IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 17 August 2022, accepted 7 September 2022, date of publication 12 September 2022, date of current version 20 September 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3206266

RESEARCH ARTICLE

Classifying Parasitized and Uninfected Malaria Red Blood Cells Using Convolutional-Recurrent Neural Networks

ADÁN ANTONIO ALONSO-RAMÍREZ[1], TAT'Y MWATA-VELU[2], CARLOS HUGO GARCÍA-CAPULÍN[2], HORACIO ROSTRO-GONZÁLEZ[2], JUAN PRADO-OLIVAREZ[1], MARCOS GUTIÉRREZ-LÓPEZ[3], AND ALEJANDRO ISRAEL BARRANCO-GUTIÉRREZ[1], (Member, IEEE)

[1]Tecnológico Nacional de México en Celaya, TecNM-Celaya, Guanajuato, Celaya 38010, Mexico
[2]Departamento de Electrónica, Universidad de Guanajuato, Palo Blanco, Salamanca 36885, Mexico
[3]Tecnológico Nacional de México en Morelia, TecNM-Morelia, Morelia, Michoacán 58120, Mexico

Corresponding author: Alejandro Israel Barranco-Gutiérrez (israel.barranco@itcelaya.edu.mx)

This work was supported in part by the Mexican National Council of Science and Technology Consejo Nacional de Ciencia y Tecnología (CONACYT), Spanish, under Grant number 725022, and in part by the Tecnológico Nacional de México (TecNM) en Celaya under Grant number D2203002. The work of Horacio Rostro-Gonzalez was supported by the Engineering Division of the University of Guanajuato.

ABSTRACT This work aims to classify malaria infected red blood cells from those uninfected using two deep learning approaches. Plasmodium parasite transmitted by a female anopheles's mosquitoes bite is the main cause of malaria. Commonly, Microbiological analyses by a microscope allows detecting cells infected from a blood sample, followed by a specialist interpretation of results to conclude the diagnosis process. Taking advantage of efficient deep learning approaches applied in computer vision field, the present framework proposes two deep learning architecture based on Convolutional-Recurrent neural Networks to detect accurately malaria infected cells. The first one implements a Convolutional Long Short-Term Memory while the second uses a Convolutional Bidirectional Long Short-Term Memory architecture. A malaria's public dataset consisting of parasitized and uninfected red blood cell images was used for training and testing the proposed models. The methods developed in this work achieved an accuracy of 99.89 % in the detection of malaria-infected red blood cells, without preprocessing data.

INDEX TERMS Bidirectional long short-term memory (BiLSTM), convolutional neural network (CNN), long short-term memory (LSTM), malaria dataset.

I. INTRODUCTION

Malaria is caused by the bites of female Anopheles mosquitoes infected with protozoan parasites. Malaria is essentially a tropical disease since mosquitoes are produced in large numbers in humid environments. Recent global statistics illustrate some 229 million cases of malaria, with 1,700 deaths per day, of which 96% lived in Africa [1]. The most affected by malaria are children under 5 years old, whom represents an 80 % mortality rate in the region according to [1].

Among other malaria pathogens, the Plasmodium falciparum and Plasmodium vivax attack human red blood cells (RBCs) causing high fever, cough, chills, headache, nausea, and vomiting, muscle pain and fatigue, transpiration, abdominal and chest pain, and the death [2], [3]. Untreated malaria patients may develop long-term pneumonia, anemia, yellow fever, respiratory or brain disorders (Cerebral Malaria) [4].

The World Health Organization (WHO) is deploying strategies for the prevention, treatment, elimination, and surveillance of Malaria to face this global pandemic [1], starting with the diagnosis of the disease. Basically, microscopic diagnosis method detecting Malaria consist in a microbiological analysis using peripheral blood slides [5], [6]. Collection of blood smears for manual microscopic analysis remains an effective method in the diagnosis of malaria, compared to other methods such as polymerase chain reaction (PCR) and in-house tests [7]. Nevertheless, analyses by standard

The associate editor coordinating the review of this manuscript and approving it for publication was Fahmi Khalifa.

microscopes typically require a long time to obtain a compelling diagnosis and related results are dependent on the equipment quality, the luminous and practical conditions of the laboratory, and finally, on the technician's analysis knowledge [8]. Standardly, the Malaria diagnosis evaluation is based on the cost per test, the sensitivity and specificity of the method, the time per test, and the qualified user skill [9]. A poorly elaborated diagnosis exposes patients to erroneous treatment, with all the possible consequences of inappropriate medication. To deal with these limitations, computer-aided diagnostic (CAD) systems for malaria analysis are developed.

Recently, numerous Deep learning approaches based on Artificial Neural Networks (ANN) have been used to make fast and accurate malaria's diagnoses employing public datasets [10], [11]. These approaches for malaria blood smear detection, segmentation, and classification use or modify the conventional CNN to classify images of infected red blood cells.

Lately, Akgül *et al.* [11] used ResNet architectures to classify the malaria parasites achieving an accuracy of 94.09% with the ResNet-50 v2 model while, Loddo *et al.* [10] performed the four-class classification on the Plasmodium falciparum stages of life using the DenseNet-201 neural network (99.40 %).

In [12], an automated Convolutional Neural Network (CNN) was implemented for the diagnosis of malaria using microscopic blood smear images. The developed approach merges various techniques as distillation, data augmentation, Autoencoder, CNN model, Support Vector Machine (SVM), and K-Nearest Neighbors (KNN) to improve the model accuracy and inference performance, obtaining an accuracy of 99.23% with 4600 floating point operations in the detection of malarial parasites. Web-based and mobile-based applications were implemented performing the model inference under 1 s per sample. Arunagiri *et al.* proposed a transfer learning approach based on Visual Geometry Group (VGG) network and Support Vector Machine (SVM) to identify infected falciparum malaria parasite [13], achieving an accuracy of 93.1 %.

Other deep learning approaches are based on feature extraction before processing images. The case of contrast enhancement methods proposed by Talha *et al.* where the DarkNet-53 and DenseNet-201 pretrained deep convolutional neural networks performed the feature extraction of the CIELAB color space images in the malaria blood smear classification [14]. For their part, Salamah *et al.* [15] proposed a segmentation technique for malaria parasite detection of microscopic images, obtaining an average precision, recall, and F-measure of 86% .

Overall, numerous techniques based on morphological operations [16], Local Composite Pattern (LBP) [17], Moving K-Means Clustering [18], deep learning [12] and transfer learning models [19] are developed to improve malaria's infected cell detection. Therefore, the challenges of diagnosing malaria red blood cells using deep and transfer learning techniques remain the detection accuracy, the diagnostic time frame and the process computational cost. In this sense, various recent works are focused on the data preprocessing step to perform the malaria detection [20], [21].

To deal with the cell detection accuracy and the diagnostic time frame challenges, this work aims to classify malaria red blood cells without a preprocessing step, using two convolutional-recurrent neural networks and the public malaria dataset from the National Library of Medicine [22].

