WILEY | Hindawi

Research Article

Computational Models-Based Detection of Peripheral Malarial Parasites in Blood Smears

Amal H. Alharbi,[1] **C. V. Aravinda** [iD],[2] **Jyothi Shetty,**[2] **Mohamed Yaseen Jabarulla** [iD],[3] **K. B. Sudeepa,**[2] **and Sitesh Kumar Singh** [iD] [4]

[1]*Department of Computer Sciences, College of Computer and Information Sciences,*
 Princess Nourah bint Abdulrahman University, P.O. Box 84428, Riyadh 11671, Saudi Arabia
[2]*NITTE Deemed to Be University, Mangalore, India*
[3]*School of Electrical Engineering and Computer Science, Gwangju Institute of Science and Technology (GIST),*
 Gwangju, Republic of Korea
[4]*Department of Civil Engineering, Wollega University, Nekemte, Oromia, Ethiopia*

Correspondence should be addressed to Sitesh Kumar Singh; sitesh@wollegauniversity.edu.et

The most common human parasite as per the medical experts is the malarial disease, which is caused by a protozoan parasite, and Plasmodium falciparum, a common parasite in humans. A microscopist with expertise in malaria diagnosis must conduct this complex procedure to identify the stages of infection. This epidemic is an ongoing disease in some parts of the world, which is commonly found. A Kaggle repository was used to upload the data collected from the NIH portal. The dataset contains 27558 samples, of which 13779 samples carry parasites and 13779 samples do not. This paper focuses on two of the most common deep transfer learning methods. Unlike other feature extractors, VGG-19's fine-tuning and pretraining made it an ideal feature extractor. Several image classification models, including VGG-19, have been pretrained on larger datasets. Additionally, deep learning strategies based on pretrained models are proposed for detecting malarial parasite cases in the early stages, in addition to an accuracy rating of 98.34* 0.51%.

1. Introduction

The leading cause of infection in all parts of the world is malaria, a deadly disease. The rapid consumption and high mortality rate of this epidemic condition have been documented throughout history. The global death toll from malaria was estimated at 4,29,000. In 2015, there were an estimated 3,03,000 children under the age of five years, and prenatal women are the most at risk of death [1]. By detecting this disease at an early stage, the death rate can be reduced and prevented. Researchers face a challenge in providing the most accurate parasite detection in the shortest amount of time, cost, and effort. During the last few decades, this visual inspection has played a vital role in being a tool in the health check field

for decision making. A thick blood smear can identify malaria parasites in blood samples [2]. This shows that it is close to eleven times more sensitive than a thin blood smear for the rapid detection of parasites. It is often used to test the development stages to create blood smears placed on a microscope glass slide. To confirm malaria infection, a pathologist uses a light microscope to identify changes in the size, shape, and perception of various RBCs. The pathologist's understanding of the disease depends on the accuracy of a plasmodium microscopic report. This technique is arduous and inefficient due to uncertainty, which can cause erroneous and contradictory diagnoses, as well as inappropriate medication and, in rare situations, the death of the patient and specimens of infected and noninfected microbes.

A mechanism for detecting malaria infection has been proposed using supervised learning. The methods for detecting malaria parasites were demonstrated by using in vitro culture image samples [3]. The samples developed in a laboratory do not contain substances other than WBCs, platelets, or parasites. When considering the severity of malaria depending on the number of deaths caused by the disease, it is reasonable to accept possible minor errors caused by an automated method during execution [4, 5]. Since the advent of deep learning techniques, feature extraction has been made far more efficient than traditional methods. Because deep learning methods still require trained experts and advanced techniques for calculating disease prediction, most of these methods still require efficient feature extraction optimization. There are many layers and levels of nonlinear mapping in CAD schemes based on ANN architecture. As a result of the layer-wise network of hidden layers, gradient-based optimization produces poor results [6, 7]. Microscopic diagnosis requires extensive training, experience, and skills. In rural areas where malaria is prevalent, manual microscopy has not proven to be an effective screening tool when performed by nonexperts [8–10]. This model was intended to enhance the model's performance by modifying the network architecture and hyper-tuning the features to achieve a better-performing model. To determine the key features in the future, this paper focuses on the network architecture. The basic VGG-19 model obtains 85% accuracy, but after fine-tuning the model and applying the data augmentation technique to the training dataset, it can attain 97.14%.

