Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Design and Patient-Level Evaluation of a Web-Based Convolutional Neural Network System for Detecting Malaria Parasites in Thin Blood Smear Cell Images

Abdusamea Omer

aia@sabu.edu.ly

Libyan Center for Electronic Systems, Software, and Aviation Research, Libya

Rabyah B. Ali

General Nursing Department, Faculty of Nursing – Surman, Sabratha University, Libya

Esraa Mustafa Melid

Department of Computer Engineering and Information Technology, Faculty of Engineering, Sabratha University, Libya

Aisha Lutfi Abdul Latif

Department of Computer Engineering and Information Technology, Faculty of Engineering, Sabratha University, Libya

Abdulhakeem Rotimi Agboola

Department of Biochemistry, College of Medical Sciences, University of Calabar, Calabar, Cross River State, Nigeria

Anietie Justin Atangwho

Department of Family Medicine, University of Calabar Teaching Hospital, Calabar Cross River State, Nigeria

Item Justin Atangwho

Department of Biochemistry, College of Medical Sciences, University of Calabar, Calabar, Cross River State, Nigeria

---

Research Article

Keywords: Malaria, Thin blood smear, Convolutional neural network, Patient-level cross-validation, Data leakage, Web application

Posted Date: November 26th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11527641/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Abstract

Light microscopy of Giemsa-stained blood smears is the reference method for diagnosing malaria, yet reading slides is slow and depends on expertise that is scarce where the disease is common. We developed a lightweight convolutional neural network (CNN) that labels segmented red blood cell images as parasitized or uninfected and placed it behind a browser-based decision-support application. The network, with three convolutional blocks and 4.83 million parameters, was evaluated on the public National Library of Medicine collection of 27,558 cell images from 200 individuals using five-fold cross-validation in which every cell from a given individual stayed in the same fold. On the held-out folds, accuracy was 94.4% (95% confidence interval [CI] 93.2–95.5%, patient-level bootstrap), sensitivity 93.4% (91.4–95.0%), specificity 95.4% (93.9–96.7%) and the area under the receiver operating characteristic curve 0.980 (0.974–0.985). Splitting the same images at random, without regard to the individual, raised accuracy to 95.2% and shrank its between-fold standard deviation from 1.35 to 0.26 percentage points, so cell-level evaluation overstated both performance and its stability. Errors were concentrated in a minority of individuals: per-patient accuracy had a median of 97.0% but fell below 90% for 15 of the 200. Model scores were well calibrated (expected calibration error 0.009), and one image was classified in about 22 ms on a standard processor. The application is intended to assist microscopists, not to replace them; validation on independently collected smears and an automated cell-segmentation stage are required before any clinical use.

1 Introduction

Malaria is a potentially fatal febrile illness that results from infection with *Plasmodium* parasites, which are passed to people through the bites of infected female *Anopheles* mosquitoes. The World Health Organization (WHO) estimates that 282 million malaria cases and 610,000 malaria deaths occurred worldwide in 2024, the great majority of them in the WHO African Region and among young children [1]. Five species routinely infect humans, namely *P. falciparum*, *P. vivax*, *P. ovale*, *P. malariae* and *P. knowlesi*, and *P. falciparum* causes most severe disease and deaths [2].

Light microscopy of Giemsa-stained thick and thin blood smears remains the reference standard for diagnosis because it confirms infection, identifies the species and allows parasite density to be estimated [3]. Its quality rests on the skill and workload of the microscopist; examining a slide is slow, and readers disagree. Rapid diagnostic tests are easier to deploy but cannot quantify parasitaemia and may miss infections at low parasite densities [3]. These constraints bite hardest where the disease burden is highest and trained laboratory staff are fewest.

Automated image analysis has long been explored as a way to make microscopy faster and less dependent on individual expertise. Early systems combined hand-crafted colour, texture and shape descriptors with conventional classifiers; more recent work trains convolutional neural networks (CNNs) that learn discriminative features from the pixels themselves [4]. The U.S. National Library of Medicine

(NLM) released a large, expert-annotated collection of segmented red blood cell images [5], and that collection has since become the common test bed for such methods.

Two weaknesses run through much of this literature. The first concerns evaluation. Every individual in the NLM collection contributes dozens to hundreds of cell images that share staining, illumination and camera characteristics, so a random split of images places near-duplicates of the same smear in both the training and the test sets. Information leaks across the split, a problem recognised across machine-learning-based science [6], and the resulting estimates describe performance on familiar slides rather than on new patients. Rajaraman et al. avoided this by cross-validating at the patient level [5, 7], but many later reports do not state how their data were partitioned. The second weakness concerns delivery: a classifier that exists only as a research notebook cannot help a laboratory technician, whereas one served through an ordinary web browser needs no local installation and can be updated centrally.

This paper addresses both points. Its contributions are (i) a compact, fully specified CNN evaluated with patient-grouped five-fold cross-validation over all 27,558 images, with confidence intervals obtained by resampling individuals; (ii) a direct comparison of patient-level and cell-level partitioning under identical training settings, which quantifies how much a random split flatters the results; (iii) an analysis of per-patient performance, score calibration and error cases; and (iv) the integration of the model into a Flask web application whose inference time was measured. Section 2 reviews related work, Section 3 describes the data, model, evaluation design and application, Section 4 reports the results, Section 5 discusses them together with the study's limitations, and Section 6 concludes.

2 Related work

Before deep learning became widespread, automated malaria microscopy relied on pipelines that segmented cells and then classified them from engineered features; Poostchi et al. survey these approaches in detail [4]. Bibin et al. moved towards learned representations by training a deep belief network on colour and texture features extracted from 4,100 peripheral blood smear images [8].

Liang et al. were among the first to apply a CNN to the NLM single-cell images. Their 16-layer network reached an average accuracy of 97.37% in ten-fold cross-validation over 27,578 cell images and clearly outperformed a transfer-learning baseline (91.99%) [9]. Rajaraman et al., who also described the dataset, used pre-trained CNNs as fixed feature extractors and evaluated them with cross-validation at the patient level; under that protocol their best model (ResNet-50) achieved 95.9% accuracy [5, 7]. A follow-up study by the same group combined several networks into ensembles, again keeping each patient's cells within a single fold [7].