The main contributions of this article are summarized as follows,

1) Because of the implemented learning acceleration, the developed models all converge with only 100 training iterations.
2) Approaches based on CNN-LSTM and CNN-BiLSTM network cascade to improve cells classification.
3) Comparative results of malaria red blood cells classification achieved using convolutional-recurrent neural networks.

As novelty, this work investigates on recurrent neural networks contribution detecting malaria red blood cells, emphasizing on data splitting, and images sizing to evaluate the processing time, since various related works are based on the basic CNN or its variants.

The paper is organized as follows. Section II presents the referred malaria dataset, artificial neural architectures and methods developed in this framework while the results encountered are reported and discussed in Section III. Finally, Section IV gives the paper conclusion and projections for the future works.

II. METHODS

In making the binary-class classification of the parasitized and uninfected cells in this paper, two neural network architectures are implemented: A cascade of Deep CNN-LSTM and CNN-BiLSTM networks.

This work considers different input images size, that is, image sizes of 96×96, 64×64, 32×32, and 28×28 pixels to evaluate the processing time of each category. For its part, the CNN component of the deep network block is configured with the same parameters merging with the LSTM or BiLSTM parts for each considered case. Dense layers are composed of two neurons each one, activated by the Softmax function evaluating the class probability for an output image.

A. OVERALL FLOWCHART

Figure 1 illustrates the step-by-step diagram of methods implemented with the same public dataset in this paper. The procedure's main purpose is to evaluate results obtained with different network architectures, classifying malaria uninfected and parasitized cells.

B. NLM–MALARIA DATASET

The malaria dataset is provided by National Library of Medicine, Lister Hill National Center for Biomedical Communications. The study was approved by the Institutional Review Board of Office of Human Subjects Research

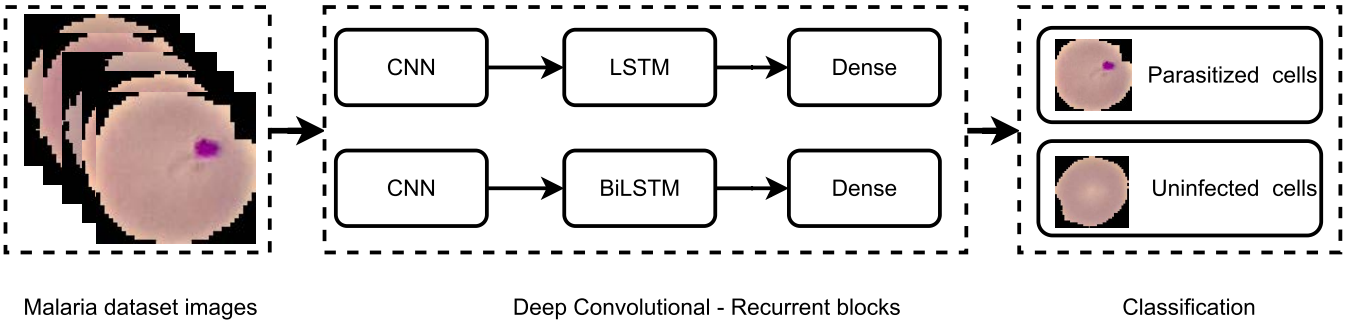


FIGURE 1. The general high level block diagram of the method. Two approaches are developed in this work to detect parasitized red blood cells from those uninfected: The first is based on deep CNN-LSTM cascade and the second uses a Deep CNN-BiLSTM combination, all followed by dense layers.

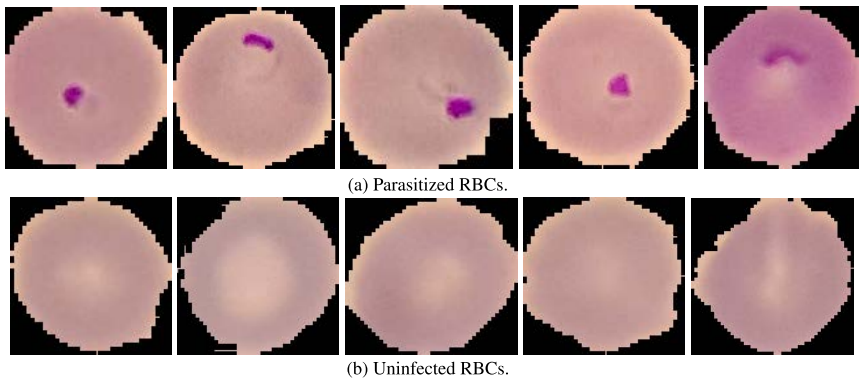


FIGURE 2. Example of cell images contained in the dataset: (a): parasitized cells and (b): uninfected cells.

(OHSR) (protocol code 12972 and date of approval 25 June 2015). This dataset contains 27,560 color images of 96×96 pixels, Giemsa-stained blood samples obtained from 193 patients and distributed evenly between parasitized and uninfected red blood cell images.

Figure 2 shows parasitized and uninfected RBCs as contained in the referred dataset. Artificial neural networks implemented classify images of infected cells and those of uninfected cells. Therefore, the malaria dataset was split into training and test subsets, as illustrated in Table 1.

C. THE CNN-LSTM ARCHITECTURE

The first block is based on a Convolutional Neural Network, recognized for its best performances processing images [23], [24]. CNNs use convolution mathematical operations at the hidden layers level to learn local data features. Therefore, RGB color images are processed directly by the CNN without any preprocessing step, performing the diagnostic delay as much as possible. The second block is constituted by an LSTM model. Initially, LSTM network was introduced to deal with vanishing or exploding gradient problem processing data long sequences with backpropagation algorithms [25]. An LSTM Network is constituted by memory blocks, composed of memory units storing the network's temporary states. The flow of sequences is controlled by recurrent connections into the network, depending essentially on the

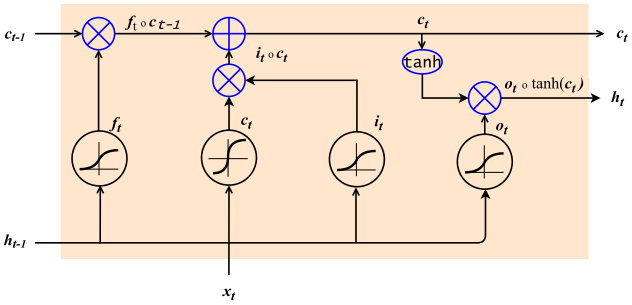


FIGURE 3. An LSTM memory unit structure. Memory units, also called *Constant Error Carousels (CECs)* store the network's temporary states. The flow of sequences is controlled by recurrent connections into the network.

number of memory unit blocks, the number of memory units per block, and how the weights are initialized.

Figure 3 illustrates the LSTM memory unit structure, centered around the input, forget, and output gates.

At time t, the input gate upshot is

$$i_t = \sigma(W_i^x x_t + W_i^h h_{t-1} + b_i), \tag{1}$$

being W_i^x and W_i^h the weight matrices, h_{t-1} the previous hidden state unit, b_i the bias vector, and $\sigma(x)$ the sigmoid activation function.