The following is the sequence under which this manuscript is structured. Section 2 represents the related work and data acquisition. Section 3 illustrates the proposed deep transfer learning methodology. Section 4 discusses the findings. Section 5 concludes the paper.

2. Related Work

There are several methods of identifying parasite RBCs. Purwar et al. preprocessed by utilizing local histograms [11]. The Hough transform and morphological operations are used for segmentation and classification of diseased and clean cells. Di Ruberto et al. employed a statistical k-means clustering algorithm [12]. Using thresholding, Ritter and Cooper [13] have segmented cells, separated overlaps, and modified division lines according to Dijkstra's algorithm. Diaz et al. [14] applied efficiency to develop templates from parasite-stained images, which would then be used to classify each cell's infection life stages. RBCs can be segmented from the background in blood images using a variety of techniques. Díaz et al. [15] determined whether the RBCs were infected by the parasite from the host or not by separating them. Savkare et al. [16] and Ross et al. [17] analyzed grayscale images using k-means and k-medians to define the overall clusters into two parts. The significance of supervised learning is to identify classes of sample data. These disease stages were identified with a blend of ML algorithms. Krizhevsky et al. [18] built a well-known CNN architecture, that is, the ALEX net, to compete with ImageNet and bagged awards. Quinn et al. [19]

compared the CNN with a classifier using trees, and the accuracy rate was determined by analyzing the middle of the operating region. To identify blood cells on blood smear images, Chowdhury et al. [20] adopted a CNN approach to detect infections that were affected by the blood smears. Raviraja et al. [6] analyzed pretrained CNN models for malaria detection in blood cell images, namely, DenseNet121, VGG-16, Alexnet, ResNet50, FastAI, and ResNet101, Meng et al. [21]. With a high precision of 97.5%, ResNet50 excelled over the other CNN models [22]. To conclude, many deep learning algorithms for detecting malaria using cell images have been presented by Yue [23]. Many of them used large pre-trained CNN models to enhance the accuracy of classification, whereas others used customized CNNs to minimize the computational time [24].

The implications of substantial quantities of improperly classified data in medical image classification are catastrophic, and the objective of proposing a medical diagnosis tool is wrecked. In addition to efficiency, additional parameters such as F1 score [25, 26], area under the curve (AUC) [27, 28], sensitivity [29, 30], and specificity [31–33] are vital in analyzing various approaches. A random sample of cell images infected/not infected with malaria is shown in Figure 1.

2.1. Data Acquisition. The data was obtained from the National Institutes of Health portal and uploaded to a Kaggle repository. There are 27558 cell images in the dataset, out of which 13779 are malaria-infected cell images and another 13779 are malaria-free cell images. The random samples were collected and separated into three sets: training, testing, and validation. Next, 8000 images were deployed for training and 3000 images for validation for each class. These 11000 images have been used to train models. Subsequently, these were used along with the remaining 2779 images for each class as test data to evaluate the proposed models to perform on images that have never been seen before, because the original dataset included images with different dimensions, and it was scaled to equal dimensions before splitting the data. Most of the images were scaled to 128×128 pixels with three channels of RGB. Having all images equal will allow the neural networks to learn more quickly and with minimal mistakes in the future. The outcomes obtained excelled all existing methods, so data augmentation was used to improve the results. A few of the images in the class folders at the beginning are significantly different than the ones at the end. As a result, this was used to sample the data at random, as this will allow the proposed models to learn more diversified features from both classes, which will reduce overfitting and make our model more extensible to data.

