Applied Medical Informatics
Original Research Vol. 48, No. 2/2026, pp: 33-48

Performance Comparison of a Hybrid Convolutional Neural Network-Long Short-Term Memory and CNN Model for Malaria Diagnosis using Microscopic Blood Smears

Ndidiamaka P. UKEJE[1*], Oluwaseun OMOEBAMIJE[2], Mistura M. USMAN[1], Francisca OGWUELEKA[1], Chukwuemeka IFECHELOBI[3]

[1] Department of Computer Science, University of Abuja, Gwagwalada Area Council, Abuja, 900211, Nigeria
[2] Department of Civil Engineering, Nigerian Army University Biu, 1 Gombe Road, PMB 1500, Biu, 603108, Nigeria
[3] APIN Public Health Initiatives, Abuja, Plot 1551, Apo Resettlement Zone E, FCT, Abuja, 90043, Nigeria
E-mails: patienceukeje3006@gmail.com; o.oluwaseunad@gmail.com; mistura.usman@uniabuja.edu.ng; francisca.ogwueleka@uniabuja.edu.ng; cifechelobi@apin.org.ng

* Author to whom correspondence should be addressed.

Received: 1 February 2026/Accepted: 26 May 2026/ Published online: 10 June 2026

Abstract
Aim: This study evaluates and compares the performance of a lightweight Convolutional Neural Network (CNN) and a hybrid CNN–Long Short-Term Memory (CNN–LSTM) model for classifying malaria-infected and uninfected microscopic blood smear images, with emphasis on performance–efficiency trade-offs. *Methods*: A dataset of 27,558 images was preprocessed through normalization and resizing, then split into training, validation, and test sets (60:20:20). Both models were implemented using TensorFlow and Keras. The CNN comprised three convolutional layers, while the CNN–LSTM incorporated sequential learning with significantly fewer parameters. Performance was evaluated using accuracy, precision, recall, F1-score, confusion matrices, and ROC curves with AUC. *Results*: The CNN achieved 94.90% accuracy (AUC: 94.91%), demonstrating strong performance despite its simplicity. The CNN–LSTM, with a much smaller parameter size, achieved 91.42% accuracy and a higher AUC of 97.15% (95% CI: 0.97–0.98). Comparative analysis with models such as VGG16, VGG19, ResNet variants, MobileNetV2, Xception, InceptionV3, and DenseNet201 showed that the CNN performs competitively with lower computational cost. *Conclusion*: Lightweight CNN architectures can deliver performance comparable to deeper models, while CNN–LSTM offers a compact alternative with strong class separability, supporting efficient and scalable malaria diagnosis in resource-constrained settings.

Keywords*:* Malaria; Deep Learning (DL); Long Short-Term Memory Networks (LSTM); Convolutional Neural Networks (CNN)

Introduction

This study analyzes malaria, a life-threatening infectious disease caused by protozoan parasites from the Plasmodium family and primarily transmitted through the bites of Anopheles mosquitoes or contaminated needles. Globally, malaria remains a major public health challenge. In 2023, an estimated 263 million cases were reported, corresponding to an incidence of 60.4 per 1,000 population at risk, representing an increase from 252 million cases and 58.6 per 1,000 in 2022. Despite this rise in cases, mortality has declined, with an estimated 597,000 deaths and a mortality rate of 13.7 per 100,000 in 2023, down from 622,000 deaths and 14.9 per 100,000 in 2020, but the burden remains heavily concentrated in the WHO African Region, which accounted for approximately 94% of

global cases, while long-term control efforts have averted an estimated 2.2 billion cases and 12.7 million deaths since 2000 [1].

Malaria is a dreadful disease that spreads when an infected mosquito bites a human. Tiny parasites can infect mosquitoes. When a mosquito bites, it injects malaria parasites into the person's bloodstream, which can cause severe health problems, such as seizures, brain damage, troubled breathing, organ failure, and death, if untreated [2]. It is most widespread in hot and humid tropical areas of the world, affecting an estimated 241 million people worldwide each year [1]. The majority of these incidents occur in Africa and South Asia. It is widespread in third-world and developing countries and in parts of the world with warm temperatures and high humidity, especially in Africa, Central and South America, Haiti, the Dominican Republic, and other Caribbean nations, Eastern Europe, South Asia, and islands in the Central and South Pacific Oceans (Oceania) [1].

Although malaria is universal, residents in Africa are at a higher risk of infection than those in other parts of the world. Toddlers, the elderly, and pregnant women have an increased risk of malaria-related mortality. People living in poverty with limited access to healthcare are most likely to suffer complications from the disease [2].

Malaria infection occurs when an infected mosquito bites a non-infected person, thereby transferring parasites into the person's bloodstream, where they reproduce and multiply [3]. Although the infection process is immediate, malaria symptoms typically appear within 10 days to one month after infection. The symptoms vary in their presentation. Some infected people may not feel sick for up to a year after being bitten by a mosquito. Parasites can live in the human system for several years without showing any symptoms [4, 5]. Some symptoms of malaria include fever and sweating, chills, headache and muscle aches, fatigue, chest pain, breathing problems, cough, diarrhea, nausea, and vomiting. As malaria progresses in the body, it can also cause anemia and jaundice (yellowing of the skin and whitening of the eyes) [6].

Malaria parasites can be identified by examining a drop of the patient's blood under a microscope and spreading it onto a microscope slide as a 'blood smear.' Before examination, the sample is stained (most frequently with Giemsa stain) to give the parasites a distinctive appearance. This approach is the gold standard for laboratory evidence of malaria. However, it depends on the quality of the reagents, the microscope, and the medical scientist's experience, which makes this method highly subjective [7, 8]. Other conventional malaria diagnoses include rapid diagnostic tests (RDTs) and molecular techniques such as polymerase chain reaction (PCR). RDTs offer the advantage of fast and easy detection without the need for specialized equipment, making them suitable for field use; however, their sensitivity can be reduced at low parasite densities [9, 10]. PCR-based methods, on the other hand, provide high sensitivity and specificity and are effective for detecting low-level infections, but they are expensive, time-consuming, and require advanced laboratory infrastructure [11, 12]. These limitations across existing diagnostic approaches highlight the need for reliable, efficient, and scalable automated solutions.

Convolutional Neural Networks (CNNs) are particularly suited for microscopic image analysis due to their ability to automatically learn hierarchical spatial features from raw pixel data [13, 14]. In the context of malaria diagnosis, CNNs can capture subtle morphological characteristics such as cell shape distortions, parasite presence, staining variations, and texture irregularities within red blood cells [15]. Through convolutional layers, pooling operations, and nonlinear activations, the network progressively abstracts low-level features (edges, color gradients, etc.) into high-level representations that are critical for distinguishing infected from uninfected cells [16, 17]. This eliminates the need for manual feature engineering, which is often subjective and prone to inconsistency in traditional image processing approaches [18]. Therefore, recent studies have applied deep learning methods as a means to realize cheap, computationally intensive artificial neural networks for malaria detection, such as those by Ozcan et al. [19]. Their model succeeded with excellent results at the time, but being a complex model, it feeds on very large amounts of datasets with labels, such as images, text, audio, and video.

