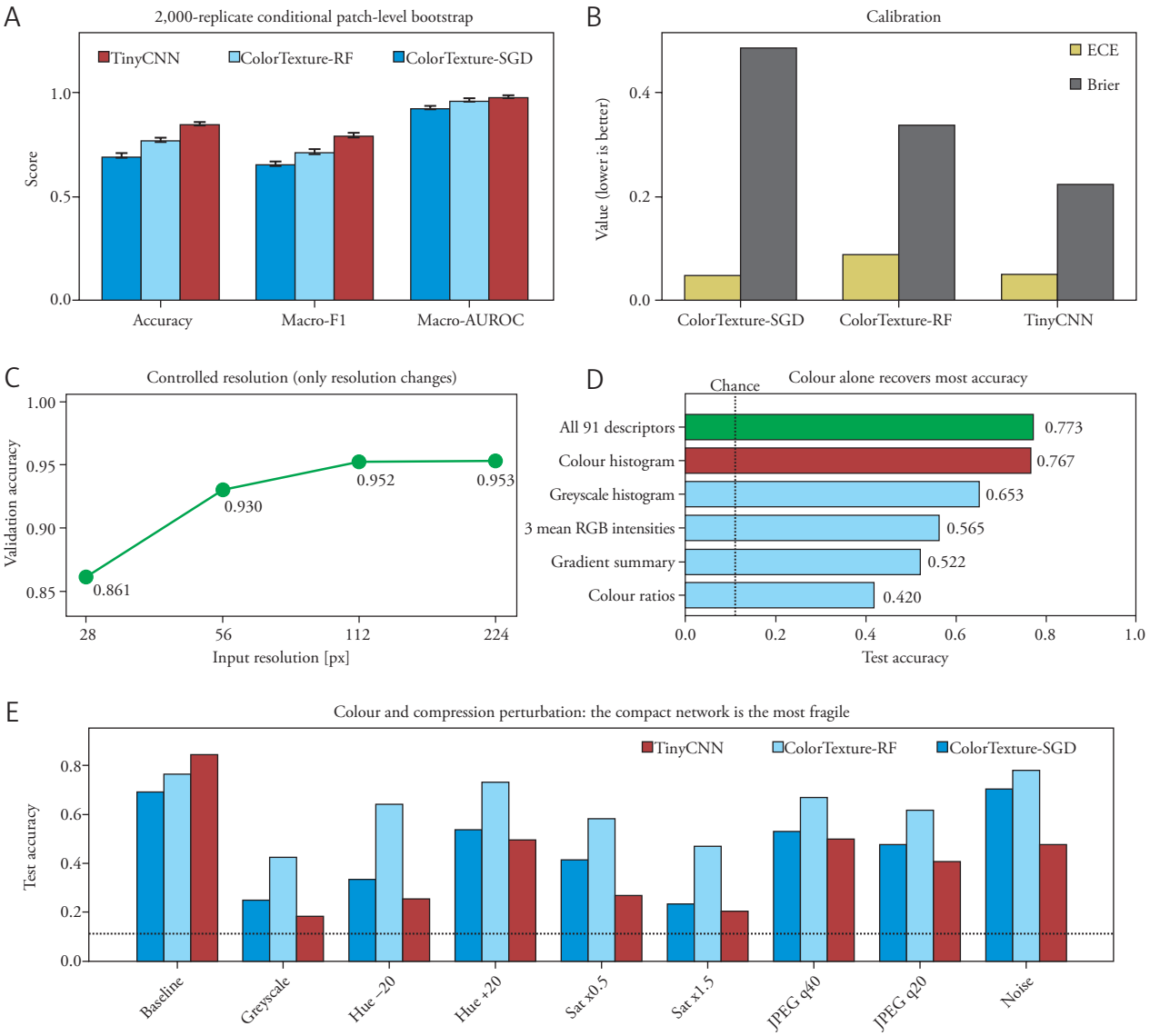
ed class for both models (one-vs.-rest ECE 0.035 for TinyCNN, 0.058 for ColorTexture-RF), and TinyCNN systematically under-predicted STR (mean predicted probability 0.037 against 0.059 prevalence), consistent with its low stroma recall.