3. Proposed Model

Morphological image processing is used to eliminate noise and recreate object features (see Figure 2). To achieve deep feature extraction and transfer learning, the convolutional layers are frozen. To classify RBCs, a fine-tuned pretrained CNN model with image augmentation is proposed (see Figure 3).
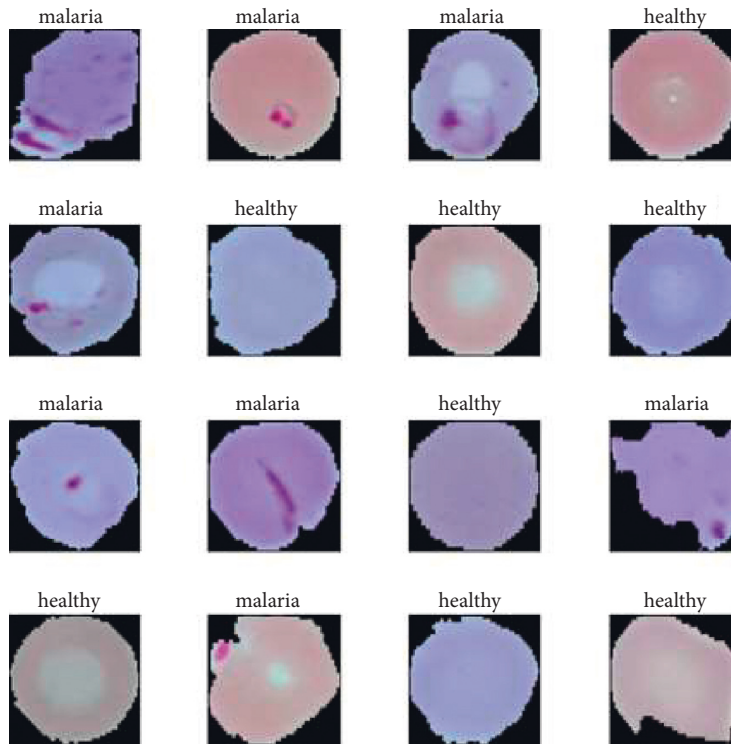
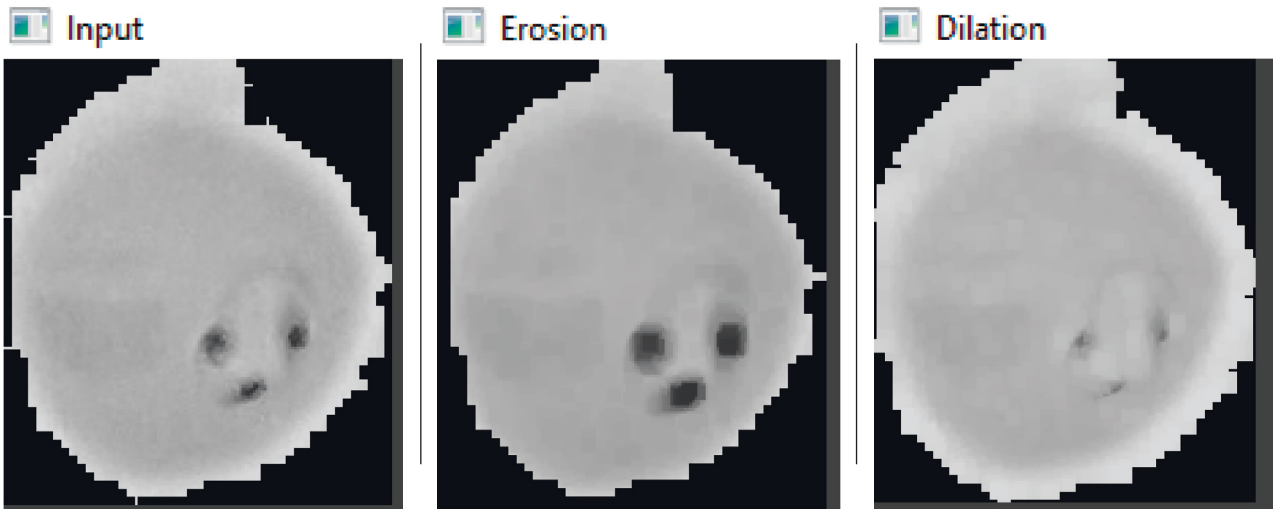FIGURE 1: A random sample of cell images infected/not infected with malaria [2].

FIGURE 2: Morphological filters applied to malaria cells.

3.1. Transfer Learning. The paths, according to K. Fukushima, are the beginning of a deep convolutional neural network architecture. The idea has been around for a while, but due to the lack of efficient high computational power, it has yet to gain momentum. In recent times, graphics processing units have advanced significantly toward high-performance computing technologies. Computational intelligence techniques have gained prominence as a result of their high possibility. CNN is the most popular type of this approach, which is composed of the layers described below.