Other deep learning approaches for malaria detection have largely focused on convolutional neural networks, particularly through transfer learning using pre-trained architectures such as VGG, ResNet, and DenseNet [20-22]. These models have demonstrated high classification accuracy due to their depth and ability to learn complex feature representations [23, 24]. For instance, Rajaraman et al. [20] posited that pre-trained models such as AlexNet and VGG can achieve classification accuracies in the range of 94–96% on microscopic blood smear datasets, with improved performance observed when fine-tuning deeper layers. While their approach benefited from robust

feature extraction, it relied heavily on large model sizes and extensive fine-tuning, raising concerns about computational efficiency and deployment feasibility. Similarly, Liang et al. [21] employed deep CNN architectures and reported classification accuracies exceeding 95%, with some models approaching 97% accuracy; however, the authors noted the need for more dataset inclusion in order to enhance generalizability.

Despite these successes, CNN-based models are inherently limited in modeling sequential dependencies within feature representations, as they primarily focus on spatial feature extraction. This limitation has motivated the exploration of hybrid architectures that integrate sequential learning mechanisms. The incorporation of Long Short-Term Memory (LSTM) networks enables the model to retain and selectively update information across feature sequences, thereby capturing contextual relationships that may exist within image data [25, 26]. In the hybrid CNN–LSTM architecture, features extracted by the CNN are reshaped into sequences and passed to the LSTM, which learns dependencies among these features. This approach is particularly relevant in medical imaging, where spatial patterns may exhibit implicit structural relationships across regions of an image [27]. By combining CNN's spatial learning capability with LSTM's temporal (or sequential) memory mechanism, hybrid models are expected to achieve improved classification robustness and generalization compared to standalone CNNs [28].

Chaubey et al. [29] further highlighted the growing application of other deep learning techniques, including recurrent neural networks (RNNs) and stacked autoencoders, in healthcare image analysis. Building on this, other studies have explored hybrid architectures that combine CNNs with recurrent models, including LSTM networks, to enhance feature learning [30, 31]. Among such works is Donahue et al. [30], which introduced Long-Term Recurrent Convolutional Networks (LRCNs) and demonstrated strong performance in sequential visual tasks such as video and image classification, and Hyun et al. [32], which implemented a CNN-LSTM hybrid model for extracting audio features for acoustic scene classification of the DCASE 2016 dataset. The idea is that the CNN layers learn the spectro-temporal locality from spectrogram images. However, the direct application of such architectures to static medical images, such as blood smears, is less straightforward, as the sequential modeling capability of LSTMs may not always align with the inherently spatial nature of the data. Consequently, the integration of CNN and LSTM must be carefully justified to ensure that the added complexity translates into meaningful performance gains.

Despite these advancements, a clear gap remains in balancing model performance with computational efficiency. Existing studies emphasize improved accuracy through increasingly deep and complex architectures, with limited attention to model size, memory requirements, and deployment feasibility [33, 34]. In addition, the effectiveness of hybrid architectures such as CNN–LSTM for static medical image classification tasks remains insufficiently explored and not yet conclusively established [35, 36]. Therefore, this study evaluated and compared the performance of a lightweight CNN and a hybrid CNN–LSTM model in classifying malaria-infected and uninfected microscopic blood smear images, with the aim of examining their effectiveness and suitability for efficient, real-world deployment.

Materials and Methods

The study employed deep learning approaches to classify microscopic blood smear images as parasitized or uninfected. The overall workflow consisted of data acquisition, dataset partitioning, and model development using a Convolutional Neural Network (CNN) and a hybrid CNN–Long Short-Term Memory (CNN–LSTM) architecture.

Dataset

The dataset used in this study is the open-source Malaria Screener Dataset, obtained from the National Institutes of Health (NIH) repository [37]. It contains microscopic images of thin and thick blood smears captured using a smartphone attached to a conventional light microscope. The images were manually annotated by expert microscopists at the Mahidol-Oxford Tropical Medicine Research Unit in Bangkok, Thailand, and the dataset includes samples of Plasmodium vivax and Plasmodium falciparum, comprising 27,558 images, evenly distributed between two classes: parasitized and uninfected (13,779 images per class), ensuring class balance.

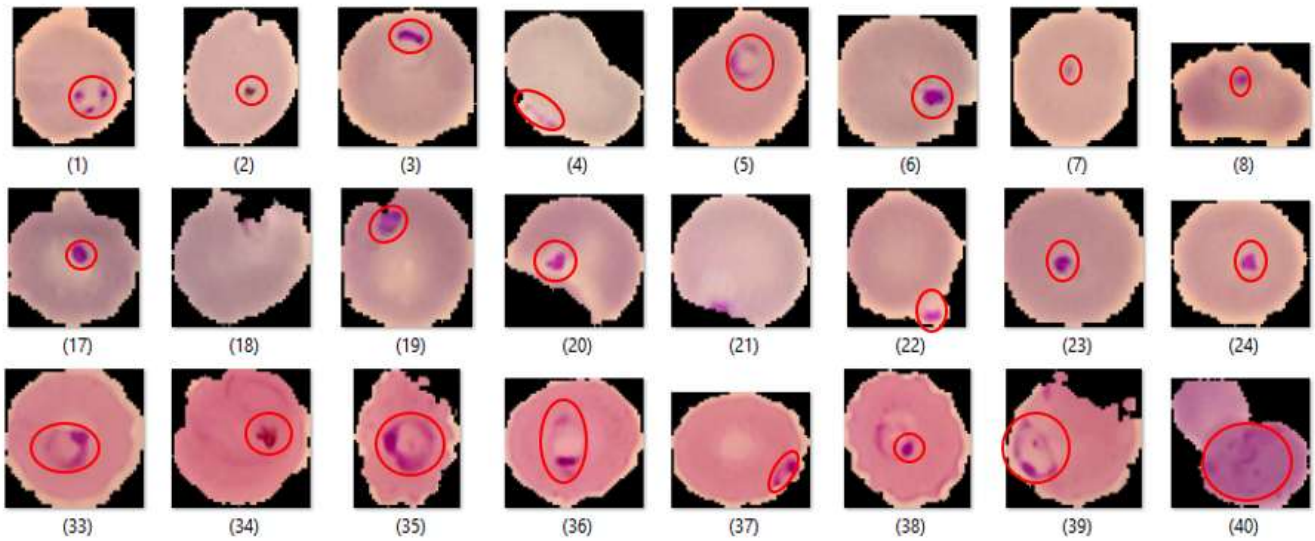


Figure 1. Annotated rings of *Plasmodium falciparum* in blood smear images.

For model development, the dataset was randomly split into training, validation, and test sets using a 60:20:20 ratio. The splitting was performed using random sampling while preserving the original class distribution across all subsets (i.e., stratified random splitting) to avoid class imbalance and ensure unbiased model evaluation. The training set was used to learn feature representations, the validation set was used for hyperparameter tuning and performance monitoring during training, and the test set, which was kept completely unseen, was then used for final model evaluation.

Data Preprocessing

To obtain optimal results, all images were preprocessed prior to being passed into the deep learning models. The preprocessing steps were designed to enforce input consistency, reduce noise, and scale pixel values to a suitable range for efficient learning. All preprocessing operations were implemented using the Open Computer Vision library (OpenCV), a widely used tool for image processing in machine learning tasks. After conversion, a portion of an image may look like this (Figure 2) to the model, that is, the portion of an image as a numerical data representation:

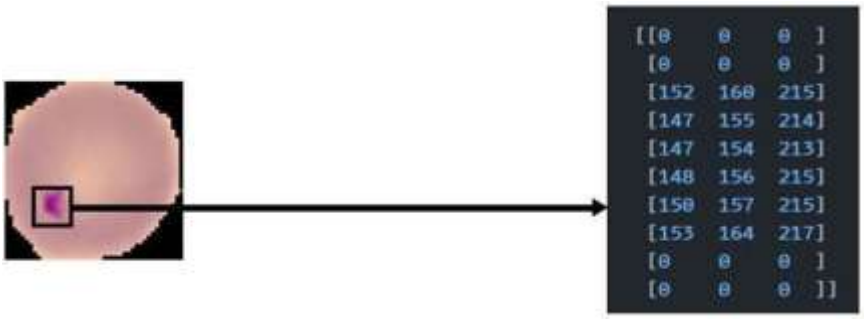


Figure 2. Portion of an Image in Numerical Data Representation.