Other work has focused on efficiency and deployment. Fuhad et al. trained compact models that reached 99.23% accuracy with only about 4,600 floating-point operations; along the way they found and corrected mislabelled images in the dataset, and they deployed their model on mobile phones and in a server-backed web application [10]. Masud et al. trained a custom CNN with a cyclical stochastic gradient descent schedule, reported 97.30% accuracy and packaged the model in a mobile application [11]. Yu et

al. built Malaria Screener, an Android application that combines image acquisition through a microscope adapter, cell detection and classification, and result management in one semi-automated workflow for thin and thick smears [12]. A relabelled version of the thin-smear images, in which questionable samples were re-examined, has also been published [13].

Table 1 lists the studies closest to the present one. The figures were produced with different partitioning schemes, label versions and evaluation levels, so they cannot be ranked against each other; the table positions this work rather than claiming superiority. What this paper adds is not a higher accuracy figure but a transparent and reproducible patient-level evaluation, a measured comparison with the cell-level alternative, and a lightweight web delivery route.

Table 1
Selected studies using the NLM thin-smear cell images

Study	Approach	Reported accuracy (%)	Deployment
Liang et al. [9]	16-layer custom CNN; ten-fold cross-validation	97.37	None reported
Rajaraman et al. [5]	Pre-trained CNNs as feature extractors (best: ResNet-50); patient-level cross-validation	95.9	None reported
Fuhad et al. [10]	Compact CNNs (about 4,600 FLOPs); corrected labels	99.23	Mobile and web
Masud et al. [11]	Custom CNN; cyclical SGD	97.30	Mobile
This study	Three-block CNN (4.83 M parameters); patient-level five-fold cross-validation	94.39 (95% CI 93.21–95.51)	Web (Flask)

CNN, convolutional neural network; FLOPs, floating-point operations; SGD, stochastic gradient descent; CI, confidence interval. Values are as reported by the original authors. Partitioning schemes, label versions and evaluation levels differ between studies, so the figures are not directly comparable.

3 Materials and methods

The system has two parts (Fig. 1). In an offline stage the CNN is trained and evaluated; in an online stage the trained model is served through a web application. Both stages apply the same image pre-processing, so the model receives inputs at inference time that are formatted exactly like those it was trained on.

3.1 Dataset

We used the publicly available NLM malaria cell image dataset [5, 14], obtained through its Kaggle mirror. It contains 27,558 single red blood cell images, 13,779 labelled parasitized and 13,779 uninfected. The cells were segmented from images of Giemsa-stained thin blood smears collected from 150 *P.*

falciparum-infected patients and 50 uninfected individuals at Chittagong Medical College Hospital, Bangladesh. The smears were photographed through a light microscope with a smartphone camera, and an expert slide reader at the Mahidol-Oxford Tropical Medicine Research Unit, Bangkok, annotated the cells [5]. The original labels were used without modification.

The dataset providers encode the source smear of every cell in its file name and also release files that map cells to patients [15]. We extracted this identifier from each file name (for example, the prefix C33P1 in C33P1thinF_IMG_20150619_114756a_cell_179.png). The procedure yielded exactly 200 groups, 150 containing parasitized cells and 50 containing only uninfected cells, which matches the documented composition of the collection; every file name was parsed. Individuals contributed between 65 and 702 cells (median 85.5). Infected patients contribute cells to both classes, because the uninfected cells on their smears are labelled as such.

3.2 Pre-processing

The source images vary in size, so each was resized to 150 × 150 pixels with three colour channels using bilinear interpolation, and pixel intensities were divided by 255 to lie between 0 and 1. No data augmentation, stain normalisation or other enhancement was applied. The web application performs the same two operations on every uploaded image.

3.3 CNN architecture

The classifier is a sequential CNN with three convolution–pooling blocks followed by two fully connected layers (Fig. 2). Each block applies a convolutional layer with 3 × 3 kernels, stride 1, no padding and a rectified linear unit (ReLU) activation, then 2 × 2 max pooling. The number of filters doubles from block to block (32, 64 and 128) while each pooling step halves the spatial resolution, reducing a 150 × 150 × 3 input to a 17 × 17 × 128 feature map. This map is flattened into 36,992 values and passed to a dense layer of 128 ReLU units and then to a single sigmoid output unit, whose value we call the model score. A cell is labelled parasitized when the score is at least 0.5.

Table 2 lists the output shape and number of trainable parameters of each layer. Of the 4,828,481 parameters, 4,735,104 (98.1%) sit in the first dense layer; the three convolutional layers together hold fewer than 0.1 million.

Table 2
Layer-wise output shapes and trainable parameters of the CNN

Layer	Configuration	Output shape	Parameters
Input	RGB image	150 × 150 × 3	0
Conv2D-1	32 filters, 3 × 3, ReLU	148 × 148 × 32	896
MaxPooling2D-1	2 × 2	74 × 74 × 32	0
Conv2D-2	64 filters, 3 × 3, ReLU	72 × 72 × 64	18,496
MaxPooling2D-2	2 × 2	36 × 36 × 64	0
Conv2D-3	128 filters, 3 × 3, ReLU	34 × 34 × 128	73,856
MaxPooling2D-3	2 × 2	17 × 17 × 128	0
Flatten	–	36,992	0
Dense-1	128 units, ReLU	128	4,735,104
Dense-2	1 unit, sigmoid	1	129
Total			**4,828,481**

3.4 Evaluation design

The primary evaluation used five-fold cross-validation grouped by individual (Fig. 3b). Folds were formed with the StratifiedGroupKFold procedure of scikit-learn [16] (shuffled, random seed 42), which assigns all cells of an individual to one fold while keeping the class balance of the folds as even as the grouping allows. In each round, one fold served as the test set. From the remaining individuals, 10% were drawn at random as a validation set used only for early stopping, and the rest formed the training set. Programmatic checks confirmed in every round that no individual appeared in more than one of the three subsets. Because each image falls in exactly one test fold, the procedure yields one out-of-fold prediction for every image in the collection. Table 3 gives the size of each subset.

For comparison, the whole procedure was repeated with an ordinary stratified five-fold split of the images (Fig. 3a), using the same seeds and training settings and a random 10% of the training images for validation. Under this scheme every test fold contained cells from all 200 individuals.