Three main CNN layers were used in this algorithm, namely, "convolution," "pooling," and "fully connected layers." Figure 3 exhibits a schematic representation of a model with one conventional layer and one maximum layer. In the feature map, activation functions are used to increase the nonlinearity of the network ReLU. A neuron with the sigmoid activation function appears in the output layer of the model. When all negative values in the activation map are replaced with zero, ReLU activation completely cancels out all negative values. A binary classification model is built using sigmoid activation with a loss of function of binary
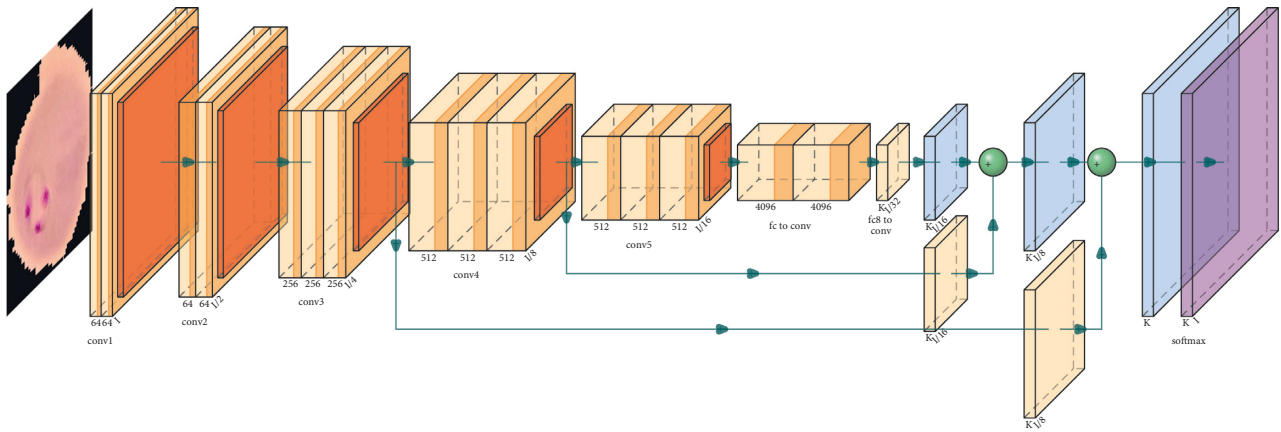
FIGURE 3: Proposed CNN-model.

cross-entropy. It has a learning rate of 0.01. In this case, the function gives a value between 0 and 1. Following the compilation of the dataset, the model will be trained based on the inputs received from the training samples, which will map the inputs to the outputs. It is a matter of choosing a set of weights that is best suited for resolving these problems.

3.2. Predictable Method. The initial model is trained by this network model using traditional methods of training for a given number of epochs. Model accuracies of 99.80% during training and 95.60% during validation were unchanged. Figures 4 and 5 show the exactness and failure graphs of the model. AUC score, specificity, sensitivity, and test accuracy were used to evaluate the performance. Table 1 depicts an implementation of the test set, whereas Table 2 exhibits CNN model architecture. The input layer $a = i1$ and output layer $io1 = 3$ and $p1 = 0$, input layer c2d, output layer $co1 = 32$ and $p1 = 896$, max input layer mp2d and output layer $mo2 = 32$, and $p1 = 0$. The input layer c2d1, output layer $c2do1 = 64$ and $p2 = 18496$, and the input layer mp2d1 and output layer $mpo1 = 64$ and $p2 = 0$. The input layer c2d2, output layer $c2o2 = 128$, and $p3 = 73856$. The input layer $= m2d2$, output layer $= m2do2 = 128$, and $p4 = 0$. The input layer flattened layer = Fl, output layer $flo = 28800$, and $p4 = 0$. Input the dense layer dl, output layer $det5 = 512$, and $p5 = 14746112$. The input layer dropout = dot, output layer $dot5 = 512$, and $p5 = 0$. The input layer dns_1 and output layer $dnst5 = 512$ and $p6 = 262656$.