The original images in the dataset varied in size, with dimensions ranging from 55×40 pixels to 364×240 pixels. To ensure uniformity across inputs, all images were resized to a fixed dimension of 120×120 pixels. Unlike earlier approaches that convert images to grayscale, this study retained the full RGB color information, as color features can provide additional discriminative cues for accurate classification. Thoroughly preprocessed image data would look as shown in Figure 3.

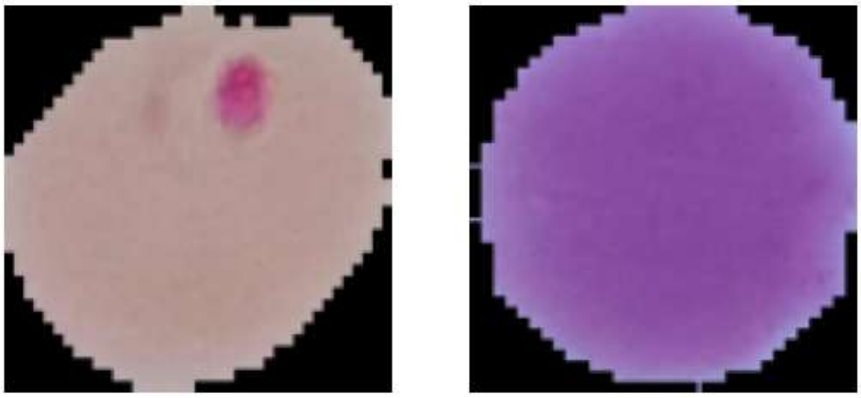


Figure 3. Preprocessed image of parasitized (left-hand image) and uninfected samples (right-hand image).

Following the resizing, the images were normalized to standardize pixel intensity values (Figure 4). Specifically, pixel values in the RGB channels, originally in the range [0, 255], were scaled to the range [0, 1] by dividing each pixel value by 255. This normalization step ensures that all input features contribute proportionately during training and improves numerical stability and convergence of the learning algorithms.

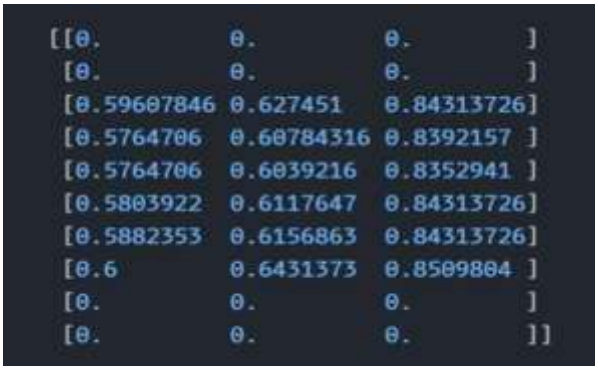


Figure 4. Array after normalization by example.

To prevent any bias arising from the order of data presentation, the dataset was randomly shuffled prior to splitting into training, validation, and test subsets. Shuffling ensures that the data distribution remains representative across subsets and avoids unintended learning patterns based on data ordering. A fixed random seed was used during shuffling to ensure reproducibility of the results, allowing the same data splits and experimental outcomes to be consistently reproduced.

Deep Learning Architectures

Convolutional neural networks (ConvNets or CNNs) are key algorithms for image recognition and classification. They perform excellently in object detection, facial recognition, and many other image-related tasks [38, 39]. Basic CNN image classification involves acquiring, processing, and classifying input images into specific categories (e.g., dogs, cats, tigers, lions, etc.) [19]. A computer treats the input image as an array of pixels, with dimensions H×W×D (H = height, W = width, and D = depth), depending on the image's resolution. For example, a 6×6×3 matrix image of RGB and a 4×4×1 matrix image of a grayscale image. Each input frame moves through a set of integrated layers with filters (or kernels), aggregations, fully connected (FC) layers, and SoftMax capabilities to identify objects with probability values between 0 and 1 [40, 42]. The model then identifies objects based on the value of the complete CNN flow parameter by processing the input image [40].

Following the input layer, the convolutional layer performs the initial feature extraction from the input image. This layer operates by applying learnable filters (kernels) to the input image matrix, enabling the network to capture spatial features such as edges, textures, and patterns while preserving pixel relationships [39, 42]. During this

process, filters slide across the image using a defined step size known as the stride. A stride of 1 shifts the filter one pixel at a time, whereas a stride of 2 results in larger movements and reduced spatial resolution [42].

To ensure proper coverage of the input image, padding is often introduced when the filter size does not perfectly align with the image dimensions. Zero-padding involves adding artificial zeros around the image borders to preserve spatial dimensions, while valid padding retains only the regions where the filter fully overlaps the input [43]. Following convolution, a nonlinear activation function is applied to introduce nonlinearity into the model. The Rectified Linear Unit (ReLU) is commonly used due to its computational efficiency and superior performance compared to alternatives such as sigmoid and tanh [40].

To further refine the extracted features and reduce computational complexity, pooling layers are introduced. Pooling, also referred to as subsampling or downsampling, reduces the spatial dimensions of feature maps while retaining the most relevant information [41, 43]. Common approaches include max pooling, which selects the largest value within a region; average pooling, which computes the mean value; and sum pooling, which aggregates all values within the region [44, 45].

After successive convolution and pooling operations, the resulting feature maps are transformed into a one-dimensional vector using a flattening operation. This vector is then passed to the fully connected layers, where high-level reasoning and classification are performed. As earliest mentioned, the Conv2D architecture enables the model to learn spatially invariant features by applying filters across the image, while MaxPooling2D reduces dimensionality and enhances robustness to small variations in the input [46, 47]. Finally, the flattened features are fed into dense layers, which map the learned representations to the output classes [48].

Dense layers connect every neuron in the subsequent layer and learn patterns from the features extracted by earlier layers through trainable weight parameters [21]. To combat overfitting, dropout randomly deactivates a portion of neurons during training, thereby forcing the network to develop more robust features by preventing co-adaptation [49, 50]. Finally, batch normalization normalizes layer activations by standardizing the output of each mini-batch, which stabilizes and enhances training by reducing internal shifts. This normalization allows for higher learning rates and introduces a slight regularization effect, working with dropout to produce models that generalize better to unseen data while training more efficiently [51].

A Long Short-Term Memory network (LSTM) is a kind of RNN that can learn long-term dependencies. Hochreiter and Schmidhuber [52] pioneered its development, and it has since been developed and popularized by many in subsequent operations. The LSTM was explicitly designed to avoid long-term dependency issues. Remembering information for a long time requires little effort once it becomes part of their basic behavior. All circular neural networks have the form of a repeating module chain of neural networks. LSTMs also have a chain-like structure; however, repeating modules have different structures. There are four such layers that interact in a special way, rather than a single neural network layer [32]. At the center of the LSTM is the state of the cells in the horizontal line that crosses the top of the figure. The state of the cell is similar to that of a conveyor belt, as it runs linearly along the entire chain with only a few linear interactions. It is very easy to flow with the unchanged information [53]. LSTMs can delete or add information to cell states, which are carefully controlled by structures called gates. The gate is an optional method for passing on information. They consist of a sigmoid neural network layer and a per-point multiplication operation [54, 55]. So far, a very typical LSTM has been outlined. However, not all LSTMs are the same. In reality, it appears that nearly every article containing LSTMs uses a slightly modified form.