TABLE 1. Dataset partition approaches for training and testing data.

Dataset	Training data		Testing data	
split	Percentage (%)	Cell images	Percentage (%)	Cell images
Split-1	90	24,804	10	2,756
Split-2	80	22,048	20	5,512
Split-3	70	19,292	30	8,268
Split-4	60	16,536	40	11,024
Split-5	50	13,780	50	13,780

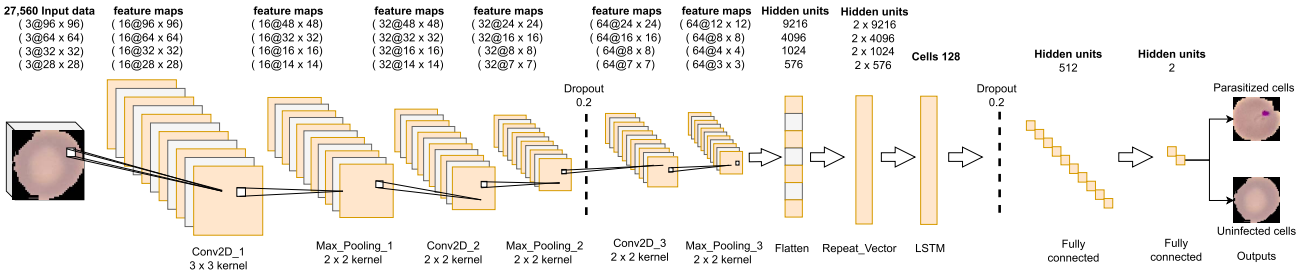


FIGURE 4. The implemented CNN-LSTM network. The model is based on three 2D-convolutional layers separated by max pooling of 2 × 2 kernel, two dropouts, one flatten, an LSTM layer of 128 cells, and two fully connected layers.

Likewise, the forget gate produces the output f_t using the $\sigma(x)$ activation function as,

$$f_t = \sigma(W_f^x x_t + W_f^h h_{t-1} + b_f). \tag{2}$$

According to the input x_t, the precedent state h_{t-1}, and the cell state c_{t-1}, the output of the output gate is,

$$\tilde{c}_t = \tanh(W_{\tilde{c}}^x x_t + W_{\tilde{c}}^h h_{t-1} + b_{\tilde{c}}), \tag{3}$$
$$c_t = i_t \odot \tilde{c}_t + f_t \odot c_{t-1}, \tag{4}$$
$$o_t = \sigma(W_o^x x_t + W_o^h h_{t-1} + b_o), \tag{5}$$
$$h_t = o_t \odot \tanh(c_t), \tag{6}$$

where $\odot$ is the Hadamard product and W_o^x, W_o^h are the output weight matrices.

Implementing a cascade of CNN-LSTM expects to perform the data classification, taking advantage of features extraction by the CNN and their classification by the LSTM. The CNN-LSTM network in Figure 4 alternately receives 96×96, 64×64, 32×32, and 28×28 RGB color images as input to make the output prediction between parasitized and uninfected RBCs. The last dense layer constituted by 2 hidden units makes the class prediction on the input data using the *SoftMax* activation function given by

$$\sigma(x)_i = \frac{e^{x_i}}{\sum_{j=1}^{2} e^{x_j}}, for\ i = 1, 2, \ldots, n \tag{7}$$
$$and\ x = (x_1, x_2, \ldots, n) \epsilon\ \mathbb{R}^2, \tag{8}$$

where the denominator represents the normalization expression for the probability distribution and x_i the feature map to be classified in its respective class.

All the convolutional layers are activated by the Rectifed Linear Activation Function (ReLU), offering gradients between 0 and 1 and defined as,

$$f(x) = max(0, x) \tag{9}$$

Table 2 outlines main parameters of the configured layers. While the other layers operate with a similar number of parameters, only the LSTM layer varies its numeral of model parameters according to the input data size.

D. THE CNN-BiLSTM ARCHITECTURE

The second approach developed in this work contemplates processing data with a CNN-BiLSTM network, as illustrated in Figure 5. A BiLSTM network is constituted by two LSTM blocks working each one in the forward and backward direction [26]. Thus, the BiLSTM architecture results in faster convergence and better classification accuracy than the basic LSTM [27]. For a given data processing t^{th} time, the BiLSTM unit considers the past and the future unit state, as shown in Figure 6.

Therefore, the t^{th} BiLSTM output merges forward and backward outputs as,

$$h_t = [\overrightarrow{For_t} \oplus \overleftarrow{Back_t}], \tag{10}$$

where $\overrightarrow{For_t}$ and $\overleftarrow{Back_t}$ are the t^{th} LSTM memory block outputs, and $\oplus$ is the element-wise sum.

The implemented BiLSTM network uses two LSTM blocks each one configured with 64 cell units, followed by two fully-connected layers of 512 and 2 neurons, respectively (see Figure 5). 4751872 and 98816 parameters are used

TABLE 2. Configuration of the CNN-LSTM network parameters.

Type of Layer	Number of parameters			
	(28 × 28) data	(32 × 32) data	(64 × 64) data	(96 × 96) data
CNN1 (Conv2D)	448	448	448	448
Max_pooling2D_1	0	0	0	0
CNN2 (Conv2D)	4640	4640	4640	4640
Max_pooling2D_2	0	0	0	0
Dropout_1	0	0	0	0
CNN3 (Conv2D).	18496	18496	18496	18496
Max_pooling2D_3	0	0	0	0
Flatten_1	0	0	0	0
Repeat_vector	0	0	0	0
LSTM	360960	590336	2163200	4784640
Dropout_2	0	0	0	0
Dense_1	66048	66048	66048	66048
Dense_2	1026	1026	1026	1026

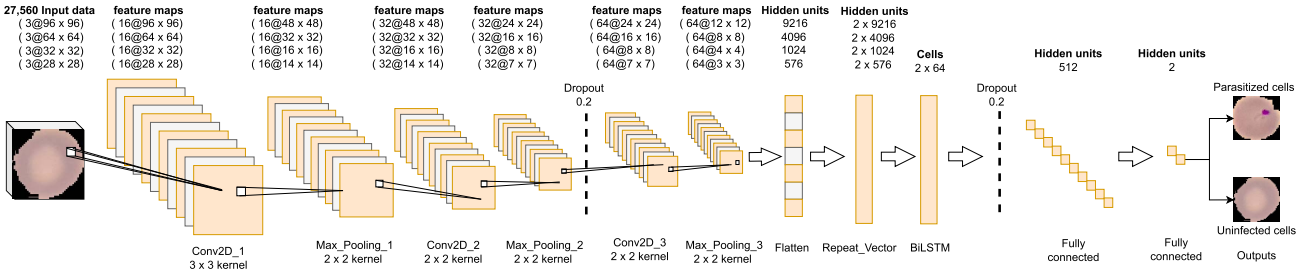


FIGURE 5. Overview of the implemented CNN-BiLSTM architecture. The BiLSTM block performs data classification taking advantage of the feature extraction fulfilled by the CNN block.