3.3. Pretrained CNN Model. Figures 6 and 7, respectively, demonstrate the accuracy model and performance for the VGG-19 pretrained model and weights as displayed in Table 3. The convolutional layers are divided into sixteen layers, and there are $3 * 3$ convolutional filters. The input layer b1-c1, output layer $t1 = 64$ and parameter $p1 = 1792$, the input layer b1-c2, output layer $t1 = 64$ and parameter $p1 = 3698$, and b1_pool and output $t1 = 64$ and parameter $p1 = 0$. The input layer b2-c1, the output layer $t2 = 128$, and the parameter $p2 = 73856$. The input layer b2-c2, the output layer $t1 = 128$, and parameters $p1 = 147584$ and b2_pool and
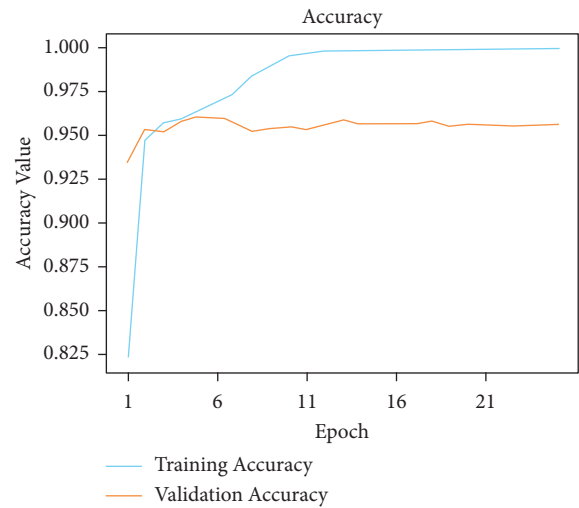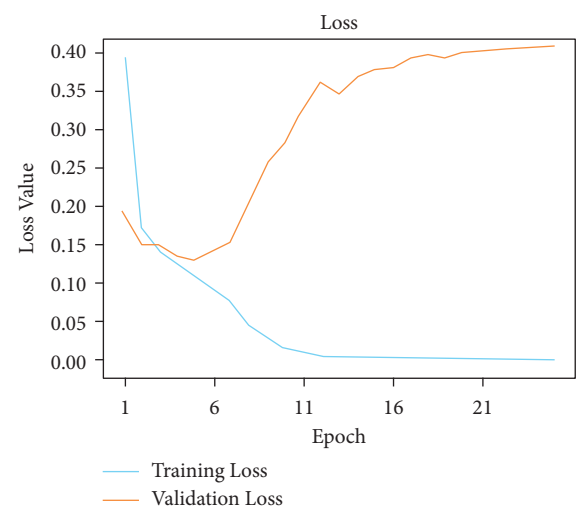

FIGURE 4: Model accuracy performance of CNN model.

FIGURE 5: Model loss performance of CNN model.

output $t2 = 1284$ and parameter $p2 = 0$. The input layer b3-c1, output layer $t3 = 256$, and parameters $p3 = 295168$, b3_c2 output layer $t3 = 256$, and $p3 = 590080$, b3-c3 and b3-c4

TABLE 1: Performance test.

Metrics	Performance (%)
Testing-accuracy	95.56
F1 Score	96.45
A∪C score	95.45
Sensitivity	96.65
Specificity	95.25

TABLE 2: Convolution neural network model.

Layer	Output	Parameter
i_1	[(N, 125, 125, io1)]	P0
c2d	(N, 125, 125, co1)	P1
mp2d	(N, 62, 62, mpo2)	p01
c2d_1	(N, 62, 62, c2do1)	P2
mp2d _1	(N, 31, 31, mp2o1)	p02
c2d2	(N, 31, 31, c2o2)	P3
m2d_2	(N, 15, 15, m2do)	p03
Fl	(N, flo)	P4
Dl	(N, det5)	14746112
Dt	(N, dot5)	p05
dns_1	(N, dnst5)	262656
Dot	(N, dot5)	d05
dense_2	(N, 1)	513
Total	—	15,102,529
Trainable	—	15,102,529
Non-trainable	—	0