As shown in the layered visual of the hybridized model in Figure 5, the LSTM layer is sandwiched between the reshape and flattening layers, which is the only difference between the CNN (standalone) and hybridized CNN-LSTM architectures. The LSTM layer processes sequential information while maintaining the memory of previous states through its specialized cell architecture with input, forget, and output gates. LSTMs excel at capturing temporal dependencies and long-range patterns in the sequence created from spatial features [31].

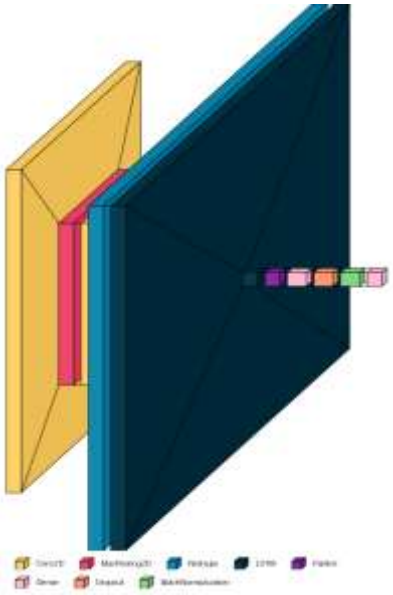


Figure 5. Proposed hybrid architecture.

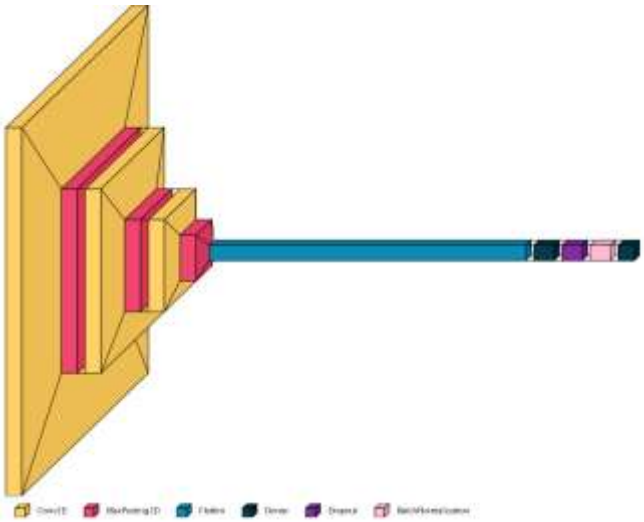


Figure 6. Proposed CNN architecture [56].

Algorithm Selection and Model Hyperparameters

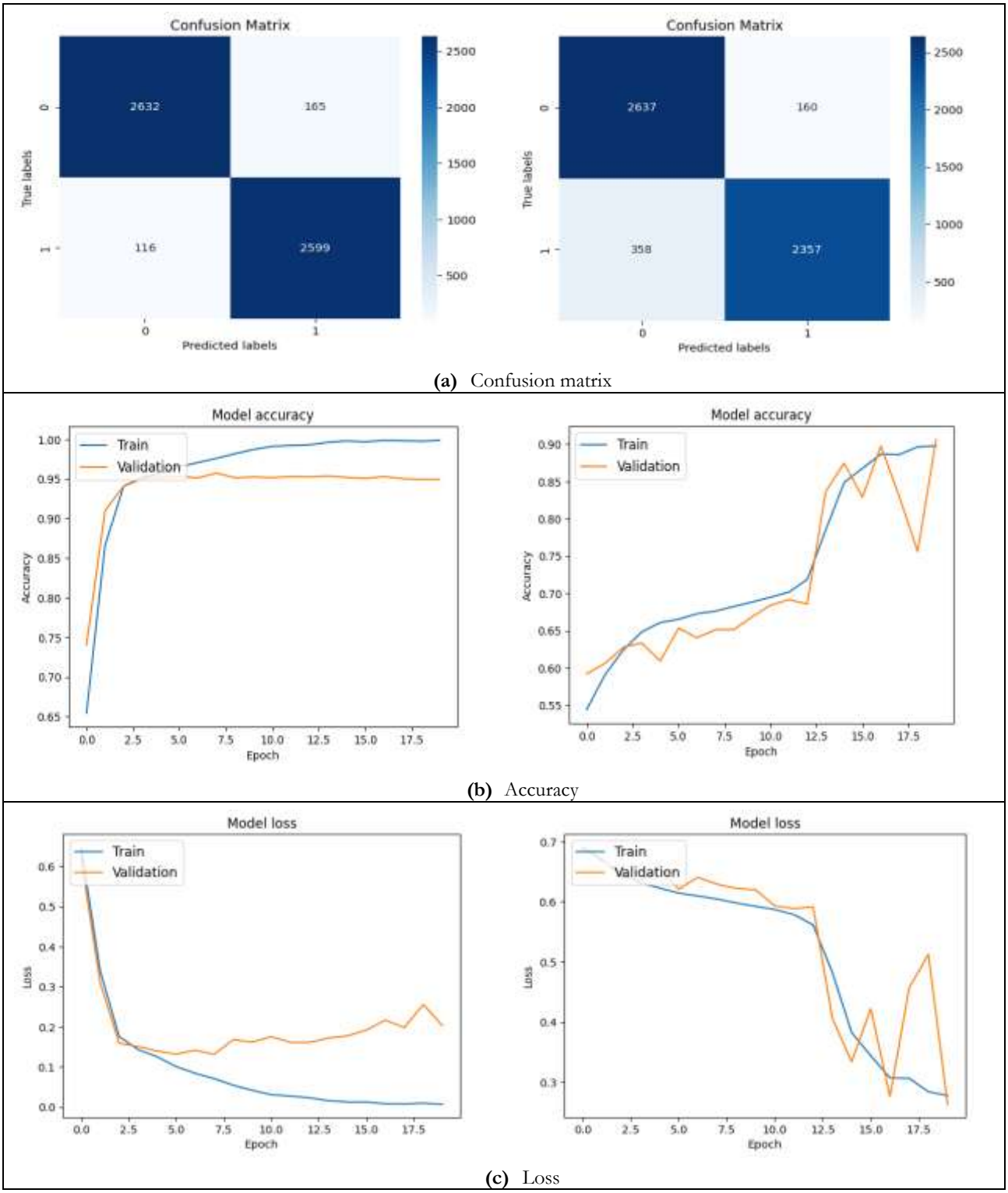
Two (2) neural network architectures were selected for this study: a Convolutional Neural Network (CNNş Figure 6) and a hybrid CNN–Long Short-Term Memory (CNN–LSTM) model. While CNNs are highly effective at extracting spatial features from image data, microscopic blood smear images contain rich color and structural information, including variations in parasite nuclei, cytoplasm, and erythrocytes, which may extend beyond simple visual patterns. To better capture these complex feature relationships, a CNN–LSTM architecture was employed, where the CNN extracts spatial and color-based representations and the LSTM models dependencies within these features, enabling the learning of both visual and structured information for improved classification performance.

Both the CNN and CNN–LSTM models were trained under the same configuration to ensure a fair comparison. The convolutional layers used 3×3 kernels with a stride of 1 and Rectified Linear Unit (ReLU) activation, while the final output layer employed a Softmax activation function for binary classification. Max-pooling with a pool size of 2×2 was applied for spatial downsampling, and the Adam optimizer with a learning rate of 0.0001. The models were trained for 20 epochs with a batch size of 64, using sparse categorical cross-entropy as the loss function.