Repeated-seed stability

Across six seeds (42, 7, 123, 2024, 7777, 31337), test accuracy was 0.860 ±0.007 (range 0.848–0.868) and macro-F1 0.808 ±0.014 (0.793–0.824). Macro-AUROC was markedly less stable at 0.969 ±0.011, spanning 0.950 to 0.979 – a range wider than the 0.976–0.981 bootstrap interval of any single run (Table III). Validation metrics were more stable (accuracy 0.968 ±0.001). Training time varied 154–366 s under an otherwise identical protocol; this reflects thermal and scheduling variation on a shared laptop-class device rather than any algorithmic difference, which is itself an argument for reporting compute cost as a range rather than a single value. The accuracy and macro-F1 conclusions are therefore stable across initialisation, whereas the macro-AUROC value should not be quoted from a single run.

Controlled resolution experiment

With image source, split assignment, architecture, optimiser, epochs, batch size, augmentation and seed



AUROC – one-vs.-rest area under the receiver operating characteristic curve, ECE – expected calibration error, RGB – red-green-blue, SGD – stochastic gradient descent, RF – random forest

Bootstrap intervals are patch-level and do not quantify uncertainty for new patients or slides

Figure 3. Uncertainty, calibration, resolution and colour dependence. **A**) Test accuracy, macro-F1 and macro one-vs.-rest area under the receiver operating characteristic curve with 2,000-replicate conditional patch-level 95% bootstrap intervals. **B**) Expected calibration error and multiclass Brier score. **C**) Controlled resolution experiment: accuracy against input resolution with all other factors held constant. **D**) Accuracy of random forests trained on isolated feature blocks, showing that a colour histogram alone recovers most achievable accuracy. **E**) Accuracy under colour and compression perturbation for all three models, with the nine-class chance level marked

all held constant, validation accuracy fell monotonically with resolution: 0.953 at 224 px, 0.952 at 112 px, 0.930 at 56 px and 0.861 at 28 px (Table IV, Figure 3C). Training time fell from 220 to 8.5 s and throughput-normalised inference from 1.33 to 0.028 ms per image across the same range. Halving resolution 224–112 px cost 0.001 accuracy while reducing training time threefold, whereas reducing to 28 px cost 0.092 accuracy. Resolution is thus a genuine and quantifiable determinant of performance on this task, and the 28 px result (0.861) is close to the PathMNIST TinyCNN result (0.848), which supports interpreting the PathMNIST setting as a low-resolution regime rather than a different problem.

Colour and compression dependence

Colour information alone accounted for most achievable performance (Figure 3D). Trained on the full split and evaluated on the test split, a random forest using only the three mean RGB intensities reached 0.565 accuracy, and one using only the 48-bin colour histogram reached 0.767 – close to the 0.773 obtained with all 91 descriptors. Greyscale histogram, gradient summary and colour-ratio blocks alone gave 0.653, 0.522 and 0.420. These values closely reproduce, at 28 × 28, the 0.538 and 0.822 reported at 224 × 224 [12], and were confirmed on independent Kaggle hardware (0.569 and 0.773).

Table III. TinyCNN repeated-seed stability on the PathMNIST test split (six seeds)

Seed	Accuracy	Macro-F1	Balanced acc.	Macro-AUROC	ECE	Brier	Training time [s]
42 (primary)	0.848	0.794	0.794	0.979	0.052	0.222	366.2
7	0.863	0.811	0.810	0.962	0.051	0.216	240.2
123	0.856	0.793	0.794	0.950	0.053	0.228	232.4
2024	0.863	0.806	–	0.972	–	–	186.9
7777	0.868	0.824	–	0.977	–	–	161.8
31337	0.865	0.823	–	0.972	–	–	153.9
Mean ± SD	0.860 ±0.007	0.808 ±0.014	–	0.969 ±0.011	0.052 ±0.001	0.222 ±0.006	223.6 ±78.4

AUROC – one-vs.-rest area under the receiver operating characteristic curve, ECE – expected calibration error

Standard deviations are sample values (ddof = 1). Seed 42 was designated the primary run before any repeated-seed result was produced; seeds 2024, 7777 and 31337 were specified at revision before their results were inspected. Macro-AUROC spans 0.950–0.979, a range wider than the bootstrap interval of any individual run.