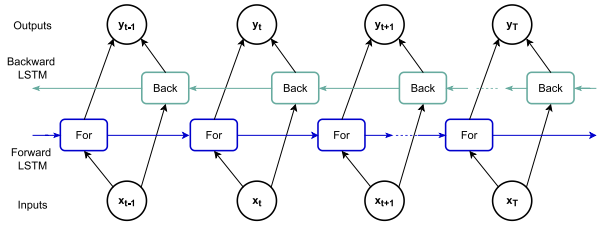


FIGURE 6. Unfolded structure of a BiLSTM network. Data are processing from the past into the future by the forward LSTM, while the backward LSTM processes data from the future into the past.

to configure the two BiLSTM cells while the remain layers are implemented with the same parameters that the CNN-LSTM network.

All the two architectures were trained on a Windows 10 desktop equipped with an NVIDIA GTX 1080 Ti GPU for 100 epochs using a batch size of 32, the Nesterov-accelerated Adaptive Moment Estimation (Nadam) optimizer, and the sparse categorical cross-entropy loss cost to estimate cell classification accuracy. In addition, the Cyclical Learning Rate (CLR) algorithm [28] was implemented to break out from local minima, accelerating the convergence of the training process. Thereby, setting the step size to 8 times the number of iterations per epoch, the minimum and maximum CLR were fixed to 7×10^{-5} and 9×10^{-3}, respectively.

III. RESULTS AND DISCUSSION

Results obtained with the two approaches are evaluated using the Precision, F1-score, Recall, and accuracy metrics. Assigning class M features to another class is called False Negative (FN) while True negatives (TN) are all features of other classes than P not assigned to class P, False Positives (FP) being all features erroneously assigned to class P. Therefore, the following statistical metrics are defined for the classification evaluation,

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \tag{11}$$

$$Recall = \frac{TP}{TP + FN}, \tag{12}$$

$$Precision = \frac{TP}{TP + FP}, \tag{13}$$

$$F1 - score = \frac{2 \times Precision \times Recall}{Precision + Recall}. \tag{14}$$

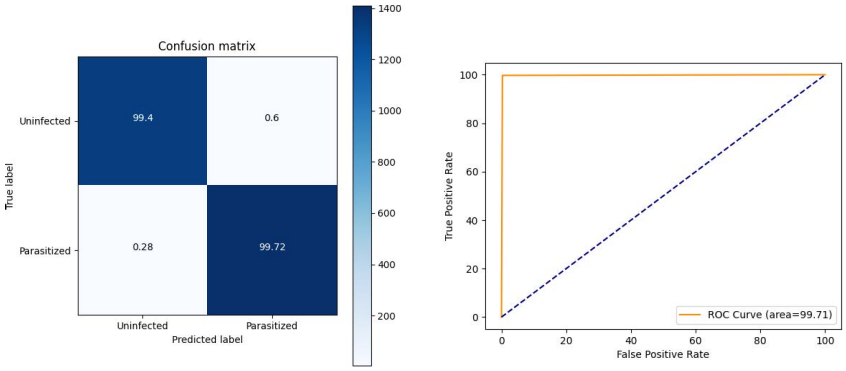
The following tables contain average results on five tests made with each architecture model, resetting the weights randomly.

TABLE 3. CNN-LSTM architecture results splitting the dataset into training and testing data.

Dataset split	Image size	Training vs. Testing (%)	Accuracy	Precision	Recall	F1-Score
split-1	28 × 28	90 vs. 10	99.63	99.43	99.85	99.64
	32 × 32	90 vs. 10	99.67	99.64	99.57	99.61
	64 × 64	90 vs. 10	99.78	99.57	99.85	99.71
	96 × 96	90 vs. 10	**99.86**	99.71	99.85	99.78
Split-2	28 × 28	80 vs. 20	99.27	98.96	99.31	99.14
	32 × 32	80 vs. 20	99.31	98.96	99.67	99.32
	64 × 64	80 vs. 20	99.29	98.89	99.42	99.15
	96 × 96	80 vs. 20	**99.36**	98.92	99.24	99.08
Split-3	28 × 28	70 vs. 30	98.87	98.5	99.14	98.82
	32 × 32	70 vs. 30	98.97	98.55	99.07	98.81
	64 × 64	70 vs. 30	98.89	98.62	98.88	98.75
	96 × 96	70 vs. 30	**98.98**	98.67	99.16	98.91
Split-4	28 × 28	60 vs. 40	98.34	97.67	98.70	98.18
	32 × 32	60 vs. 40	98.30	97.58	98.63	98.10
	64 × 64	60 vs. 40	98.37	97.29	98.63	97.95
	96 × 96	60 vs. 40	**98.42**	97.87	97.62	97.75
Split-5	28 × 28	50 vs. 50	97.74	96.61	96.91	96.76
	32 × 32	50 vs. 50	97.79	98.15	97.5	96.82
	64 × 64	50 vs. 50	97.93	96.16	97.92	96.87
	96 × 96	50 vs. 50	**98.00**	97.32	97.24	97.28

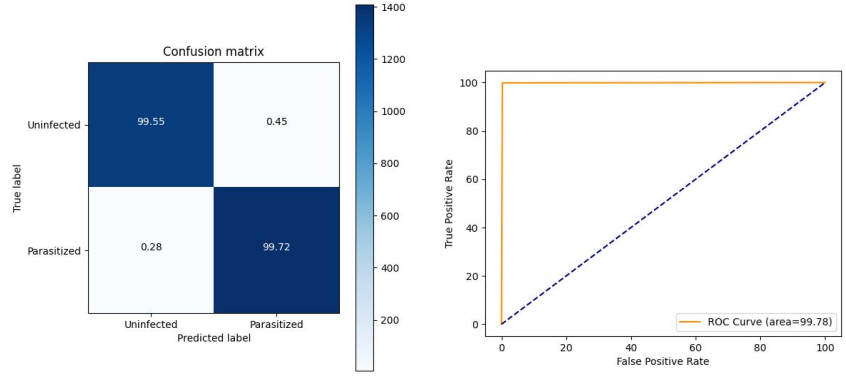
TABLE 4. Results achieved with the proposed CNN-BiLSTM for all dataset splits.