Results

The performance of the CNN (Figure 8) and hybrid CNN–LSTM (Figure 9) models is presented in Figure 7(a, b, c, d, and e) in terms of confusion matrices, accuracy and loss curves, and ROC curves with corresponding AUC values.

CNN Model **CNN-LSTM**



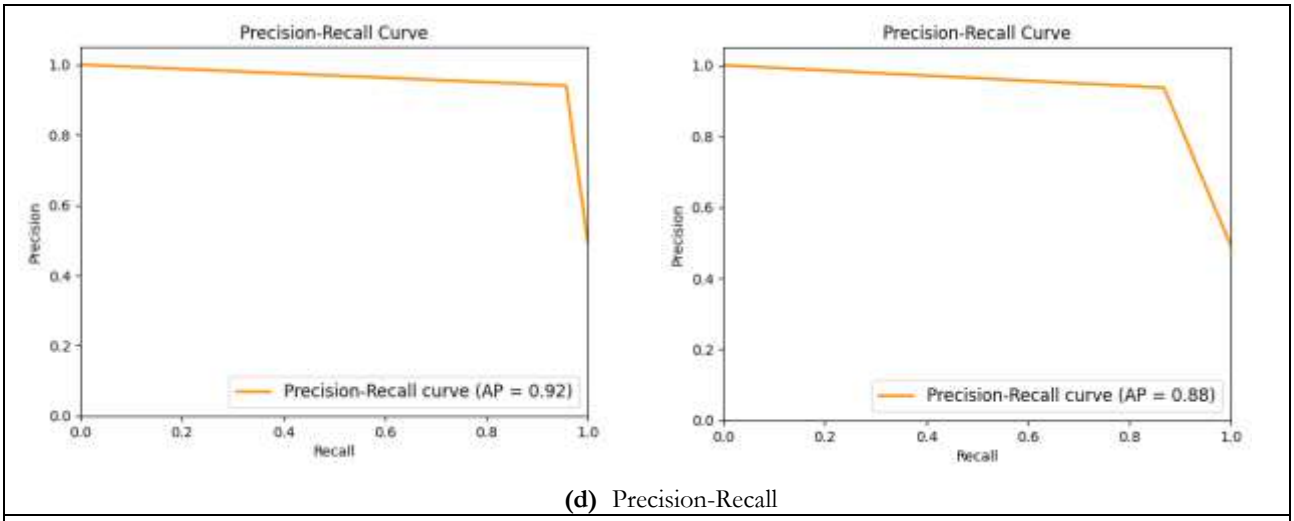
(a) Confusion matrix

(b) Accuracy

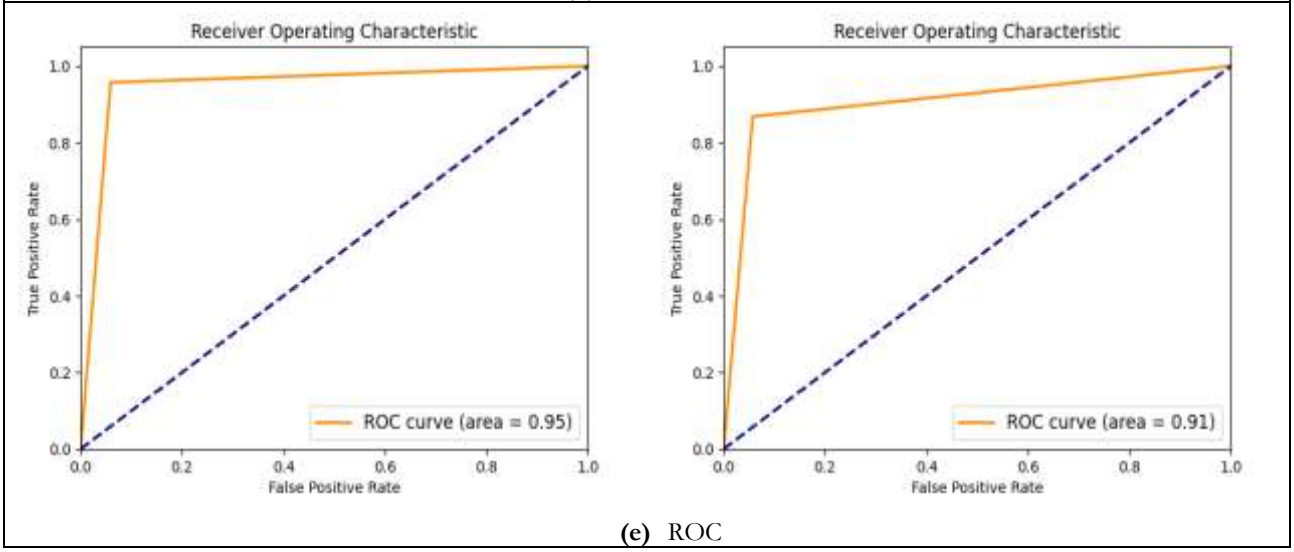
(c) Loss

Figure 7. Performance comparison of the CNN and hybrid models: (a) Confusion matrix; (b) Accuracy; (c) Loss

CNN Model **CNN-LSTM**



(d) Precision-Recall



(e) ROC

Figure 7. (continuation) Performance comparison of the CNN and hybrid models: (d) Precision-Recall; (e) ROC

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 128, 128, 64)	640
max_pooling2d (MaxPooling2D)	(None, 60, 60, 64)	0
conv2d_1 (Conv2D)	(None, 60, 60, 64)	36,928
max_pooling2d_1 (MaxPooling2D)	(None, 30, 30, 64)	0
conv2d_2 (Conv2D)	(None, 30, 30, 128)	73,856
max_pooling2d_2 (MaxPooling2D)	(None, 15, 15, 128)	0
flatten (Flatten)	(None, 28800)	0
dense (Dense)	(None, 256)	7,373,056
dropout (Dropout)	(None, 256)	0
batch_normalization (BatchNormalization)	(None, 256)	1,024
dense_1 (Dense)	(None, 2)	514

Total params: 7,486,018 (28.56 MB)
Trainable params: 7,485,506 (28.55 MB)
Non-trainable params: 512 (2.00 KB)

Figure 8. Parameters for the CNN model.

Layer (type)	Output Shape	Param #
conv2d_3 (Conv2D)	(None, 120, 120, 64)	640
max_pooling2d_3 (MaxPooling2D)	(None, 60, 60, 64)	0
conv2d_4 (Conv2D)	(None, 60, 60, 64)	36,928
max_pooling2d_4 (MaxPooling2D)	(None, 30, 30, 64)	0
conv2d_5 (Conv2D)	(None, 30, 30, 128)	73,856
max_pooling2d_5 (MaxPooling2D)	(None, 15, 15, 128)	0
reshape_1 (Reshape)	(None, 225, 128)	0
lstm_2 (LSTM)	(None, 225, 128)	131,584
lstm_3 (LSTM)	(None, 64)	49,408
flatten_1 (Flatten)	(None, 64)	0
dense_2 (Dense)	(None, 256)	16,640
dropout_1 (Dropout)	(None, 256)	0
batch_normalization_1 (BatchNormalization)	(None, 256)	1,024
dense_3 (Dense)	(None, 2)	514

Total params: 310,594 (1.18 MB)
Trainable params: 310,082 (1.18 MB)
Non-trainable params: 512 (2.00 KB)

Figure 9. Parameters for CNN-LSTM model.