Table IV. Controlled resolution experiment and perturbation robustness

A Controlled resolution – only input resolution changes

Input resolution [px]	Accuracy	Macro-F1	Macro-AUROC	Training time [s]	Inference [ms/image]
224 (native)	0.953	0.954	0.998	219.9	1.333
112	0.952	0.952	0.998	71.8	0.225
56	0.930	0.930	0.995	19.6	0.124
28	0.861	0.862	0.987	8.5	0.028

Identical source images, split assignment, Lite224CNN architecture, optimiser, epochs, batch size, augmentation, seed and checkpoint rule.

B Robustness on the PathMNIST test split (accuracy; change from baseline)

Perturbation	ColorTexture-SGD	ColorTexture-RF	TinyCNN
Baseline	0.695	0.770	0.848
Grayscale	0.254 (–0.442)	0.428 (–0.342)	0.188 (–0.660)
Hue –20°	0.338 (–0.357)	0.645 (–0.126)	0.259 (–0.589)
Hue +20°	0.544 (–0.151)	0.734 (–0.036)	0.499 (–0.349)
Saturation × 0.5	0.419 (–0.276)	0.587 (–0.183)	0.273 (–0.575)
Saturation × 1.5	0.240 (–0.455)	0.473 (–0.297)	0.208 (–0.640)
JPEG q = 40	0.537 (–0.159)	0.673 (–0.098)	0.501 (–0.347)
JPEG q = 20	0.483 (–0.213)	0.620 (–0.150)	0.411 (–0.437)
Gaussian noise σ = 5/255	0.708 (+0.013)	0.784 (+0.014)	0.483 (–0.365)

AUROC – one-vs.-rest area under the receiver operating characteristic curve, SGD – stochastic gradient descent, RF – random forest

Chance accuracy is 0.111 for nine classes. Full perturbation results, including hue ±10° and brightness scaling, are in Supplementary S4.