Dataset Split	Image size	Training vs. Testing (%)	Accuracy	Precision	Recall	F1-Score
split-1	28 × 28	90 vs. 10	99.74	99.71	99.78	99.75
	32 × 32	90 vs. 10	99.81	99.78	99.85	99.82
	64 × 64	90 vs. 10	99.85	99.71	99.71	99.71
	96 × 96	90 vs. 10	**99.89**	99.73	99.89	99.76
Split-2	28 × 28	80 vs. 20	99.29	99.06	99.24	99.15
	32 × 32	80 vs. 20	99.25	98.86	99.60	99.23
	64 × 64	80 vs. 20	99.31	98.82	99.35	99.08
	96 × 96	80 vs. 20	**99.34**	98.89	99.24	99.06
Split-3	28 × 28	70 vs. 30	98.85	97.90	98.95	98.42
	32 × 32	70 vs. 30	98.80	98.40	98.57	98.48
	64 × 64	70 vs. 30	98.85	98.51	99.21	98.86
	96 × 96	70 vs. 30	**98.89**	98.64	98.61	98.63
Split-4	28 × 28	60 vs. 40	98.30	96.52	97.81	97.65
	32 × 32	60 vs. 40	98.33	96.53	99.15	97.82
	64 × 64	60 vs. 40	98.39	96.05	97.39	97.31
	96 × 96	60 vs. 40	**98.47**	96.29	98.00	97.14
Split-5	28 × 28	50 vs. 50	97.75	96.42	97.57	96.99
	32 × 32	50 vs. 50	97.66	96.40	97.89	97.14
	64 × 64	50 vs. 50	97.86	96.86	98.42	97.63
	96 × 96	50 vs. 50	**97.93**	96.17	96.42	96.29



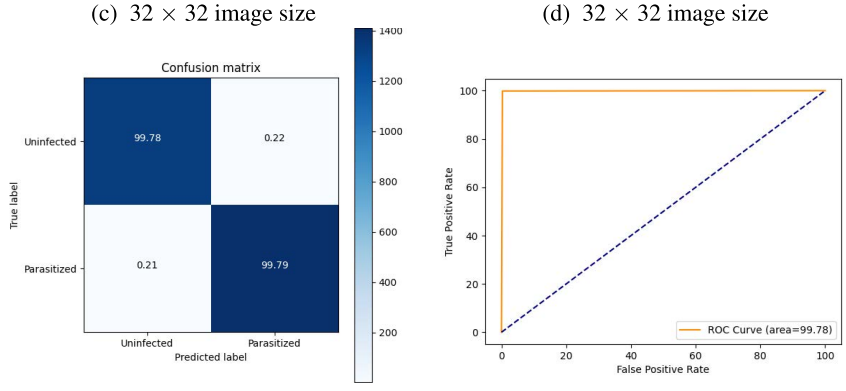
(a) 28 × 28 image size

(b) 28 × 28 image size



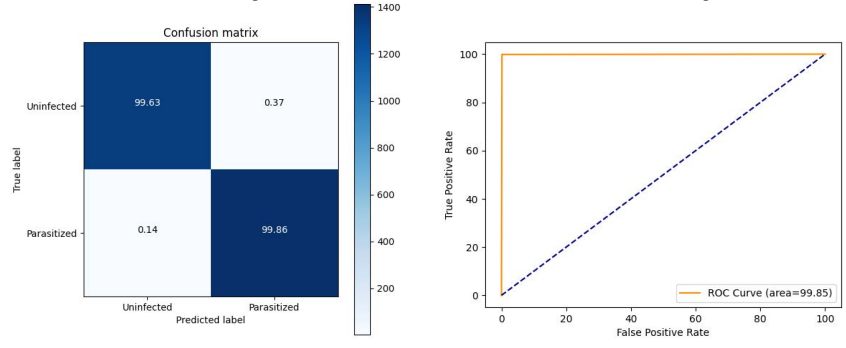
(c) 32 × 32 image size

(d) 32 × 32 image size



(e) 64 × 64 image size

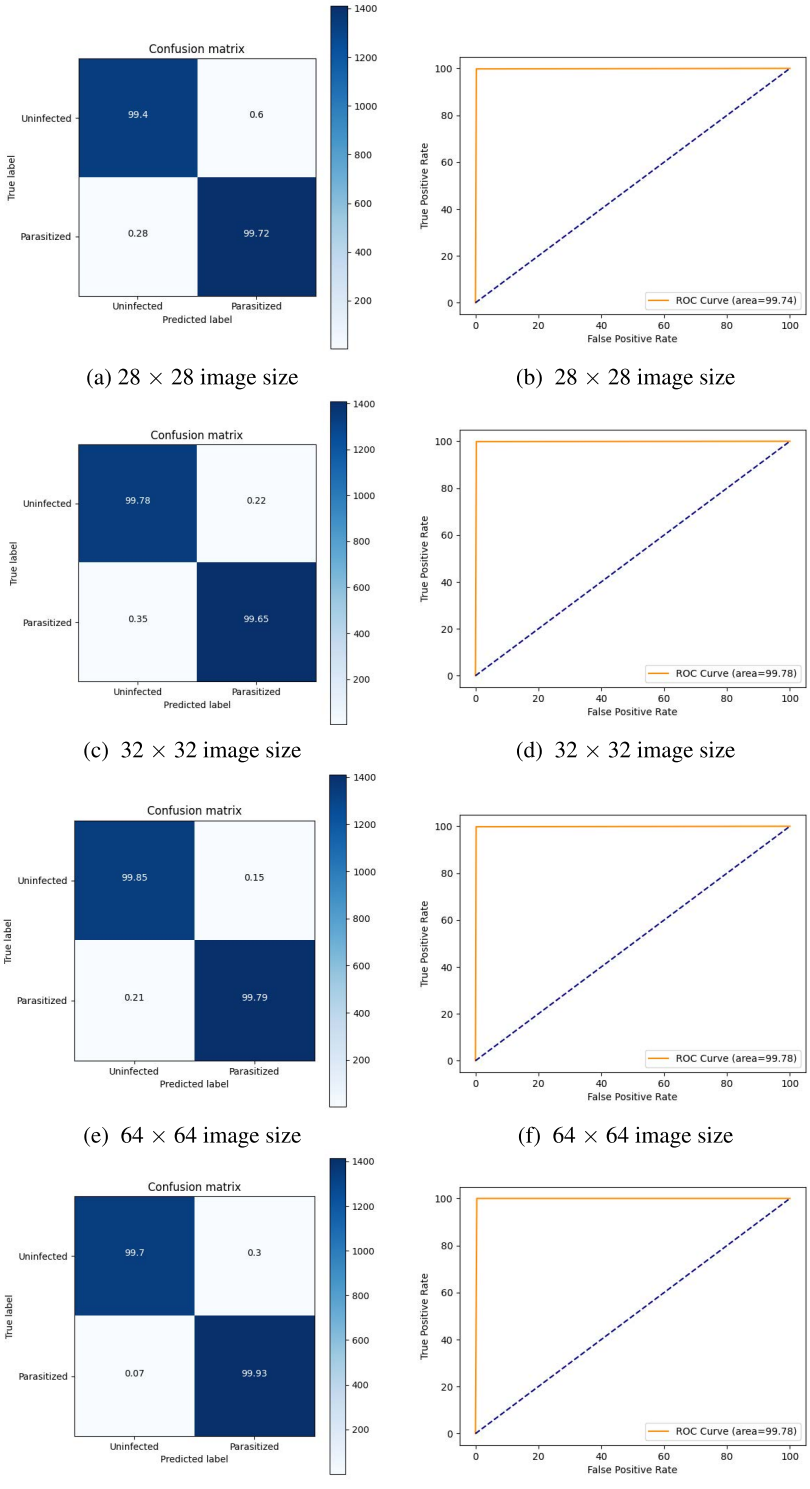
(f) 64 × 64 image size



(g) 96 × 96 image size

(h) 96 × 96 image size

FIGURE 7. Confusion matrices (on the left column) and ROC curves (on the right column) evaluating the classifier performances for split-1 data partition (90 vs. 10) using the CNN-LSTM network.