Discussion

Performance Analysis of CNN and CNN–LSTM Models

A closer look at the results reveals a balance between model complexity and predictive performance. The standalone CNN model, built with only three convolutional layers, achieved an accuracy of 94.90% with an AUC of 94.91% (Table 1). Considering its relatively modest architecture (7.49 million parameters), this level of performance is notable, particularly for a medical image classification task that typically benefits from deeper and more complex networks [17]. The model demonstrates a strong ability to extract and generalize discriminative spatial features from microscopic blood smear images, suggesting that the morphological differences between parasitized and uninfected cells can be effectively captured even with a lightweight design.

Table 1. Comparative analysis with state-of-the-art models

Model	**Precision**	**Recall**	**Accuracy**	**F1-score**	**AUC**
CNN (our model)	94.03	95.73	94.90	94.87	94.91
CNN_LSTM (our model)	88.03	95.58	91.42	91.65	97.15
VGG16 [57]	95.88	94.22	95.16	95.04	98.81
VGG19 [57]	94.49	96.02	95.28	95.25	98.59
ResNet50V2 [58]	90.40	96.06	93.03	93.14	97.88

MobileNetV2 [34]	93.63	96.39	94.99	94.99	98.73
Xception [33]	93.35	94.03	93.76	93.69	98.40
ResNet152V2 [58]	92.75	94.66	93.72	93.69	98.01
InceptionV3 [59]	88.66	93.30	90.82	90.92	96.92
DenseNet201 [60]	92.64	96.94	94.70	94.74	98.68

In contrast, the hybrid CNN–LSTM model presents a different trade-off. With only 310,594 parameters, it is significantly more compact, yet it achieved an AUC of 97.15%, surpassing the CNN in this regard, although with a lower overall accuracy of 91.42%. This indicates that while the CNN–LSTM model may not classify as many samples correctly in absolute terms, it exhibits a stronger ability to distinguish between classes across varying decision thresholds. The inclusion of the LSTM layer appears to enhance the model's sensitivity to feature relationships, potentially capturing dependencies within the extracted feature maps that a purely convolutional architecture might overlook [31].

However, the expected advantage of the hybrid approach is not fully realized in terms of classification accuracy. One plausible explanation is that the features extracted from individual blood smear images may not inherently possess strong sequential dependencies, which limits the contribution of the LSTM component. The reduced parameter size of the CNN–LSTM model, while beneficial for computational efficiency, may constrain its representational capacity, leading to a slight drop in overall accuracy.

The CNN–LSTM hybrid model shares identical convolutional blocks but introduces two LSTM layers (with 131,584 and 49,408 parameters), as shown in Figure 9 following the reshape layer. The recurrent layers altered the model's learning dynamics during the 20 epochs. The training progression revealed that the hybrid model initially followed similar learning trajectories to the CNN but began to diverge around the middle epochs. The LSTM layers developed a progressive bias toward negative predictions, explaining the disproportionate false negative rate observed in the final model [32].

The reshape layer between the convolutional and LSTM components represents a critical juncture in the hybrid architecture. Converting spatial feature maps into sequential input for LSTMs requires flattening two dimensions into one, potentially losing important spatial relationships that the CNN has extracted [25, 26]. This architectural choice reframes a spatial problem as a sequential one, creating a mismatch that becomes more pronounced as training progresses through the epochs. The batch normalization layer included in both models helps maintain stable training [51].

The final dense layers in both models serve as classifiers operating on the extracted features; however, their effectiveness ultimately depends on the quality of these features. After 20 epochs, the CNN's dense layers operated on spatially coherent features directly relevant to the classification task, whereas the hybrid model's dense layers received features that had undergone potentially disruptive sequential processing. This fundamental difference in information flow through the network layers explains much of the performance gap, despite both models having sufficient training iterations.

Taken together, these findings indicate that although the CNN model offers higher overall classification accuracy, the CNN–LSTM model remains a viable alternative, achieving strong class separability with substantially lower computational expenses.

Comparison with State-of-the-Art Models

When compared with established deep learning architectures, the results further highlight the efficiency of the proposed models. Networks such as VGG16 and VGG19 achieved slightly higher performance metrics, with VGG16 recording the highest AUC of 98.81% and strong scores across all evaluation metrics. These models are

known for their depth and large parameter sizes, which enable them to learn highly complex feature representations. However, this comes at the cost of increased computational demand and memory usage.

What stands out in this study is how closely the proposed CNN model approaches these state-of-the-art results despite its significantly simpler architecture. Achieving an accuracy of 94.90% places it within a narrow margin of deeper networks like VGG16 (95.16%) and VGG19 (95.28%), suggesting that increasing architectural complexity does not necessarily yield proportionate performance gains for this task. This reinforces the idea that the discriminative features required for malaria detection in blood smear images may be relatively well-defined and do not always require very deep feature hierarchies.

Similarly, the CNN–LSTM model, despite its much smaller size, maintains competitive performance across several metrics. Its AUC of 97.15% (95% CI: 0.97–0.98) further highlights its strong class separability, exceeding that of InceptionV3 (96.92%) among the deeper architectures evaluated. This is particularly important in medical diagnosis, where the ability to distinguish between classes reliably across thresholds can be as critical as raw accuracy.

Collectively, these results have practical significance for real-world deployment, especially in low-resource settings where access to high-end computational infrastructure is limited. These findings open avenues for further optimization, including pruning, quantization, and attention mechanisms that could push lightweight architectures even closer to the performance ceiling of deeper models while retaining their computational advantages, ultimately supporting scalable and accessible malaria diagnosis systems.

Future Research

Future research could explore the integration of attention mechanisms to enhance feature prioritization and improve classification performance, particularly for challenging or low-quality smear images. Expanding the dataset to include more diverse samples, variations in staining conditions, and multi-species malaria infections would further strengthen model generalization. Additionally, investigating model optimization techniques such as pruning, quantization, or knowledge distillation could yield even more efficient architectures suitable for real-time deployment on low-power devices. Finally, incorporating explainable AI methods would improve model interpretability and support clinical adoption by providing insights into the decision-making process.

Conclusion

The results show that a shallow CNN can achieve accuracy comparable to significantly deeper state-of-the-art architectures, while the CNN–LSTM model, with a substantially smaller parameter size, maintains competitive performance and strong class separability. Our findings suggest that increasing model complexity does not necessarily yield proportional performance gains for this task and that well-designed, efficient architectures can provide a practical and scalable solution for automated malaria diagnosis, particularly in resource-constrained settings.

List of Abbreviations: CNN - convolutional neural networks; LSTM - long short-term memory; CDC - Center for Disease Control; RNN - recurrent neural networks; WHO - World Health Organization; DBN - deep belief networks; DBM - deep Boltzmann machine; FC - fully connected; ReLU - rectified linear unit; RGB - red, green, and blue; LHNCBC - Lister Hill National Center for Biomedical Communications; IGMS - iterative global minimum screening; OpenCV - Open Computer Vision; AUC - area under the curve; ROC - receiver operating characteristics.

Author Contributions: Conceptualization: N.P.U.; Formal Analysis: N.P.U., O.O.; Methodology: N.P.U., O.O.; Project Administration: N.P.U., O.O.; Supervision: MMU, FO, C.I.; Writing: N.P.U., O.O.; Writing - review & editing: O.O., and C.I. All authors read and approved the final manuscript.

Conflict of Interest: The authors declare no conflicts of interest.

Funding: This research received no funding.

Ethics Statement: Not applicable**.**

Data Availability Statement: The dataset used in this study is publicly available from the Lister Hill National Center for Biomedical Communications (LHNCBC) at https://lhncbc.nlm.nih.gov/LHC-downloads/downloads.html#malaria-datasets (last accessed 27 March 2026).