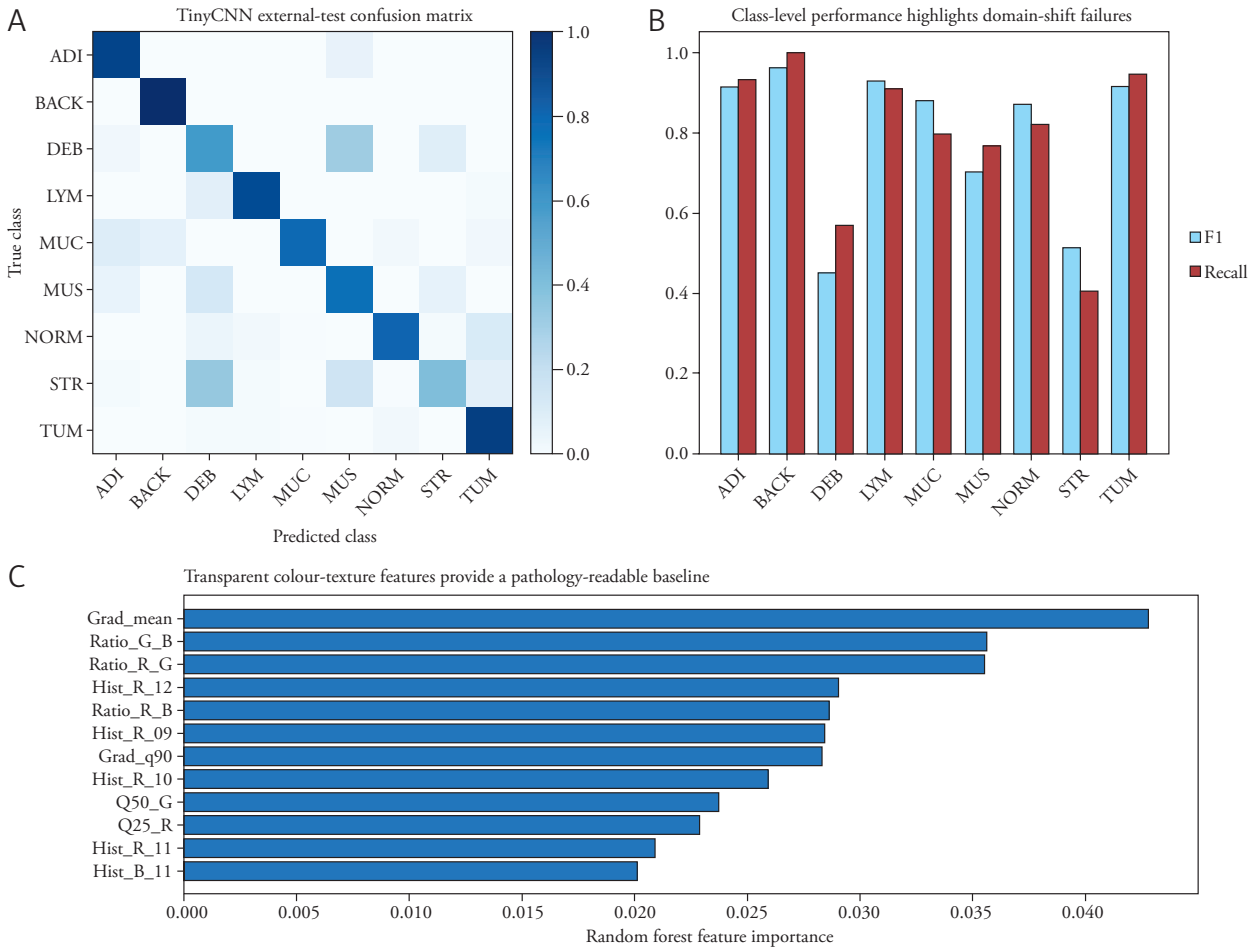
Perturbation testing showed that the compact neural network was the most colour-dependent model, not the least (Table IV, Figure 3E). Greyscale conversion reduced TinyCNN from 0.848 to 0.188 – close to the 0.111 chance level for nine classes – while ColorTexture-RF fell to 0.428 and ColorTexture-SGD to 0.254. Hue shifts of ±20° cost TinyCNN up to 0.589 accuracy against 0.126 for ColorTexture-RF; a 1.5× saturation change cost 0.640 against 0.297. JPEG re-compression to quality 20 cost TinyCNN 0.437 and ColorTexture-RF 0.150, and additive Gaussian noise cost TinyCNN 0.365 while leaving both transparent models unchanged. Class-wise screening confirmed distinct per-class colour signatures, and dynamic-range clipping was confined to a single class: 2.45% of DEB pixels sat at the 8-bit ceiling and 4.7% of DEB test patches had more than 10% saturated pixels, while every other class showed none. TinyCNN's advantage on the unperturbed benchmark therefore rests substantially on colour and low-level texture regularities of this dataset family and should not be interpreted as superior morphological modelling.

Independent cross-platform verification

Re-executing the analyses on Kaggle hardware with a CUDA backend reproduced every substantive finding (Supplementary S5): colour-only feature blocks agreed to within 0.007, the monotone resolution decline was reproduced (0.949, 0.950, 0.916 and 0.858), greyscale conversion again collapsed TinyCNN, to 0.131, and across six seeds accuracy was 0.867 ±0.008 against 0.860 ±0.007 locally. Individual seeds did not reproduce exactly, as expected when neither backend guarantees deterministic reduction order; this reinforces the case for multi-seed reporting. Macro-AUROC, however, varied far less across seeds on CUDA (SD 0.002) than locally (SD 0.011), so the magnitude of that instability is itself platform-dependent.

Class-specific confusion patterns and interpretable descriptors

The TinyCNN confusion matrix (Figure 4A) showed strong performance for BACK, LYM, ADI and TUM



ADI – adipose tissue, BACK – background, DEB – debris, LYM – lymphocytes, MUC – mucus, MUS – smooth muscle, NCT – National Center for Tumor Diseases, NORM – normal colon mucosa, STR – cancer-associated stroma, TUM – colorectal adenocarcinoma epithelium

Importance indicates which inputs the model uses and does not establish pathological relevance.