(a) 28 × 28 image size (b) 28 × 28 image size

(c) 32 × 32 image size (d) 32 × 32 image size

(e) 64 × 64 image size (f) 64 × 64 image size

(g) 96 × 96 image size (h) 96 × 96 image size

FIGURE 8. Confusion matrices (on the left column) and ROC curves (on the right column) for the split-1 data partition (90 vs. 10) using the proposed CNN-BiLSTM network.

A. RESULTS ACHIEVED WITH THE CNN-LSTM ARCHITECTURE

Table 3 presents the results achieved with the first approach using the CNN-LSTM network. With the first partition data, considering 90 % for the model training and 10 % for the model testing, accuracies of 99.63 %, 99.67 %, 99.78 %, and 99.86 % were achieved classifying malaria parasitized and uninfected RBCs. Precision, Recall, and F1-Score metrics are

reported next to each data size for this partition, respectively. For the split-2, split-3, split-4 and split-5 data partition, the best accuracies of 99.36 %, 98.98 %, 98.42 %, and 98.00 % were obtained, all with 96 × 96 size images.

Figure 7 presents the confusion matrix and receiver operating characteristic (ROC) curves evaluating the quality of the classifier output using only the split-1 data partition that gave best performances. On the confusion matrix, the predicted labels competing to equalize to the true labels are represented by the diagonal elements, while the mislabeled elements are located on the matrix off-diagonal. For its part, the ROC curve built from the true positive rate (TPR) against the false positive rate (FPR) typically illustrates the trade-off between the sensitivity and specificity metrics. Classifiers with better performance present ROC curves closer to the graph's top-left corner. On the diagonal, the TPR is equal to the FPR, given a random classifier. The area under the ROC curve (AUC), directly linked to the accuracy metric, calculates the classifier's performance in discriminating between parasitized and uninfected cells.

In the case of 28 × 28 image size, 99.72 % of cell images were correctly detected as parasitized against 99.40 % for uninfected cell images. 99.65 % of samples were accurately diagnosed as parasitized in contrast to 99.78 % for uninfected cells considering the 32 × 32 image size. According to the Figure 7(e) where 64 × 64 samples were used, parasitized and uninfected cells got close predictions of 99.79 % and 99.78 % comparing with their true labels. Finally, the CNN-LSTM classifier correctly predicted 99.86 % as parasitized and 99.63 % of 96 × 96 samples as uninfected.

B. RESULTS OBTAINED WITH THE CNN-BiLSTM ARCHITECTURE

By replacing the LSTM block of the previous architecture with two BiLSTM layers, results achieved are shown on Table 4, where the best accuracy of 99.89 % was obtained with the split-1 data partition, followed by accuracies of 99.34 %, 98,89 %, 98.47 % and 97.93 % for the split-2, split-3, split-4, and split-5, in this order. These performances were reached using 96 × 96 image size. The precision, Recall and F1-score metrics follow the same decreasing trend while the data partitions pass from split-1 to split-5.

Figure 8 shows the confusion matrices and ROC curves for the CNN-BiLSTM model using the split-1 data partition.

Parasitized and uninfected RBC predictions varied depending on image pixel size and the database splitting. 99.99 % of 96 × 96 sized-images were correctly labeled as parasitized against 99.7 % for uninfected cells. 0.3 % of uninfected cells were mislabeled against 0.07 of parasitized.

C. DISCUSSION

By evaluating the architectural models proposed in this work, three important parameters were observed, namely, the size of images to be processed, the adjusted rate of the training against the testing data and the processing time required by the classifier. Talking about the size of samples being processed, the results in Tables 3 and 4 support the observation that the large-sized images produced better performance in their processing than the small-sized. This is understood by the large quantity of pixels necessary for the features learning by the CNN block. Splitting the database into training and testing data on top carries out a significant role. The same tables report performances for the split-1 which considers 90 % images for training and 10 % for testing than for the other partitions. The split-5 partition produced the lowest results of all splits (98 % and 97.93 %), despite the considered image sizes.

Finally, the processing time that the classifier considers to decide whether a selected sample is infected or not, should be added to these observations. Large-sized images take longer (despite < 1 s) than small-sized, as revealed in Table 5. Therefore, comparing the two proposed architecture models for the 96 × 96 image size, The CNN-BiLSTM model took longer than the CNN-LSTM classifier to detect a data sample, justifiable by the data twice-processing operated by the BiLSTM. Obviously, the results presented in Table 5 correspond to the used computer resources and can be improved using a more powerful computer. Therefore, the reasonable compromise should be negotiated amidst these three observations for a usage application, give-and-take between the size of images — data partition — processing time.

The literature is flourishing with scientific research relating to the detection of cells infected and uninfected by malaria employing the NLM database and various deep learning approaches. Among all these specific techniques efficiently deployed, the present work contributes in meaningfully comparing results of two combined architectures of convolutional and recurrent neural networks, without preprocessing data. Table 6 presents the comparison between the results of recent state-of-the-art works and those achieved in this framework.

Recently, [29] implemented a customized CNN to detect peripheral malarial parasites in blood Smears, achieving an accuracy of 98 % with 27500 images resized at 128 × 128 pixel size. Next, their used morphological filters and a a fine-tuned pretrained CNN model to classify parasitized and uninfected malaria cells, obtaining an average accuracy of 98.34 ± 0.51%.

Likewise, [20] proposed a model based on Barnacles Mating Optimizer with Deep Transfer Learning (BMODTL) to detect and classify malaria parasite. Concretely, they preprocessed data using the Gaussian filter (GF) and Graph cuts (GC) segmentation technique to locate the contaminated areas in the blood smear images, before classifying malaria parasites. An accuracy of 99.04 % was achieved with their model. In another work, the malaria detection based on offline and Web application was implemented using various approaches contemplating distillation, morphology, Autoencoder, CNN - Support Vector Machine (SVM) or K-Nearest Neighbors (KNN) [12]. They reported an accuracy of 99.23 processing malaria RBC images of 28 × 28, 32 × 32, and 64 × 64 pixel size. Splitting data in 80 %

TABLE 5. Classifier running time to detect a sample of malaria cell.