References

1. United Nations Office for Disaster Risk Reduction (UNDRR). Malaria [Internet]. PreventionWeb UNDRR. 2025 [cited 2026 Mar 27]. Available from: https://www.undrr.org/terms/hips/BI0219
2. George AM, Ansumana R, De Souza DK, Niyas VKM, Zumla A, Bockarie MJ. Climate change and the rising incidence of vector-borne diseases globally. International Journal of Infectious Diseases [Internet]. 2023 ;139:143–5. https://doi.org/10.1016/j.ijid.2023.12.004.
3. Mayo Clinic. Malaria Transmission Cycle [Internet]. Mayo Clinic. Available from: https://www.mayoclinic.org/diseases-conditions/malaria/multimedia/malaria-transmission-cycle/img-20006373
4. Mkali HR, Reaves EJ, Lalji SM, Al-Mafazy AW, Joseph JJ, Ali AS, et al. Risk factors associated with malaria infection identified through reactive case detection in Zanzibar, 2012–2019. Malaria Journal [Internet]. 2021;20(1):485. https://doi.org/10.1186/s12936-021-04025-1.
5. Haldar T. Malaria: Biology, Disease, and Control—A Comprehensive Overview. International Journal of Advanced Research. 2025 ;13(05):451–6. https://doi.org/10.21474/ijar01/20920.
6. Bria YP, Yeh CH, Bedingfield S. Significant symptoms and non-symptom-related factors for malaria diagnosis in endemic regions of Indonesia. Research Square. 2020 Feb 12; Available from: https://doi.org/10.21203/rs.2.23268/v1.
7. ICDC - DPDX - diagnostic procedures - blood specimens [Internet]. [cited 2026 April 2]. Available from: https://www.cdc.gov/dpdx/diagnosticprocedures/blood/specimenproc.html
8. Mbanefo A, Kumar N. Evaluation of malaria diagnostic methods as a key for successful control and elimination programs. Tropical Medicine and Infectious Disease. 2020;5(2):102. https://doi.org/10.3390/tropicalmed5020102.
9. Programme GM. Malaria rapid diagnostic test performance. Results of WHO product testing of malaria RDTs: Round 8 (2016-2018) [Internet]. 2018 [cited 2026 Apr 2]. Available from: https://www.who.int/publications/i/item/9789241514965.
10. Moody A. Rapid diagnostic tests for malaria parasites. Clinical Microbiology Reviews. 2002;15(1):66–78. https://doi.org/10.1128/cmr.15.1.66-78.2002.
11. Snounou G, Viriyakosol S, Zhu XP, Jarra W, Pinheiro L, Rosario VED, et al. High sensitivity of detection of human malaria parasites by the use of nested polymerase chain reaction. Molecular and Biochemical Parasitology. 1993;61(2):315–20. Available from: https://doi.org/10.1016/0166-6851(93)90077-b.
12. Bharti AR, Letendre SL, Patra KP, Vinetz JM, Smith DM. Malaria diagnosis by a polymerase chain reaction–based assay using a pooling strategy. American Journal of Tropical Medicine and Hygiene. 2009;81(5):754–7. https://doi.org/10.4269/ajtmh.2009.09-0274.
13. Maqsood A, Farid MS, Khan MH, Grzegorzek M. Deep malaria parasite detection in thin blood smear microscopic images. Applied Sciences. 2021;11(5):2284. https://doi.org/10.3390/app11052284.
14. Jameela T, Athota K, Singh N, Gunjan VK, Kahali S. Deep learning and transfer learning for malaria detection. Computational Intelligence and Neuroscience. 2022;2022:1–14. https://doi.org/10.1155/2022/2221728.
15. Minarno AE, Izzah TN, Munarko Y, Basuki S. Classification of malaria using convolutional neural network method on microscopic image of blood smear. JOIV International Journal on Informatics Visualization. 2024;8(3):1469. https://doi.org/10.62527/joiv.8.3.2154.
16. Shekar G, Revathy S, Goud EK. Malaria Detection using Deep Learning. 2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184). 2020, pp. 746–50. Available from: https://doi.org/10.1109/icoei48184.2020.9143023

17. Joshi AM, Das AK, Dhal S. Deep learning based approach for malaria detection in blood cell images. In: 2020 IEEE Region 10 Conference (TENCON); 2020 Nov; Osaka, Japan. pp. 241–246. https://doi.org/10.1109/TENCON50793.2020.9293753.
18. Ghosh H, Rahat IS, Ravindra JVR, J B, Khan MAU, Somasekar J. Convolutional neural networks in malaria diagnosis: A study on cell image classification. EAI Endorsed Transactions on Pervasive Health and Technology. 2024;10:11p. http://doi.org/10.4108/eetpht.10.5551.
19. Ozcan T. Applications of Deep Learning Techniques in Healthcare Systems: a review. Journal of Clinical Practice and Research. 2024;46(6):527–36. http://dx.doi.org/10.14744/cpr.2024.25381.
20. Rajaraman S, Antani SK, Poostchi M, Silamut K, Hossain MdA, Maude RJ, et al. 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.
21. Liang Z, Powell A, Ersoy I, Poostchi M, Silamut K, Palaniappan K, et al. CNN-based image analysis for malaria diagnosis. In: 2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM). Shenzhen, China; 2016, pp. 493–496. https://doi.org/10.1109/BIBM.2016.7822567.
22. Ramos-Briceño DA, Flammia-D'Aleo A, Fernández-López G, Carrión-Nessi FS, Forero-Peña DA. Deep learning-based malaria parasite detection: convolutional neural networks model for accurate species identification of Plasmodium falciparum and Plasmodium vivax. Scientific Reports. 2025;15(1):3746. https://doi.org/10.1038/s41598-025-87979-5.
23. Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. Communications of the ACM. 2017;60(6):84–90. https://doi.org/10.1145/3065386.
24. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Las Vegas, NV, USA; 2016, pp. 770–8. https://doi.org/10.1109/CVPR.2016.90.
25. Salehin I, Islam MdS, Amin N, Baten MdA, Noman SM, Saifuzzaman M, et al. Real-Time Medical Image Classification with ML Framework and Dedicated CNN–LSTM Architecture. Journal of Sensors. 2023;1:3717035. https://doi.org/10.1155/2023/3717035.
26. Hayat MdT, Allawi YM, Alamro W, Sultan SM, Abadleh A, Kang H, et al. A hybrid convolutional neural network–long short-term memory (CNN–LSTM)–attention model architecture for precise medical image analysis and disease diagnosis. Diagnostics. 2025;15(21):2673. https://doi.org/10.3390/diagnostics15212673.
27. Wang X. Application of multimodal fusion in automatic image classification: Combining CNN and RNN models. Intelligent Decision Technologies. 2024;19(2):928–42. https://doi.org/10.1177/18724981241299605.
28. Mehta S, Kundra D. A robust CNN-LSTM framework for sensitive and specific brain tumor classifications. In: 2025 International Conference on Automation and Computation (AUTOCOM); Dehradun, India. 2025, pp. 1622–6. https://doi.org/10.1109/AUTOCOM64127.2025.10956792.
29. Chaubey NK, Jayanthi P. Disease diagnosis and treatment using deep learning algorithms for the healthcare system. In: Advances in medical technologies and clinical practice book series [Internet]. 2020. p. 99–114. https://doi.org/10.4018/978-1-7998-2101-4.ch007
30. Donahue J, Hendricks LA, Rohrbach M, Venugopalan S, Guadarrama S, Saenko K, et al. Long-Term recurrent convolutional networks for visual recognition and description. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2016;39(4):677–91. https://doi.org/10.1109/tpami.2016.2599174.
31. Aditi, Nagda MK, Poovammal E. Image Classification using a Hybrid LSTM-CNN Deep Neural Network. International Journal of Engineering and Advanced Technology. 2019;8(6):1342–8. https://doi.org/10.35940/ijeat.f8602.088619.
32. Bae SH, Choi I, Kim NS. Acoustic Scene Classification Using Parallel Combination of LSTM and CNN. In Detection and Classification of Acoustic Scenes and Events 2016; 3 September 2016, Budapest, Hungary. 2016, 5 pages. Available from: https://dcase.community/documents/workshop2016/proceedings/Bae-DCASE2016workshop.pdf