Note: _1 = "input1", c2d = "convolutional2d", mp2d = " max_poolint2d", c2d_1 = "convolutional2d1", mp2d _1 = "max_poolint2d1", c2d2 = "convolutional2d2 ", m2d_2 = "max_poolint2d2 ", f1 = "Flatten Layers", d1 = " dropout", dt = " dense1", dns_1 = "dense2", dot = "dropout1". io1 = "3", co1 = " ", mpo2 = " 32", c2do1 = "64 ", mp2o1 = "64 ", c2o2 = "128 ", m2do = " 128",flo = " 28800", det5 = "512 ", dot5 = "512 ", dnst5 = " 512", p0 = "0", p1 = " 896", p01 = "0", p2 = " 18496", p02 = "0", p3 = " 73856", p03 = "0", p4 = " 0", p05 = "0", d05 = "0".
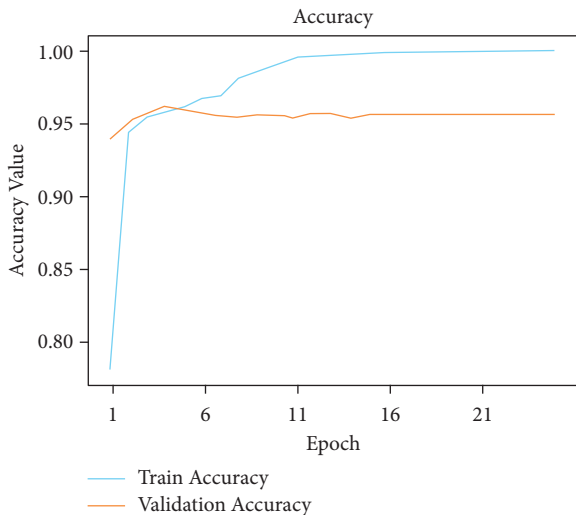

FIGURE 6: Accuracy performance of modified CNN model.

FIGURE 7: Loss performance of modified CNN model.

output layer $t3 = 256$, and output layer $p4 = 590080$. The b3-pool and output $t3 = 256$ and $p3 = 0$. The input layer b4-c1, the output layer $t4 = 512$, and $p4 = 1180160$. The input layer b4-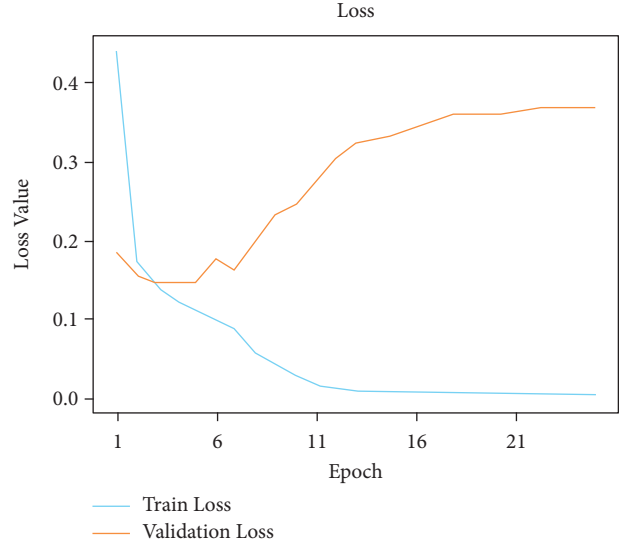c2, b4-c3, b4-$c4 = 512$ and $p4 = 2359808$. The input layer b4-pool and $t4 = 512$ and the output layer $p4 = 0$. The input layer b5-c1, b5-c2, b5-c3, b5-c4, and $t5 = 512$, and the output layer $p5 = 2359808$. The input layer fla-1 and the output layer $t5 = 4608$ and $p5 = 0$. The input layer dse3 and output layer dset = 512 and $p5 = 2359808$. The input layer dtt2 and the output layer dtt5 = 512 and $p5 = 0$. The input layer dse4 and output layer dset5 = 512 and $p5 = 262656$. In addition to that, there are max pooling filters for downscaling and two fully connected hidden layers with 4096 units each. The remaining dense layer consists of one thousand units, each representing one of the image categories in ImageNet. The dense layer is fully connected, so the last three layers are skipped, and the five layers are concentrated to use the vgg19 model for feature extraction. This model was built from scratch using the original datasets that include all 19 trainable layers. The ReLU served as the activation function for the network. The Adam optimization was used to create the loss function. The model was built using transfer learning and using pre-trained frozen layers. Later, this model was used to generate the output of the images. This was implemented using the sigmoid activation function as a simple feature extractor by freezing all five convolutional blocks to prevent the weights from moving across epochs. This third model is fine-tuned and is built by freezing the first three blocks from the image net and then training blocks four and five from the malarial datasets. To fine-tune the VGG-model, blocks 4 and 5 were changed so that their weights are updated each time the model is evaluated. These preprocessing strategies were applied to this model, which includes normalization, data augmentation, and standardization. A sigmoid activation function with two methods was applied to solve the classification problem to gain an output of 1 for infected and 0 for healthy.