Figure 4. Class-specific confusion patterns and interpretable descriptors. **A**) Row-normalised TinyCNN confusion matrix on the test split. **B**) Test-set F1-score and recall per tissue class; debris and cancer-associated stroma were weakest. **C**) Random-forest importance for low-level colour-texture descriptors

(test F1 0.962, 0.928, 0.915, 0.916). The weakest were DEB and STR (F1 0.452 and 0.514); STR recall was 0.406, DEB recall was 0.569 with precision of 0.375 (Figure 4B). Confusion matrices and per-class metrics for all models are provided in Supplementary S4. These are observed confusion patterns. Representative correctly and incorrectly classified patches with their predicted probabilities and dataset labels are provided as Supplementary S6, but no pathologist adjudication of individual cases was performed, so we do not attribute them to morphological ambiguity as distinct from labelling error, image corruption or preprocessing artefact. Debris is the only class in this split showing dynamic-range clipping, and the highest-confidence correct DEB predictions are visibly saturated patches (Supplementary S6). Image corruption is therefore a plausible contributor to this class's behaviour that morphological interpretation alone would miss.

Random-forest feature importance ranked gradient mean, green/red/blue colour ratios, red-channel histogram bins and the gradient 90th percentile most highly (Figure 4C). These are interpretable low-level image descriptors, not pathologist-defined morphological features, and the perturbation results above indicate that they partly encode dataset-specific acquisition and preprocessing signatures. Feature importance identifies which inputs the model uses; it does not establish pathological or causal relevance.

Discussion

This study shows that a resource-aware computational pathology workflow can classify CRC HE tissue categories on public datasets while explicitly reporting the cost, uncertainty and limitations of the modelling choices. The most consequential finding of this revision is negative: on this dataset family, colour-histogram descriptors alone reach 0.767 accuracy, and removing colour reduces our compact neural network to near chance. This independently reproduces, at low resolution and on separate hardware, the dataset-bias analysis of Ignatov and Malivenko [12]. Reported accuracy on NCT-CRC-derived benchmarks should therefore be interpreted as performance on a task that is substantially solvable from staining and acquisition signatures, and improvements of a few percentage points between architectures may reflect low-level image properties rather than better morphological modelling.

This reframes the study's contribution. The defensible claim is not that TinyCNN is a good CRC tissue classifier, but that a reporting protocol combining uncertainty, calibration, stability, standardised cost, bias probing and controlled resolution reveals limitations that accuracy alone conceals. Three examples arose here. First, TinyCNN's macro-AUROC advantage over a random forest disappeared or reversed depending on the random seed, which single-run reporting would have hidden. Second, between-seed variation exceeded patch-level bootstrap intervals, so those intervals overstate the security of the estimate. Third, the model with the best benchmark accuracy was the most fragile under colour perturbation. Each of these is invisible in a conventional accuracy-only report.

Two reviewer-motivated controls strengthened rather than weakened the original conclusions. Matching the training budget left the ranking unchanged, and learning curves showed both families plateau well before the full split, so the performance gap is attributable to model family rather than data volume. The controlled resolution experiment, in which only resolution changed, showed a monotone decline from 0.953 at 224 px to 0.861 at 28 px, replacing an earlier comparison that confounded resolution with dataset, split, architecture and training settings.

The findings align with the broader movement toward reproducible and accountable digital pathology [1–3]. Hardware, staining and scanner variation remain first-order determinants of reproducibility [1], and our perturbation results are a low-cost proxy for exactly that concern: a model that loses 0.44 accuracy under moderate JPEG re-compression is unlikely to survive a change of scanner or laboratory.

The Green AI framing is not a claim of reduced carbon emissions. We did not measure device power draw and did not estimate carbon dioxide equivalents. Training time, model size, parameter count and inference latency are reported as computational-resource proxies rather than environmental-impact measures, and we accordingly use "resource-efficient" rather than "sustainable" [15, 16]. The width sweep supports this concretely: width 24 outperformed width 32 with 43% fewer parameters, so capacity increases are not reliably rewarded.