33. Chollet F. Xception: deep learning with depthwise separable convolutions. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Honolulu, HI, USA; 2017, pp. 1800–7. https://doi.org/10.1109/CVPR.2017.195.
34. Sandler M, Howard A, Zhu M, Zhmoginov A, Chen L. MobileNetV2: inverted residuals and linear bottlenecks. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Salt Lake City, UT, USA; 2018, pp. 4510–20. https://doi.org/10.1109/CVPR.2018.00474.
35. Hou L, et al. Automatic histopathology image analysis with CNNs. In: 2016 New York Scientific Data Summit (NYSDS). New York, NY, USA; 2016. p. 1–6. doi:10.1109/NYSDS.2016.7747812.
36. Litjens G, Kooi T, Bejnordi BE, Setio AAA, Ciompi F, Ghafoorian M, et al. A survey on deep learning in medical image analysis. Medical Image Analysis. 2017;42:60–88. https://doi.org/10.1016/j.media.2017.07.005.
37. Lister Hill National Center for Biomedical Communications. NIH malaria dataset [Internet]. Available from: https://lhncbc.nlm.nih.gov/LHC-downloads/downloads.html#malaria-datasets. (accessed 1 April 2026)
38. Vijayalakshmi A, Rajesh K. Deep learning approach to detect malaria from microscopic images. Multimedia Tools and Applications. 2019;79(21–22):15297–317. https://doi.org/10.1007/s11042-019-7162-y.
39. Chen D, Liu S, Kingsbury P, Sohn S, Storlie CB, Habermann EB, et al. Deep learning and alternative learning strategies for retrospective real-world clinical data. Npj Digital Medicine. 2019;2(1):43. https://doi.org/10.1038/s41746-019-0122-0.
40. Yamashita R, Nishio M, Gian RK DO, Togashi K. Convolutional neural networks: an overview and application in radiology. Insights Into Imaging. 2018;9(4):611–29. https://doi.org/10.1007/s13244-018-0639-9.
41. Brownle J. A tour of machine learning algorithms [Internet]. 2019. Available from: https://machinelearningmastery.com/a-tour-of-machine-learning-algorithms (accessed 1 April 2026)
42. Poostchi M, Silamut K, Maude RJ, Jaeger S, Thoma G. Image analysis and machine learning for detecting malaria. Translational Research. 2018;194:36–55. https://doi.org/10.1016/j.trsl.2017.12.004.
43. Saha S. A comprehensive guide to convolutional neural networks [Internet]. A comprehensive guide to convolutional neural networks. 2018. Available from: https://ise.ncsu.edu/wp-content/uploads/sites/9/2022/08/A-Comprehensive-Guide-to-Convolutional-Neural-Networks-%E2%80%94-the-ELI5-way-_-by-Sumit-Saha-_-Towards-Data-Science.pdf (accessed 1 April 2026)
44. Lee H, Song J. Introduction to convolutional neural network using Keras; an understanding from a statistician. Communications for Statistical Applications and Methods. 2019;26(6):591–610. https://doi.org/10.29220/csam.2019.26.6.591.
45. Marsola TC, Lorena AC. Meteor Detection Using Deep Convolutional Neural Networks. Anais Do 14o Simpósio Brasileiro De Automação Inteligente. 2019, pp. 19-24. https://doi.org/10.17648/sbai-2019-112456.
46. Montalbo FJP, Alon AS. Empirical Analysis of a Fine-Tuned Deep Convolutional Model in Classifying and Detecting Malaria Parasites from Blood Smears. KSII Transactions on Internet and Information Systems. 2021;15(1):147–65. https://doi.org/10.3837/tiis.2021.01.009.
47. Ansari A, Ogunfunmi T. A multi-stride convolution acceleration algorithm for CNNs. In: 2024 IEEE International Symposium on Circuits and Systems (ISCAS); 2024; Singapore, Singapore. p. 1–5. doi:10.1109/ISCAS58744.2024.10557906
48. Jain N, Chauhan A, Tripathi P, Moosa SB, Aggarwal P, Oznacar B. Cell image analysis for malaria detection using deep convolutional network. Intelligent Decision Technologies. 2020;14(1):55–65. https://doi.org/10.3233/idt-190079.
49. Militante SV. Malaria disease recognition through adaptive deep learning models of convolutional neural network. In: 2019 IEEE 6th International Conference on Engineering Technologies and Applied Sciences (ICETAS). Kuala Lumpur, Malaysia; 2019, pp. 1–6. https://doi.org/10.1109/ICETAS48360.2019.9117446.
50. Prabhu R. Understanding of convolutional neural network (CNN)—deep learning [Internet]. Available from: https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148 (accessed 2 April 2026).

51. Anwar SM, Majid M, Qayyum A, Awais M, Alnowami M, Khan MK. Medical Image Analysis using Convolutional Neural Networks: A Review. Journal of Medical Systems 2018;42(11):226. https://doi.org/10.1007/s10916-018-1088-1.
52. Hochreiter S, Schmidhuber J. Long Short-Term Memory. Neural Computation. 1997;9(8):1735–80. https://doi.org/10.1162/neco.1997.9.8.1735.
53. Amir M. A complete guide to understand long short-term memory (LSTM) networks [Internet]. A complete guide to understand long short-term memory (LSTM) networks. 2017. Available from: http://www.sefidian.com/2019/08/15 (accessed 2 April 2026).
54. Saxena S. Introduction to long short-term memory (LSTM) [Internet]. Introduction to long short-term memory (LSTM). 2021. Available from: https://www.analyticsvidhya.com/blog/2021/03/introduction-to-long-short-term-memory-lstm/ (accessed 2 April 2026).
55. Oinkina H. Understanding LSTM networks [Internet]. Understanding LSTM Networks. Available from: https://colah.github.io/posts/2015-08-Understanding-LSTMs/ (accessed 2 April 2026).
56. Omoebamije O, Omoniyi TM, Musa A, Duna S. An improved deep learning convolutional neural network for crack detection based on UAV images. Innovative Infrastructure Solutions. 2023;8(9):236. https://doi.org/10.1007/s41062-023-01209-3.
57. Simonyan K, Zisserman A. Very deep convolutional networks for Large-Scale image recognition. arXiv (Cornell University) [Internet]. 2014 Sep 4; Available from: http://arxiv.org/abs/1409.1556
58. He K, Zhang X, Ren S, Sun J. Identity mappings in deep residual networks. In: Lecture notes in computer science [Internet]. 2016. p. 630–45. Available from: https://doi.org/10.1007/978-3-319-46493-0_38
59. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z. Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2016. p. 2818–2826. doi:10.1109/CVPR.2016.308
60. Huang G, Liu Z, Van Der Maaten L, Weinberger KQ. Densely connected convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2017. p. 2261–2269. doi:10.1109/CVPR.2017.243.