Table 3
Composition of the patient-level cross-validation folds

Fold	Test individuals	Test cells	Parasitized in test (%)	Training cells	Validation cells
1	40	4,801	41.1	20,486	2,271
2	40	4,872	44.4	19,832	2,854
3	39	5,295	49.6	19,472	2,791
4	41	5,821	51.6	20,011	1,726
5	40	6,769	59.2	18,364	2,425

Training and validation sets were drawn from the individuals not in the test fold; no individual appeared in more than one subset of a round.

3.5 Training configurations

The model was implemented in Python 3.13 with TensorFlow 2.21 and its Keras interface [17, 18]; scikit-learn, NumPy and Matplotlib were used for partitioning, numerical work and plotting [16, 19, 20]. Training ran in a Kaggle cloud notebook on one NVIDIA Tesla T4 graphics processing unit. All runs minimised the binary cross-entropy loss in mini-batches of 32 images, monitored the validation loss for early stopping, and restored the weights of the epoch with the lowest validation loss.

Two optimiser settings were trained (Table 4). Configuration A used the Adam optimiser [21] with a learning rate of 10^{-4}, at most 30 epochs and a patience of five epochs; these values were adopted from the authors' preliminary experiments. Configuration B used plain stochastic gradient descent with the Keras default learning rate of 0.01 and no momentum, at most 20 epochs and a patience of three epochs. Configuration A was designated the primary model because it reached the lower mean best validation loss across the patient-level folds (0.153 versus 0.167). This criterion uses only validation data, so the choice did not depend on the test folds.

Table 4
Training configurations

Setting	Configuration A (primary)	Configuration B
Optimiser	Adam	SGD (no momentum)
Learning rate	1×10^{-4}	0.01
Maximum epochs	30	20
Early-stopping patience (validation loss)	5	3
Loss / batch size	Binary cross-entropy / 32	Binary cross-entropy / 32
Best weights restored	Yes	Yes
Input / normalisation	150 × 150 × 3 / pixel ÷ 255	150 × 150 × 3 / pixel ÷ 255
Data augmentation / dropout	None	None
Decision threshold	0.5	0.5
Mean best validation loss (patient-level folds)	0.153	0.167

3.6 Performance measures

The parasitized class was treated as positive. A true positive (TP) is a parasitized cell predicted as parasitized, a true negative (TN) an uninfected cell predicted as uninfected, a false positive (FP) an uninfected cell predicted as parasitized and a false negative (FN) a parasitized cell predicted as uninfected. The following measures were computed:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

1

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

2

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

3

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}}$$

4

$$\mathrm{F1} = \frac{2 \times \ \mathrm{Precision} \times \ \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}}$$

5

$$\mathrm{MCC} = \frac{\mathrm{TP} \times \ \mathrm{TN} - \mathrm{FP} \times \ \mathrm{FN}}{\sqrt{(\mathrm{TP} + \mathrm{FP})(\mathrm{TP} + \mathrm{FN})(\mathrm{TN} + \mathrm{FP})(\mathrm{TN} + \mathrm{FN})}}$$

6

We also report the negative predictive value, balanced accuracy, and the areas under the receiver operating characteristic curve (ROC-AUC) and the precision–recall curve (PR-AUC). Primary estimates were computed from the pooled out-of-fold predictions; the mean and standard deviation (SD) across the five folds are given alongside. Cells from the same individual are not independent observations, so 95% confidence intervals were obtained with a cluster bootstrap that resampled individuals rather than cells (2,000 resamples, percentile intervals) [22].

Calibration, that is, the agreement between model scores and observed frequencies, was assessed with the Brier score [23], the expected calibration error over ten equal-width score bins [24] and a reliability diagram. Accuracy was also computed separately for each individual to examine how performance varied across patients. Finally, as an exploratory analysis, we read from the pooled ROC curve the thresholds at which sensitivity reached 95% and 98%. These thresholds were taken from the same predictions they describe and are therefore optimistic.

3.7 Web application

The web application (Fig. 1b) follows a conventional client–server design. Its front end, written in HTML, CSS and JavaScript, adapts to the screen size and can be used from desktop computers, laptops and smartphones without installing software. The back end is a Flask application [25] that loads the saved Keras model once, when the server starts, and keeps it in memory. A user selects a cell image and submits it; the image is sent to the server in an HTTP POST request, pre-processed as in Sect. 3.2 and passed to the model. The server returns a page showing the predicted class and the model score. Inference time was measured on the four-core Intel Xeon processor (2.00 GHz) of the Kaggle environment as the duration of one forward pass for a single image, over 200 images after ten warm-up passes; image decoding and network transfer were excluded.

The application expects a single segmented red blood cell image of the kind found in the training data. Whole microscope fields of view must first be divided into individual cells, a step the system does not yet perform. The application has so far been run and tested only on a local development server and has not been deployed publicly; policies for storing or deleting uploaded images, and the associated security measures, will need to be defined before any public deployment.

3.8 Use of AI-assisted tools

During the preparation of this work the authors used Claude (Anthropic), a large language model, to help restructure and edit the manuscript text, to draft the cross-validation and analysis code, and to generate the data figures from the authors' results. The authors executed all experiments in their own computing environment, checked the code, results, references and text, and take full responsibility for the content of this publication.

4 Results

4.1 Training behaviour

With Configuration A, training stopped after 9 to 15 epochs, and the restored weights came from epochs 4 to 10, depending on the fold. One fold took between 2.8 and 4.3 min to train. At the restored epoch, training accuracy ranged from 94.6% to 96.8%. In every fold the training loss kept falling after the validation loss had reached its minimum (Fig. 4a), the usual sign that further epochs would have fitted the training cells more closely without improving performance on unseen individuals; early stopping halted training at that point. The validation curves also differed between folds: fold 5 ended with the highest best validation loss (0.217) and fold 4 with the lowest (0.105), an early hint that difficulty depended on which individuals a fold contained.