The study has several limitations. First, the primary benchmark is 28 × 28, and the controlled experiment quantifies what that costs (0.092 accuracy relative to 224 px) rather than removing the constraint. Second, the 224 × 224 subset is drawn from the same source family as PathMNIST, lacks patient and slide identifiers, and is therefore image-level validation, not independent external validation. Third, bootstrap intervals are conditional and patch-level; a patient- or slide-clustered bootstrap would be wider but is impossible without identifiers. Fourth, six seeds characterise but do not exhaust run-to-run variability, and repeated-seed training was not performed at native resolution. Fifth, no hyperparameter search was performed for any model, so all reported values are untuned. Sixth, no pathologist adjudicated individual error cases, so confusion patterns are described but not attributed to a cause. Seventh, we did not evaluate the non-normalised version of the NCT dataset, which would allow the colour-bias analysis to be separated

from the Macenko normalisation step. Eighth, no energy or carbon measurement was performed.

Future work should apply this reporting protocol to the complete native-resolution NCT-CRC-HE-100K and CRC-VAL-HE-7K patch sets with patient-level splits, to datasets that do not share this family's colour and compression artefacts, and to whole-slide cohorts with scanner and site metadata and pathologist-reviewed error cases [18, 19, 22, 23]. Benchmarks whose headline metric is largely recoverable from colour histograms are poor substrates for claims about morphological understanding, and the field would benefit from routinely reporting a colour-ablation result alongside accuracy.

Conclusions

Compact neural networks and transparent colour-texture baselines were evaluated on public CRC HE datasets derived from NCT-CRC-HE-100K and CRC-VAL-HE-7K. Support for the reporting protocol should be distinguished from support for any particular classifier. We do not conclude that TinyCNN is a preferred CRC tissue classifier: its accuracy advantage over transparent baselines was reproducible, but its macro-AUROC advantage was seed-dependent, and it was the most fragile model under colour and compression perturbation. What the study does support is that reporting discrimination together with paired uncertainty, fully specified calibration, multi-seed stability, standardised compute cost, a colour-ablation probe and a controlled resolution experiment materially changes what a benchmark result means. On this dataset family, colour alone recovers most of the achievable accuracy, and any performance claim should be qualified accordingly. Conclusions are limited to retrospective public-dataset benchmarks and do not extend to clinical deployment.

Disclosures

1. Institutional review board statement: Not applicable.
2. Assistance with the article: No.
3. Financial support and sponsorship: None.
4. Conflicts of interest: None.