3.4. Image Augmentation with a Fine-Tuned Pretrained Model. In Figure 8, the existing images from the training samples were reworked and transformed to create a new,

TABLE 3: Pretrained convolution neural network model (VGG-19).

Layer	Output	Parameter
input_2	[(N, 125, 125,3)]	0
b1-c1	(N, 125, 125, t1)	$P1$
b1-c2	(N, 125, 125, t1)	$P1$
b1-pool	(N, 62, 62, t1)	0
b2-c1	(N, 62, 62, t2)	$P2$
b2-c2	(N, 62, 62, t2)	$P2$
b2-pool	(N, 31, 31, t3)	0
b3-c1	(N, 31, 31, t4)	$P3$
b3-c2	(N, 31, 31, t4)	$P3$
b3-c3	(N, 31, 31, t4)	$P3$
b3_c4	(N, 31, 31, t4)	$P3$
b3-pool	(N, 15, 15, t4)	0
b4-c1	(N, 15, 15, t5)	$P4$
b4-c2	(N, 15, 15, t5)	$P4$
b4-c3	(N, 15, 15, t5)	$P4$
b4-c4	(N, 15, 15, t5)	$P4$
b4-pool	(N, 7, 7, t5)	0
b5-c1	(N, 7, 7, t5)	$P5$
b5-c2	(N, 7, 7, t5)	$P5$
b5-c3	(N, 7, 7, t5)	$P5$
b5-c4	(N, 7, 7, t5)	$P5$
b5-pool	(N, 3, 3, t5)	0
fla_1	(N, t5)	0
Dse-3	(N, dset5)	$P5$
Dt-2	(N, dttt5)	0
Dse-4	(N, dset5)	262656
Dt-3	(N, dt5)	0
Dse-5	(N, 1)	513
Total params:		22,647,361
Trainable params:		2,622,977
Nontrainable:		20,024,384

Note: b1-c1 and b1-c2 = " block1_conv1 and block1_conv2", b2-c1 and b2-c2 = " block2_conv1 and block2_conv2", b3-c1 and b3-c2 and b3-c3 and b3-c4 = " block3_conv1 and block3_conv2 and block3_conv3 and block3_conv4", b5-c1 and b5-c2 and b5-c3 and b5-c4 and b5-c5 = " block5_conv1 and block5_conv2 and block5_conv3 and block5_conv4". fla1 = " flatten_1" $t5 = 4608$. dse3, dse4, dse5 = " dense_3, dense_4, dense_5", dt2 = " droupout_2.
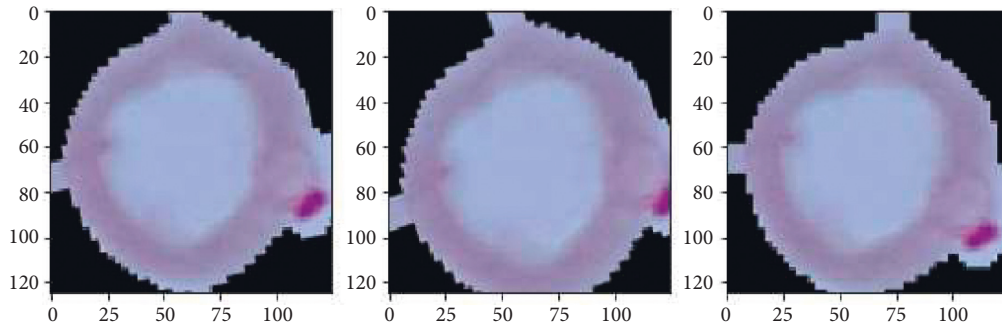

FIGURE 8: Sample augmented images.

modified version of the originals because of rotation, shearing, translation, zooming, and so on. Figures 6 and 7 illustrate how the random transformation, model accuracy, and loss of the model are what are required to obtain the same images every time. The augmentation of the model accuracy is shown in Figures 9 and 10.