4.2 Patient-level performance

Table 5 summarises the performance of the primary model on held-out individuals. Over the 27,558 out-of-fold predictions, accuracy was 94.39% (95% CI 93.21–95.51%), sensitivity 93.38% (91.41–94.99%) and specificity 95.41% (93.88–96.68%). The ROC-AUC was 0.980 (0.974–0.985) and the Matthews correlation coefficient (MCC) 0.888 (0.864–0.910). Of the 13,779 parasitized cells, 912 were missed; of the 13,779 uninfected cells, 633 were flagged as parasitized (Fig. 5a). The fold-level means agreed closely with the pooled values, and the ROC curves of the individual folds were similar (Fig. 6a), with AUCs between 0.969 and 0.987. Per-fold results are listed in Table 6.

Table 5
Performance of the primary model (Configuration A) under patient-level five-fold cross-validation

Measure	Pooled estimate (95% CI)	Mean ± SD across folds
Accuracy (%)	94.39 (93.21–95.51)	94.44 ± 1.35
Sensitivity (%)	93.38 (91.41–94.99)	93.33 ± 2.39
Specificity (%)	95.41 (93.88–96.68)	95.44 ± 2.53
Precision (%)	95.31 (93.05–96.74)	95.23 ± 2.29
Negative predictive value (%)	93.51 (91.03–95.68)	93.55 ± 2.42
F1-score (%)	94.34 (92.85–95.44)	94.24 ± 1.45
Balanced accuracy (%)	94.39 (93.21–95.49)	94.38 ± 1.44
MCC	0.888 (0.864–0.910)	0.888 ± 0.027
ROC-AUC	0.980 (0.974–0.985)	0.981 ± 0.007
PR-AUC	0.981 (0.974–0.986)	0.981 ± 0.007
Brier score	0.045 (0.037–0.054)	–

Pooled estimates use all 27,558 out-of-fold predictions; 95% confidence intervals (CI) are from a bootstrap that resampled the 200 individuals (2,000 resamples). SD, standard deviation; MCC, Matthews correlation coefficient; ROC-AUC and PR-AUC, areas under the receiver operating characteristic and precision–recall curves. Predictive values depend on the proportion of parasitized cells (50% in this collection) and would differ in routine practice.

Table 6
Per-fold results of the primary model

Fold	Accuracy (%)	Sensitivity (%)	Specificity (%)	MCC	ROC-AUC	Epochs run (restored)
1	95.67	95.24	95.97	0.911	0.985	9 (4)
2	93.33	89.32	96.53	0.865	0.969	15 (10)
3	95.96	94.33	97.56	0.920	0.987	12 (7)
4	92.96	94.74	91.05	0.859	0.981	13 (8)
5	94.27	93.01	96.09	0.884	0.982	11 (6)

4.3 Patient-level versus cell-level evaluation

Table 7 sets the two partitioning schemes side by side for both training configurations. With Configuration A, the cell-level split reported an accuracy of 95.23 ± 0.26% against 94.44 ± 1.35% under

the patient-level split, and an ROC-AUC of 0.985 against 0.981. The gap in mean accuracy was modest, at 0.8 percentage points (1.0 point for Configuration B), but the spread across folds was five times smaller under the cell-level scheme. The same pattern held for every measure: the cell-level folds looked both slightly better and far more uniform. Under both schemes Configuration A exceeded Configuration B in accuracy, sensitivity, MCC and ROC-AUC, consistent with its lower validation loss; specificity was similar.

Table 7
Effect of the partitioning scheme and training configuration (mean ± SD across five folds)

Configuration / split	Accuracy (%)	Sensitivity (%)	Specificity (%)	MCC	ROC-AUC
A (Adam) / patient-level	94.44 ± 1.35	93.33 ± 2.39	95.44 ± 2.53	0.888 ± 0.027	0.981 ± 0.007
A (Adam) / cell-level	95.23 ± 0.26	94.16 ± 1.06	96.31 ± 0.62	0.905 ± 0.005	0.985 ± 0.001
B (SGD) / patient-level	93.29 ± 1.97	91.07 ± 4.74	95.71 ± 2.15	0.867 ± 0.038	0.976 ± 0.011
B (SGD) / cell-level	94.30 ± 1.20	92.82 ± 2.23	95.77 ± 1.07	0.887 ± 0.024	0.981 ± 0.003

Patient-level folds keep all cells of an individual in one fold; cell-level folds assign images at random, so every test fold contained cells from all 200 individuals. SD, standard deviation; MCC, Matthews correlation coefficient.

4.4 Variation across individuals and error cases

Performance was far from uniform across individuals (Fig. 5b). Per-patient accuracy had a median of 97.0% (interquartile range 93.8–98.5%), but it fell below 90% for 15 of the 200 individuals and below 80% for 8; the lowest value was 43.0%. Errors were concentrated accordingly. Sixteen of the 150 infected patients had a per-patient sensitivity below 80%, and together they accounted for 232 of the 912 missed cells (25.4%). Conversely, 43 infected patients had no missed parasitized cell at all. On the specificity side, 12 individuals with per-patient specificity below 90% contributed 284 of the 633 false positives (44.9%); three infected patients alone contributed 154 (24.3%), because most of the cells labelled uninfected on their smears received high scores.

Figure 7 shows the most confident errors. Several cells labelled parasitized but scored near zero show no distinct stained inclusion at this resolution, whereas several cells labelled uninfected but scored near one contain clearly stained structures, some inside the cell and some at its edge. These observations are qualitative and were not confirmed by a second expert reader.

4.5 Calibration, operating points and inference time

The model scores tracked observed frequencies closely (Fig. 6b): the Brier score was 0.045 (95% CI 0.037–0.054) and the expected calibration error 0.009. Most scores lay near the extremes, with 12,118 images scoring below 0.1 and 11,699 above 0.9. In the exploratory threshold analysis, lowering the threshold from 0.5 to 0.29 raised pooled sensitivity to 95.0% at a specificity of 92.3%, and a threshold of 0.05 gave a sensitivity of 98.0% at a specificity of 74.7%. A forward pass for one image took a median of 22.1 ms (95th percentile 22.9 ms) on the four-core processor.

5 Discussion

5.1 Principal findings

A three-block CNN with fewer than five million parameters classified cells from unseen individuals with an accuracy of 94.4% and an ROC-AUC of 0.980, produced well-calibrated scores, and ran in about 22 ms per image on an ordinary processor. Evaluating the same model with a random split of the images gave a higher and, more strikingly, a much more stable picture. The errors that remained were not spread evenly: a small group of individuals accounted for a large share of them.