Dataset split	Model	Running time for sample (second)			
		28×28	32×32	64×64	96×96
Split-1	CNN-LSTM	0.111	0.133	0.261	0.446
	CNN-BiLSTM	0.142	0.173	0.346	0.581
Split-2	CNN-LSTM	0.082	0.103	0.221	0.439
	CNN-BiLSTM	0.098	0.113	0.269	0.462
Split-3	CNN-LSTM	0.042	0.052	0.228	0.389
	CNN-BiLSTM	0.054	0.055	0.184	0.401
Split-4	CNN-LSTM	0.028	0.040	0.147	0.311
	CNN-BiLSTM	0.074	0.093	0.190	0.317
Split-5	CNN-LSTM	0.025	0.031	0.125	0.252
	CNN-BiLSTM	0.027	0.039	0.130	0.276

TABLE 6. Comparison of the results with the state of the art.

Author	Method	Accuracy	Precision	Recall	F1-score
Alharbi [29]	Customized CNN	98.0	–	98.5	98.7
Alharbi [30]	CNN/VGG-19	97.04	97.04	97.00	97.00
Dutta [20]	BMODTL-BMPC	99.04	99.05	99.05	99.04
Fuhad [12]	Autoencoder	99.23	98.92	99.52	99.22
Madhu [31]	CapsNet	98.82	98.82	98.36	99.12
Proposed model 1	CNN-LSTM	99.86	99.71	99.85	99.78
Proposed model 2	CNN-BiLSTM	99.89	99.73	99.89	99.76

for training and 20 % for testing, [31] obtained an accuracy of 98.82 % with their model based on an imperative dynamic routing mechanism with fully trained capsule networks, classifying malaria RBCs.

This work, in addition to exploring various database splittings for the model training and testing, offers competitive results comparing with those of the state-of-the-art. Approaches developed combining convolutional and recurrent neural networks to accurately detect parasitized and uninfected malaria cells, manfully helped to achieve this goal.

IV. CONCLUSION

This work aimed to classify parasitized and uninfected malaria cells using two deep learning architectures without any preprocessing stage. The proposed networks combined the basic convolutional neural network and two specific recurrent neural network, the Long Term - Short Memory (LSTM) to solve the vanishing gradient problem and the Bidirectional LSTM to perform the classification, processing data twice. The accuracy, precision, recall, F1-score and confusion matrix metrics were used to evaluate the model performances, considering the images size, data splitting for model training and testing, and the processing time per data sample. The proposed models achieved competitive results in comparison with those of the state-of-the-art, offering new alternatives for further works. Samples of 96×96 pixel size gave the best results splitting 90 % of data for the training and 10 % for the testing. Accuracies of 99.86 and 99.89 % were achieved with the CNN-LSTM and CNN-BiLSTM, training these models with 100 epochs at least. Implementations of the developed architectures on FPGAs are projected as future works to accelerate image processing, with the real-time detection purpose. To anticipate this specific projection, this work considered image processing of various sizes to analyze a practical trade-off between processing delay and cell detection accuracy. The code sources for theses experiments are released on the github `https://github.com/adanantonio07A/malaria_classification`.

REFERENCES

[1] O. M. de la Salud. (2021). *Informe Mundial de Malaria 2021*. Accessed: Apr. 26, 2022. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/malaria

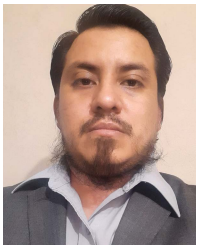
[2] V. Thathy and D. Fidock, "Malaria parasite beats the heat," *Nature Microbiol.*, vol. 6, pp. 1–3, Aug. 2021.

[3] N. White, "Malaria parasite clearance," *Malaria J.*, vol. 16, pp. 1–14, Feb. 2017.

[4] C. Moxon, G. Grau, and A. Craig, "Malaria: Modification of the red blood cell and consequences in the human host," *Brit. J. Haematol.*, vol. 154, pp. 670–679, May 2011.

[5] A. Loddo, C. Di Ruberto, and M. Kocher, "Recent advances of malaria parasites detection systems based on mathematical morphology," *Sensors*, vol. 18, pp. 1–21, Feb. 2018.

[6] B. Gitta and N. Kilian, "Diagnosis of malaria parasites plasmodium SPP. In endemic areas: Current strategies for an ancient disease," *BioEssays*, vol. 42, pp. 1–12, Dec. 2019.
[7] T. Hanscheid, "Diagnosis of malaria: A review of alternatives to conventional microscopy," *Clin. Lab. Haematol.*, vol. 21, pp. 45–235, Sep. 1999.
[8] T. Hanscheid and E. Valadas, "Malaria diagnosis," *Amer. J. Tropical Med. Hygiene*, vol. 61, p. 179, Sep. 1999.
[9] M. Poostchi, K. Silamut, R. J. Maude, S. Jaeger, and G. Thoma, "Image analysis and machine learning for detecting malaria," *Transl. Res.*, vol. 194, pp. 36–55, Apr. 2018.
[10] A. Loddo, C. Fadda, and C. Di Ruberto, "An empirical evaluation of convolutional networks for malaria diagnosis," *J. Imag.*, vol. 8, no. 3, p. 66, Mar. 2022.
[11] I. Akgül and V. Kaya, "Classification of cells infected with the malaria parasite with ResNet architectures," *J. Sci. Rep. A*, no. 48, pp. 52–54, Apr. 2022.
[12] K. M. Fuhad, J. Tuba, M. Sarker, S. Momen, N. Mohammed, and T. Rahman, "Deep learning based automatic malaria parasite detection from blood smear and its smartphone based application," *Diagnostics*, vol. 10, p. 329, May 2020.
[13] V. Arunagiri and B. R. Kanna, "Deep learning approach to detect malaria from microscopic images," *Multimedia Tools Appl.*, vol. 79, pp. 15297–15317, Jun. 2020.
[14] T. Imran, M. A. Khan, M. Sharif, U. Tariq, Y.-D. Zhang, Y. Nam, Y. Nam, and B.-G. Kang, "Malaria blood smear classification using deep learning and best features selection," *Comput., Mater. Continua*, vol. 70, no. 1, pp. 1875–1891, 2022.
[15] U. Salamah, R. Sarno, A. Z. Arifin, A. S. Nugroho, I. E. Rozi, and P. B. S. Asih, "A robust segmentation for malaria parasite detection of thick blood smear microscopic images," *Int. J. Adv. Sci., Eng. Inf. Technol.*, vol. 9, no. 4, p. 1450, Aug. 2019.
[16] J. Arco, J. Gorriz, J. Ramírez, I. Illan, and C. Puntonet, "Digital image analysis for automatic enumeration of malaria parasites using morphological operations," *Exp. Syst. Appl.*, vol. 42, pp. 3041–3047, Apr. 015.
[17] J. Ovi, M. E. Haque, A. Kalam, S. Jarin, M. Ali, and M. Hasan, "Malaria detection using local composition pattern," *J. Phys., Conf.*, vol. 1803, Feb. 2021, Art. no. 012014.
[18] A. S. A. Nasir, M. Mashor, and Z. Mohamed, "Segmentation based approach for detection of malaria parasites using moving K-means clustering," in *Proc. IEEE-EMBS Conf. Biomed. Eng. Sci. (IECBES)*, Dec. 2012, pp. 652–658.
[19] A. Alassaf and M. Y. Sikkandar, "Intelligent deep transfer learning based malaria parasite detection and classification model using biomedical image," *Comput., Mater. Continua*, vol. 72, no. 3, pp. 5273–5285, 2022.
[20] A. Dutta, R. Mageswari, A. Gayathri, J. Bruxella, M. Ishak, S. Mostafa, and H. Hamam, "Barnacles mating optimizer with deep transfer learning enabled biomedical malaria parasite detection and classification," *Comput. Intell. Neurosci.*, vol. 2022, pp. 1–12, Jun. 2022.
[21] N. Mustare, "Modified CNN model for malaria diagnosis," *J. Positive School Psychol.*, vol. 6, pp. 6483–6492, Jul. 2022.
[22] N. Library of Medecine. *Malaria Cell Images Dataset*. Accessed: May 1, 2022. [Online]. Available: https://lhncbc.nlm.nih.gov/LHC-research/LHC-projects/image-processing/malaria-datasheet.html
[23] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, "Convolutional neural networks: An overview and application in radiology," *Insights Imag.*, vol. 9, pp. 611–629, Aug. 2018.
[24] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidie, and L. Farhan, "Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions," *J. Big Data*, vol. 8, p. 53, Mar. 2021.
[25] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Netw.*, vol. 61, pp. 85–117, Jan. 2014.
[26] Y. Yu, X. Si, C. Hu, and Z. Jianxun, "A review of recurrent neural networks: LSTM cells and network architectures," *Neural Comput.*, vol. 31, no. 7, pp. 1235–1270, 2019.
[27] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures," *Neural Netw.*, vol. 18, no. 5, pp. 602–610, 2005.
[28] L. N. Smith, "Cyclical learning rates for training neural networks," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2017, pp. 464–472.
[29] A. Alharbi, C. V. Aravinda, L. Meng, B. Ashwini, M. Y. Jabarulla, and M. A. Shah, "Detection of peripheral malarial parasites in blood smears using deep learning models," *Comput. Intell. Neurosci.*, vol. 2022, pp. 1–11, May 2022.
[30] A. Alharbi, C. V. Aravinda, J. Shetty, M. Y. Jabarulla, K. B. Sudeepa, and S. Singh, "Computational models-based detection of peripheral malarial parasites in blood smears," *Contrast Media Mol. Imag.*, vol. 2022, pp. 1–9, Jun. 2022.
[31] G. Madhu, A. Govardhan, B. S. Srinivas, K. S. Sahoo, N. Zaman, S. Vardhan, and R. Boddeda, "Imperative dynamic routing between capsules network for malaria classification," *Comput., Mater. Continua*, vol. 680, pp. 903–919, Mar. 2021.