References

1. Li Y, Li T. Hardware and quality-control factors influencing digital pathology reproducibility. Pol J Pathol 2025; 76: 275-283.
2. Bera K, Schalper KA, Rimm DL, Velcheti V, Madabhushi A. Artificial intelligence in digital pathology – new tools for diagnosis and precision oncology. Nat Rev Clin Oncol 2019; 16: 703-715.
3. Song AH, Jaume G, Williamson DFK, Lu MY, Vaidya A, Miller TR, et al. Artificial intelligence for digital and computational pathology. Nat Rev Bioeng 2023; 1: 930-949.
4. Miller M, Przybylska M, Pakuło S, Świderska-Chadaj Ż. The role of computational pathology in predicting expression of selected immunohistochemical markers. Pol J Pathol 2026; 77: 209-215.
5. Korski K. Generation of virtual stains: can artificial intelligence imitate chemistry? Pol J Pathol 2026; 77: 203-208.
6. Campanella G, Hanna MG, Geneslaw L, Miraflor A, Werneck Krauss Silva V, Busam KJ, et al. Clinical-grade computational pathology using weakly supervised deep learning on whole slide images. Nat Med 2019; 25: 1301-1309.
7. Janowczyk A, Madabhushi A. Deep learning for digital pathology image analysis: a comprehensive tutorial with selected use cases. J Pathol Inform 2016; 7: 29.
8. Kather JN, Krisam J, Charoentong P, Luedde T, Herpel E, Weis CA, et al. Predicting survival from colorectal cancer histology slides using deep learning: a retrospective multicenter study. PLoS Med 2019; 16: e1002730.
9. Kather JN, Weis CA, Bianconi F, Melchers SM, Schad LR, Gaiser T, et al. Multi-class texture analysis in colorectal cancer histology. Sci Rep 2016; 6: 27988.
10. Kather JN, Halama N, Marx A. 100,000 histological images of human colorectal cancer and healthy tissue (v0.1). Zenodo 2018. Available from: https://doi.org/10.5281/zenodo.1214456.
11. Halder A, Dalal A, Gharami S, Woźniak M, Ijaz MF, Singh PK. A fuzzy rank-based deep ensemble methodology for multi-class skin cancer classification. Sci Rep 2025; 15: 6268.
12. Ignatov A, Malivenko G. NCT-CRC-HE: not all histopathological datasets are equally useful. In: Lecture notes in computer science. Springer, Cham 2025, 300-317.
13. Yang J, Shi R, Wei D, Liu Z, Zhao L, Ke B, et al. MedMNIST v2 – a large-scale lightweight benchmark for 2D and 3D biomedical image classification. Sci Data 2023; 10: 41.
14. Dolezal JM, Srisuwananukorn A, Karpeyev D, Ramesh S, Kochanny S, Cody B, et al. Uncertainty-informed deep learning models enable high-confidence predictions for digital histopathology. Nat Commun. 2022; 13: 6572.
15. Schwartz R, Dodge J, Smith NA, Etzioni O. Green AI. Commun ACM 2020; 63: 54-63.
16. Lannelongue L, Grealey J, Inouye M. Green algorithms: quantifying the carbon footprint of computation. Adv Sci 2021; 8: 2100707.
17. Goh KW, Surono S, Afiatin MYF, Mahmudah KR, Irsalinda N, Chaimanee M, et al. Comparison of activation functions in convolutional neural network for Poisson noisy image classification. Emerg Sci J 2024; 8: 592-602.
18. Tejani AS, Klontzas ME, Gatti AA, Mongan JT, Moy L, Park SH, et al. Checklist for artificial intelligence in medical imaging (CLAIM): 2024 update. Radiol Artif Intell 2024; 6: e240300.
19. Sounderajah V, Guni A, Liu X, Collins GS, Karthikesalingam A, Markar S, et al. The STARD-AI reporting guideline for diagnostic accuracy studies using artificial intelligence. Nat Med 2025; 31: 3283-3289.
20. McGenity C, Bossuyt P, Treanor D. Reporting of artificial intelligence diagnostic accuracy studies in pathology abstracts: compliance with STARD for abstracts guidelines. J Pathol Inform 2022; 13: 100091.
21. Mitchell M, Wu S, Zaldivar A, Barnes P, Vasserman L, Hutchinson B, et al. Model cards for model reporting. In: Proceedings of the Conference on Fairness, Accountability, and Transparency 2019, 220-229.
22. Collins GS, Moons KGM, Dhiman P, Riley RD, Beam AL, van Calster B, et al. TRIPOD+AI statement: updated guidance for reporting clinical prediction models that use regression or machine learning methods. BMJ 2024; 385: e078378.
23. Vasey B, Nagendran M, Campbell B, Clifton DA, Collins GS, Denaxas S, et al. Reporting guideline for the early-stage clinical evaluation of decision support systems driven by artificial intelligence: DECIDE-AI. Nat Med 2022; 28: 924-933.
24. Macenko M, Niethammer M, Marron JS, Borland D, Woosley JT, Guan X, et al. A method for normalizing histology slides for

quantitative analysis. 2009 IEEE International Symposium on Biomedical Imaging: From Nano to Macro 2009, 1107-1110.
25. Breiman L. Random forests. Mach Learn 2001; 45: 5-32.
26. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: machine learning in Python. J Mach Learn Res 2011; 12: 2825-2830.
27. Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, et al. PyTorch: an imperative style, high-performance deep learning library. Adv Neural Inf Process Syst 2019; 32: 8024-8035.

Address for correspondence

Fucai Xie
Jiangxi University of Software Professional Technology/No. 89
Luoting Avenue, Xinjian District
Nanchang 330041
Jiangxi, China
e-mail: xiefucai42@gmail.com