5.2 What the partitioning scheme changes

The cell-level split raised mean accuracy by less than one percentage point, which might suggest that leakage matters little for this collection. The fold-to-fold variation tells a different story. Under the cell-level scheme every test fold contained cells from every individual, so each fold posed essentially the same problem and the standard deviation of accuracy fell to 0.26 points. Under the patient-level scheme a fold could contain an unusually difficult set of individuals, and the standard deviation rose to 1.35 points. Our reading is that cell-level evaluation on this dataset mainly creates false precision: confidence intervals computed as if cells were independent would be too narrow, and differences between methods of one to two percentage points could arise from the choice of split alone. The patient-level bootstrap interval for accuracy reported here (93.21–95.51%) is about four times as wide as the interval obtained when every cell is treated as an independent observation (94.12–94.66%, Wilson method [26]).

This has a bearing on the published figures in Table 1. Results of 97–99% obtained under cell-level or unspecified splits are not necessarily inflated by several points, since our data suggest a gap closer to one point for this kind of model, but they cannot be compared with each other at the precision their decimals imply. The 95.9% reported by Rajaraman et al. under patient-level cross-validation [5, 7] is the most comparable figure, and our 94.4% lies close to it with a much smaller model.

5.3 Where the model fails

The concentration of errors in a few individuals points to factors that act at the level of the smear rather than the cell. Staining intensity, illumination and focus vary between slides, and a model trained on other slides may misread an unusual one systematically. Label noise is a second candidate: three infected patients contributed a quarter of all false positives because the model scored most of their “uninfected”

cells as parasitized, and the most confident errors in Fig. 7 include apparently empty cells labelled parasitized and stained inclusions in cells labelled uninfected. Mislabelled images are known to exist in this collection [10, 13]. The present data cannot separate these explanations; re-evaluating the model on the relabelled version of the images [13], and having an expert re-read the slides of the poorly classified individuals, would test them directly.

5.4 Implications for the web-based system

Calibration matters here because the application shows its users a score, not only a label. Within this collection a score of 0.8 corresponded to roughly 80% of cells being parasitized, so the displayed value carries real information about uncertainty. That property was measured on the same population the model was trained on, however, and neural network scores often become over-confident on data from new settings [24]; calibration would need to be checked again before scores from other laboratories could be read the same way. The threshold analysis illustrates the trade-off a screening deployment would face: catching 98% of parasitized cells cost a quarter of uninfected cells being flagged for review. Any such threshold should be fixed on separate validation data, not on the predictions reported here.

Serving the model through a browser removes software installation, ensures that every user runs the same model version, and allows the model to be replaced centrally. A forward pass of 22 ms on a modest processor means that the server, not the model, is likely to limit response time. The cost is the need for a network connection; applications that run entirely on the phone, such as those of Fuhad et al. and Yu et al., avoid it at the price of a more complex installation and update process [10, 12].

5.5 Limitations

All images in the collection come from one hospital, were photographed with a smartphone camera on a light microscope, and contain only *P. falciparum* infections in Giemsa-stained thin smears [5]. Patient-level cross-validation measures generalisation to new individuals from that same setting; it says nothing about other species, staining protocols, microscopes or cameras, including those in Libyan and Nigerian laboratories. External validation on independently collected smears is the most important next step.

The model classifies pre-segmented single cells. Diagnosis is made per patient, from a whole smear, and requires cell detection and aggregation of cell-level results, neither of which this study evaluated. The collection was also assembled with roughly equal numbers of parasitized and uninfected cells, so predictive values and any patient-level decision rule would behave differently at realistic parasite densities.

Only two training configurations were compared, without augmentation, dropout or a wider hyperparameter search, and each configuration was trained once per fold. The original labels were used even though some are known to be wrong [10, 13], which affects both training and the measured error rates. Patient identifiers were derived from file names; the derivation reproduced the documented numbers of infected and uninfected individuals exactly, but it was not cross-checked against the providers' mapping files [15]. Finally, the web application has been tested only on a local server; it has

not been deployed publicly or evaluated with laboratory staff, and its security and data-protection measures have not been formally assessed.

6 Conclusions

A compact CNN served through a Flask web application classified thin-smear red blood cell images from previously unseen individuals with 94.4% accuracy (95% CI 93.2–95.5%) and an ROC-AUC of 0.980, with well-calibrated scores and an inference time of about 22 ms per image. Evaluating the same images with a random cell-level split slightly raised the estimated accuracy and sharply understated its variability, and the remaining errors clustered in a small number of individuals. Studies on this dataset should therefore partition by patient and report uncertainty that accounts for the clustering of cells.

Future work will test the model on independently collected smears from laboratories in Libya and Nigeria, including species other than *P. falciparum*; re-evaluate it on the relabelled images; add automated cell detection so that whole field-of-view images can be analysed and parasitaemia estimated; and assess the usability, response time and data security of the web application with laboratory staff. The system is meant to support microscopists, and its outputs should always be interpreted alongside clinical findings and confirmatory testing.

Declarations

Acknowledgements

The authors thank the Lister Hill National Center for Biomedical Communications of the U.S. National Library of Medicine for making the malaria cell image dataset publicly available.

Funding

The authors declare that this research received no financial support from any funding agency, commercial entity or other external source.

Data availability

The dataset analysed during the current study is publicly available in the U.S. National Library of Medicine (Lister Hill National Center for Biomedical Communications) Malaria Datasets repository, https://lhncbc.nlm.nih.gov/LHC-research/LHC-projects/image-processing/malaria-datasheet.html [14], and is described in Rajaraman et al. [5] (https://doi.org/10.7717/peerj.4568). The fold assignments and out-of-fold predictions generated during this study are included in this published article as Supplementary File 1.

Code availability

The code used for the cross-validation experiments is included as Supplementary File 2.

Ethics approval and consent to participate

Not applicable. This study used only a publicly available, de-identified image dataset and involved no new data collection from human participants. The source images were collected and archived by the dataset providers under their own institutional review (NLM IRB#12972) [15].

Consent for publication

Not applicable.

Clinical trial number

Not applicable.

Competing interests

The authors declare no competing interests.