ADÁN ANTONIO ALONSO-RAMÍREZ was born in Huixtla Chiapas, Mexico, in July 1988. He received the bachelor's degree in computer science in the autonomous "Benito Juárez" University of Oaxaca, in 2014, and the master's degree in electrical engineering (instrumentation and digital systems) from Guanajuato University, in 2018. He is currently pursuing the Ph.D. degree with the National Tecnnological of Mexico, Celaya, Guanajuato, Mexico. His current investigations are related to image processing and deep learning, focus on embedded systems with the Autotronic Department.

TAT'Y MWATA-VELU was born in Kikwit, Democratic Republic of Congo. He received the bachelor's degree in electronics and telecommunications from the Higher Pedagogical and Technical Institute of Kinshasa (ISPT-Kinshasa), Democratic Republic of Congo, in 2008, and the master's degree in electrical engineering (instrumentation and digital systems) and the Ph.D. degree in electrical engineering from the University of Guanajuato, in 2018 and 2022, respectively. His research interests include biological signal processing, robotics, artificial intelligence, embedded systems, image processing, brain–computer interfaces, deep learning, telecommunications, and bio-inspired systems. Currently, he is collaborating with the Telematic and Digital Signal Processing Research Groups, Department of Electronics Engineering, University of Guanajuato, Mexico.

CARLOS HUGO GARCÍA-CAPULÍN received the B.S. degree in electronics engineering from the Instituto Tecnológico de Ciudad Madero, in 1998, the master's degree in electrical engineering from the Universidad de Guanajuato, in 2004, and the Ph.D. degree in optics from the Centro de Investigaciones en Óptica, A. C., in 2014. He is currently a full-time Professor at the Department of Electronics, Universidad de Guanajuato. He has authored several journals and conferences proceedings papers. His research interests include computational intelligence, evolutionary algorithms, bio-inspired computing and robotics, reconfigurable electronics, and parallel computing.

HORACIO ROSTRO-GONZÁLEZ received the B.S. degree in electronic engineering from the Instituto Tecnologico de Celaya, Mexico, in 2003, the M.E. degree (Hons.) in electrical engineering from the Universidad de Guanajuato, Mexico, in 2006, and the D.Sc. degree in computational neuroscience from the University of Nice Sophia Antipolis, France, in 2011. From 2011 to 2012, he was a Research Fellow of the University of Cyprus, where he was involved in the FP7 EU SCANDLE Project. Since 2012, he has been a Full Professor with the Department of Electronics, Universidad de Guanajuato. He has authored around 40 journals and conferences proceeding papers. His current research interests include reconfigurable electronic, neuromorphic engineering, computational neuroscience, bio-inspired algorithms, parallel computing, and signal processing.

JUAN PRADO-OLIVAREZ received the Ph.D. degree in instrumentation and micro-electronics from Henri Poincaré University, Nancy I, France. Currently, he is working as a Teacher and a Researcher with the Department of Electrical and Electronics, Technological Institute of Celaya. His research interest includes bioelectronics, particularly electrical impedance spectroscopy.

MARCOS GUTIÉRREZ-LÓPEZ received the master's degree in engineering sciences from the Technological Institute of Celaya, in 2015, and the Doctor of Engineering Sciences degree from the National Technological Institute of Mexico, Campus Celaya, in 2019. He is currently a Professor at the National Technological Institute of Mexico. He has recently taught at the Tecnológico Nacional de México, Campus Morelia, in mechatronic engineering and has collaborated in the instrumentation of a wearable analyzer bioimpedance with the Department of Electronic Engineering. His research interests include bioimpedance studies in the medical area stand out and the classification of breast biopsy images using neural networks and medical instrumentation in general.

ALEJANDRO ISRAEL BARRANCO-GUTIÉRREZ (Member, IEEE) received the joint M.S. degree in mechatronics from CICATA-IPN, Mexico, in 2002, and Meijo University, Japan, the B.S. degree in telematics from the Instituto Politecnico Nacional, UPIITA, Mexico, in 2003, and the Ph.D. degree in advanced technology from CICATA-IPN, in 2010. He was a Professor at ICEL University, Mexico, from 2006 to 2011. He worked as an Assistant Professor with the Tláhuac Institute of Technology (CDMX), from 2008 to 2014. He has been a Professor with the Tecnológico Nacional de México en Celaya, Mexico, since 2014. His research interests include computer vision, fuzzy logic, and autonomous cars.