Purpose

The primary results in this manuscript were produced on a single Apple-silicon workstation using the Metal Performance Shaders (MPS) backend. To test whether any reported finding depends on that machine, that accelerator backend or that software stack, the revision analyses were re-executed under an identical protocol on free Kaggle Notebook hardware, using an NVIDIA CUDA backend and a different operating system, Python version and PyTorch build.

The script is included in the public repository as 07_Kaggle_Ready/pjp_revision_kaggle_verification.py and the notebook is publicly runnable, so any reader can repeat this verification without local hardware.

Platforms compared

	Primary (local)	Verification (Kaggle)
Device	Apple M5, 10 logical cores	Tesla P100-PCIE-16GB, 4 vCPU
Accelerator backend	Metal Performance Shaders	CUDA 12.4
Operating system	macOS 26.5.1	Linux 6.12.90, glibc 2.35
Python	3.9.6	3.12.13
PyTorch	2.8.0	2.6.0+cu124
scikit-learn	1.6.1	1.6.1
NumPy	2.0.2	2.0.2

A note on the PyTorch version: Kaggle's stock image ships a CUDA 12.8 build, which no longer compiles kernels for compute capability 6.0. The Tesla P100 belongs to that generation, so every kernel launch failed while torch.cuda.is_available() still returned True. The verification script detects this by executing a probe convolution in a subprocess before importing PyTorch, and installs a CUDA 12.4 build, which still contains sm_60 kernels. Both platforms therefore ran on their respective GPUs.

A. Dataset-bias feature blocks

Random forests trained on isolated feature blocks, full training split, evaluated on the CRC-VAL-derived MedMNIST test split.

Feature block	Local (M5 / MPS)	Kaggle (P100 / CUDA)	Difference
3 mean RGB intensities	0.5652	0.5685	0.0033
Colour histogram (48 bins)	0.7667	0.7730	0.0063
All 91 transparent descriptors	0.7730	0.7758	0.0028

The manuscript's central new claim — that colour-only descriptors recover most of the achievable accuracy on this dataset family — reproduces to within 0.007 on independent hardware. These models are CPU-only, so the residual difference reflects random-forest non-determinism across platforms rather than any accelerator effect.

B. Repeated-seed TinyCNN training

Six pre-specified seeds, identical architecture, optimiser, schedule, epochs, batch size, augmentation and checkpoint-selection rule.

Seed	Local accuracy	Kaggle accuracy	Difference
42	0.8482	0.8695	0.0213
7	0.8632	0.8663	0.0031
123	0.8560	0.8529	0.0031
2024	0.8628	0.8678	0.0050
7777	0.8678	0.8720	0.0042
31337	0.8646	0.8742	0.0096
Mean ± SD	**0.8604 ± 0.0071**	**0.8671 ± 0.0075**	0.0067

The across-seed mean agrees to 0.007 and the across-seed spread is almost identical (SD 0.0071 versus 0.0075). Individual seeds do not reproduce exactly, and are not expected to: neither backend guarantees bitwise-deterministic convolution or reduction order, so the same seed indexes a different trajectory on a different accelerator.

This observation reinforces rather than weakens the manuscript's argument. If a single seed is not reproducible across hardware, then a single-run number is not a stable description of a model, and multi-seed reporting is necessary rather than optional.

A platform-dependent nuance in macro-AUROC

	Local (MPS)	Kaggle (CUDA)
Macro-AUROC range across six seeds	0.9501 – 0.9788	0.9715 – 0.9768
Macro-AUROC SD across six seeds	0.0108	0.0019

Macro-AUROC was markedly less stable across seeds on the local platform than on Kaggle. The instability reported in the manuscript is therefore real on the platform that produced the primary results, but **its magnitude is itself platform-dependent** and should not be read as a universal property of the architecture. We report this explicitly rather than selecting whichever platform gives the tidier story. The substantive conclusion is unchanged and, if anything, strengthened: macro-AUROC on this task is sensitive to factors outside the model definition, so it should not be quoted from a single run.

C. Controlled resolution experiment

Identical source images, split assignment, architecture, optimiser, epochs, batch size, augmentation and seed; only input resolution changes.

Input resolution	Local accuracy	Kaggle accuracy	Difference	Local train (s)	Kaggle train (s)
224 px	0.9533	0.9489	0.0044	219.9	98.4
112 px	0.9522	0.9500	0.0022	71.8	24.2
56 px	0.9300	0.9156	0.0144	19.6	11.6
28 px	0.8611	0.8578	0.0033	8.5	8.3

The monotone decline with decreasing resolution reproduces on both platforms, as does the specific shape of the curve: 224 px and 112 px are within 0.005 of each other on both, while 28 px costs approximately 0.09 accuracy on both. The manuscript's quantitative claim about the cost of resolution reduction is therefore not an artefact of one machine.

Training times differ as expected between a discrete GPU and an integrated one, which is why the manuscript reports compute cost with the hardware pathway stated for every value rather than as a hardware-independent quantity.

D. Colour and compression perturbation

TinyCNN, seed 42, evaluated on perturbed copies of the test split.

Perturbation	Local accuracy	Kaggle accuracy	Difference
Baseline	0.8482	0.8695	0.0213
Grayscale	**0.1882**	**0.1305**	0.0577
Hue −20°	0.2589	0.2749	0.0160
Hue +20°	0.4992	0.4908	0.0084
Saturation × 0.5	0.2728	0.2439	0.0290
Saturation × 1.5	0.2084	0.1646	0.0437
JPEG q = 40	0.5008	0.5318	0.0309
JPEG q = 20	0.4110	0.4369	0.0259

The collapse under grayscale conversion reproduces on both platforms and is in fact more severe on Kaggle (0.131 versus 0.188; chance accuracy is 0.111 for nine classes). Every colour perturbation reduces accuracy by a large margin on both platforms. The manuscript's conclusion — that this compact network's benchmark performance depends substantially on the colour signature of the dataset rather than on tissue morphology — is confirmed on independent hardware.

Summary

Claim	Verified independently?
Colour-only descriptors recover most achievable accuracy	Yes, within 0.007
Grayscale conversion collapses the compact network toward chance	Yes, more severe on Kaggle
All colour perturbations cause large accuracy loss	Yes
Accuracy declines monotonically with resolution, 224 ≈ 112 ≫28	Yes
Across-seed accuracy mean and spread	Yes, mean within 0.007, SD 0.0071 vs 0.0075
Individual seeds reproduce exactly across platforms	**No** — and this supports multi-seed reporting
Magnitude of macro-AUROC seed instability	**Platform-dependent** — reported explicitly above

Raw outputs — per-seed metrics, per-epoch logs, predicted probabilities, model weights, environment capture and the run log — are included in the public repository under 07_Kaggle_Ready/kaggle_results_v3/.