Author contributions

A.O.: Conceptualization, Methodology, Investigation, Supervision, Project administration, Writing – review & editing. R.B.A.: Validation, Formal analysis, Writing – original draft. E.M.E.M.: Software, Data curation. A.L.A.L.: Writing – review & editing. A.R.A.: Writing – review & editing. A.J.A.: Writing – review & editing. I.J.A.: Writing – review & editing. All authors read and approved the final manuscript.

References

1. World Health Organization. World malaria report 2025. Geneva: World Health Organization. 2025. https://www.who.int/teams/global-malaria-programme/reports/world-malaria-report-2025. Accessed 8 Nov 2026.
2. Talapko J, Škrlec I, Alebić T, Jukić M, Včev A. Malaria: the past and the present. Microorganisms. 2019;7(6):179. https://doi.org/10.3390/microorganisms7060179.
3. Tangpukdee N, Duangdee C, Wilairatana P, Krudsood S. Malaria diagnosis: a brief review. Korean J Parasitol. 2009;47(2):93–102. https://doi.org/10.3347/kjp.2009.47.2.93.

4. Poostchi M, Silamut K, Maude RJ, Jaeger S, Thoma G. Image analysis and machine learning for detecting malaria. Transl Res. 2018;194:36–55. https://doi.org/10.1016/j.trsl.2017.12.004.
5. Rajaraman S, Antani SK, Poostchi M, Silamut K, Hossain MA, Maude RJ, Jaeger S, Thoma GR. Pre-trained convolutional neural networks as feature extractors toward improved malaria parasite detection in thin blood smear images. PeerJ. 2018;6:e4568. https://doi.org/10.7717/peerj.4568.
6. Kapoor S, Narayanan A. Leakage and the reproducibility crisis in machine-learning-based science. Patterns. 2023;4(9):100804. https://doi.org/10.1016/j.patter.2023.100804.
7. Rajaraman S, Jaeger S, Antani SK. Performance evaluation of deep neural ensembles toward malaria parasite detection in thin-blood smear images. PeerJ. 2019;7:e6977. https://doi.org/10.7717/peerj.6977.
8. Bibin D, Nair MS, Punitha P. Malaria parasite detection from peripheral blood smear images using deep belief networks. IEEE Access. 2017;5:9099–108. https://doi.org/10.1109/ACCESS.2017.2705642.
9. Liang Z, Powell A, Ersoy I, Poostchi M, Silamut K, Palaniappan K, Guo P, Hossain MA, Antani S, Maude RJ, Huang JX, Jaeger S, Thoma G. CNN-based image analysis for malaria diagnosis. In: Proceedings of the 2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM). Shenzhen: IEEE; 2016. pp. 493–6. https://doi.org/10.1109/BIBM.2016.7822567
10. Fuhad KMF, Tuba JF, Sarker MRA, Momen S, Mohammed N, Rahman T. Deep learning based automatic malaria parasite detection from blood smear and its smartphone based application. Diagnostics. 2020;10(5):329. https://doi.org/10.3390/diagnostics10050329.
11. Masud M, Alhumyani H, Alshamrani SS, Cheikhrouhou O, Ibrahim S, Muhammad G, Hossain MS, Shorfuzzaman M. Leveraging deep learning techniques for malaria parasite detection using mobile application. Wirel Commun Mob Comput. 2020;2020:8895429. https://doi.org/10.1155/2020/8895429.
12. Yu H, Yang F, Rajaraman S, Ersoy I, Moallem G, Poostchi M, Palaniappan K, Antani S, Maude RJ, Jaeger S. Malaria Screener: a smartphone application for automated malaria screening. BMC Infect Dis. 2020;20:825. https://doi.org/10.1186/s12879-020-05453-1.
13. Chakradeo K, Delves M, Titarenko S. Malaria dataset (thin blood smear, updated) [dataset]. IEEE DataPort; 2020. https://doi.org/10.21227/mrgx-rh96
14. Lister Hill National Center for Biomedical Communications, U.S. National Library of Medicine. Malaria datasets [dataset]. https://lhncbc.nlm.nih.gov/LHC-research/LHC-projects/image-processing/malaria-datasheet.html. Accessed 8 Nov 2026.
15. Lister Hill National Center for Biomedical Communications, U.S. National Library of Medicine. Malaria Screener data repository, including patient-ID to cell mapping files. https://lhncbc.nlm.nih.gov/LHC-downloads/mobile-app.html. Accessed 8 Nov 2026.
16. 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;12:2825–30.

17. Abadi M, Barham P, Chen J, Chen Z, Davis A, Dean J et al. TensorFlow: a system for large-scale machine learning. In: Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16). Savannah: USENIX Association; 2016. pp. 265–83.
18. Chollet F et al. Keras. 2015. https://keras.io. Accessed 8 Nov 2026.
19. Harris CR, Millman KJ, van der Walt SJ, Gommers R, Virtanen P, Cournapeau D, et al. Array programming with NumPy. Nature. 2020;585(7825):357–62. https://doi.org/10.1038/s41586-020-2649-2.
20. Hunter JD. Matplotlib: a 2D graphics environment. Comput Sci Eng. 2007;9(3):90–5. https://doi.org/10.1109/MCSE.2007.55.
21. Kingma DP, Ba J. Adam: a method for stochastic optimization. In: Proceedings of the 3rd International Conference on Learning Representations (ICLR). San Diego; 2015. https://doi.org/10.48550/arXiv.1412.6980
22. Field CA, Welsh AH. Bootstrapping clustered data. J R Stat Soc Ser B Stat Methodol. 2007;69(3):369–90. https://doi.org/10.1111/j.1467-9868.2007.00593.x.
23. Brier GW. Verification of forecasts expressed in terms of probability. Mon Weather Rev. 1950;78(1):1–3. https://doi.org/10.1175/1520-0493(1950)078%3C0001:VOFEIT%3E2.0.CO;2.
24. Guo C, Pleiss G, Sun Y, Weinberger KQ. On calibration of modern neural networks. In: Proceedings of the 34th International Conference on Machine Learning. PMLR. 2017;70:1321–30.
25. Pallets Projects. Flask documentation. https://flask.palletsprojects.com/. Accessed 8 Nov 2026.
26. Wilson EB. Probable inference, the law of succession, and statistical inference. J Am Stat Assoc. 1927;22(158):209–12. https://doi.org/10.1080/01621459.1927.10502953.

Figures

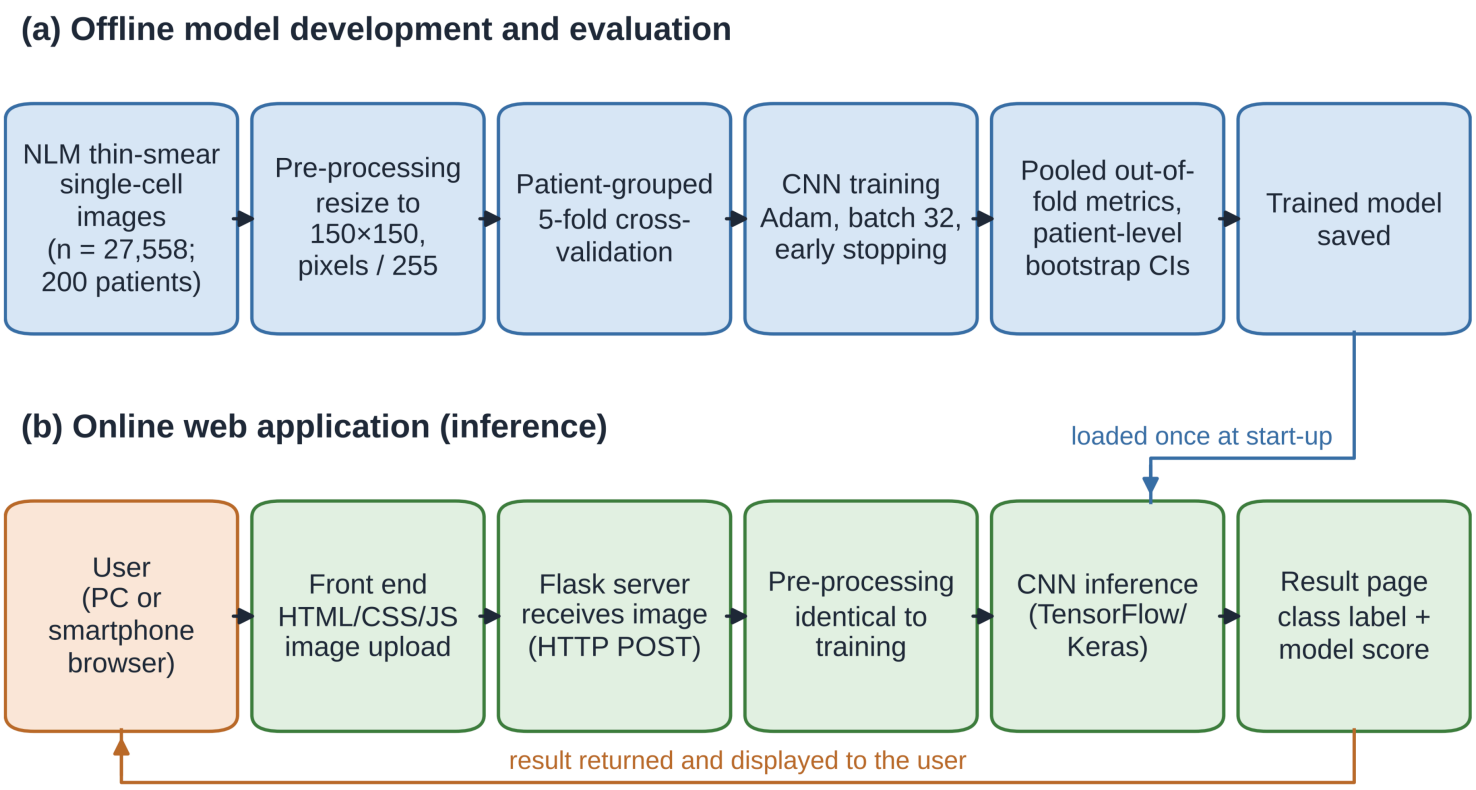

Figure 1

Overview of the proposed system. **a** Offline development and patient-level evaluation of the convolutional neural network (CNN). **b** Online use of the trained model through the web application. CI, confidence interval

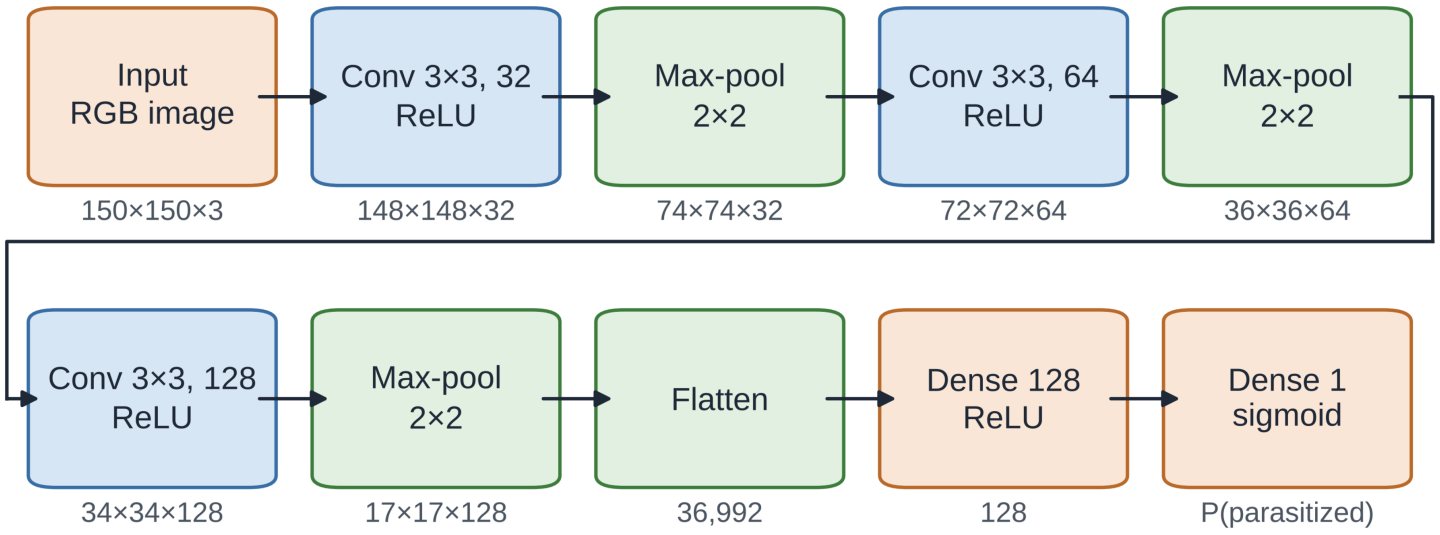

Figure 2

Architecture of the convolutional neural network. The output shape of each layer is shown below the corresponding block

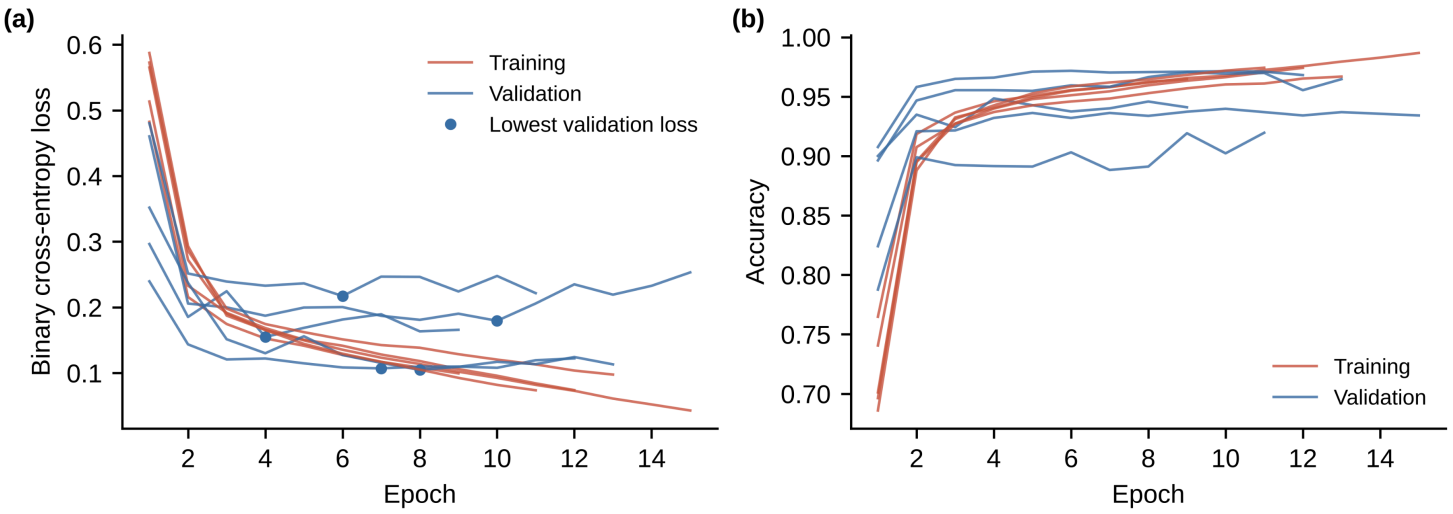

Figure 4

Learning curves of Configuration A for the five patient-level folds. **a** Training and validation loss; markers show the epoch with the lowest validation loss, whose weights were restored. **b** Training and validation accuracy

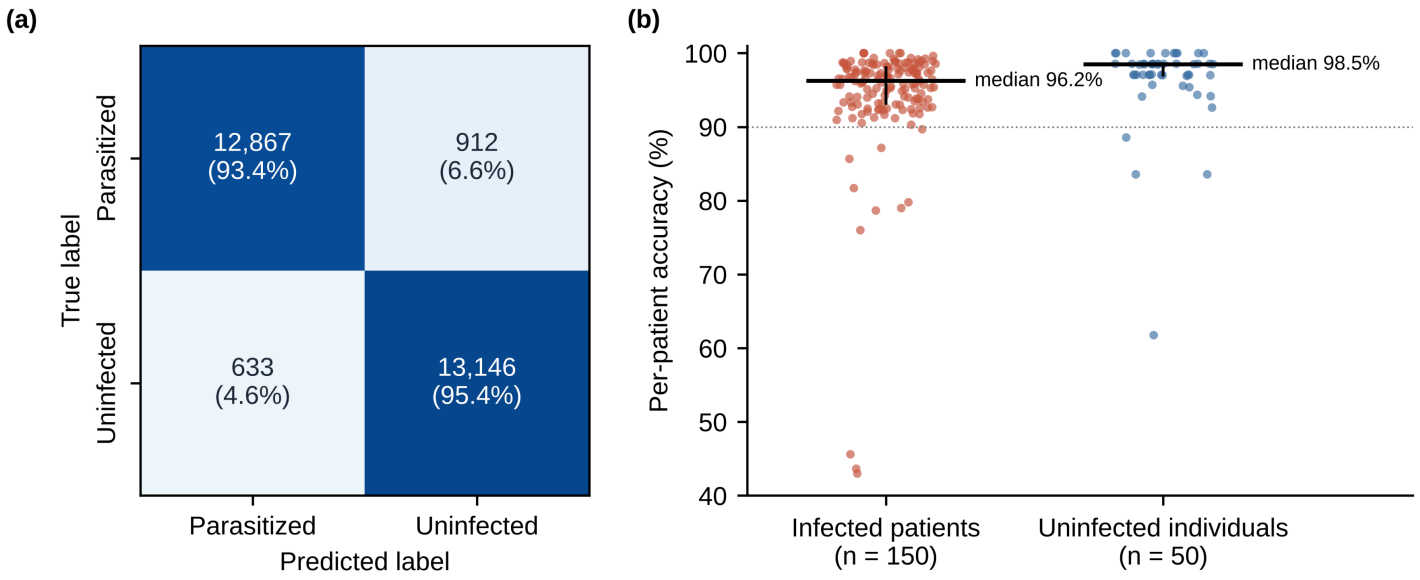

Figure 5

Errors of the primary model. **a** Pooled confusion matrix of the 27,558 out-of-fold predictions; percentages are relative to each true class. **b**Accuracy computed separately for each individual. Horizontal bars show medians and vertical bars interquartile ranges; the dotted line marks 90%