The confusion matrix (FN) is used to evaluate the number of positive and negative predictions as shown in Figure 11. The confusion matrix is used to determine whether a prediction is a truly positive or true negative.

4. Discussion

Individual red blood cell smear images are investigated to evaluate if they are infected or healthy. The study comprises a range of pretrained convolutional neural networks with transfer learning that are fine-tuned and registered on the malaria dataset. On VGG-19 and Transfer Learning, research shows that different preprocessing approaches like normalization and scaling do not affect model performance, however, the data augmentation technique has shown
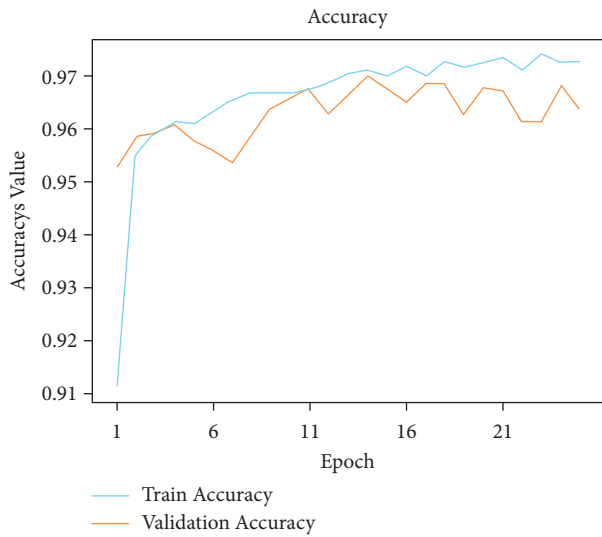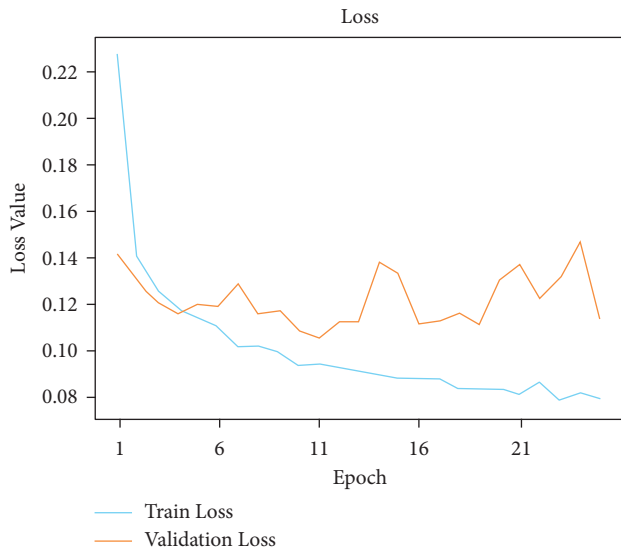
Accuracy

FIGURE 9: Augmentation accuracy results.

Loss

FIGURE 10: Augmentation loss results.

FIGURE 11: Confusion matrix results.

TABLE 4: Confusion matrix-based analyses.

Models	Accuracy	F1 score	Precision	Recall
Basic CNN	0.9397 ± 0.23	0.9397 ± 0.13	0.9397 ± 0.19	0.9397 ± 0.27
VGG-19 frozen	0.9486 ± 0.13	0.9482 ± 0.12	0.9456 ± 0.15	0.9480 ± 0.12
VGG-19 fine-tuned	0.9704 ± 0.06	0.9640 ± 0.06	0.9740 ± 0.07	0.9700 ± 0.03

TABLE 5: The performance report of the model classification.

	Precision	Recall	F1 score	Support
Healthy sample	0.97	0.96	0.96	4085
Malaria-sample	0.96	0.96	0.95	4173
Micro-average	0.97	0.97	0.97	8158
Macro-average	0.97	0.97	0.97	8158
Weighted-average	0.97	0.97	0.97	8158

wonderful strategy that can be used to create promising results, according to the research and the performance analyses depicted in Table 5.

5. Conclusion

The deep learning neural network model was applied to improve the model's performance. It was shown that standardization and normalization had less impact on classification. The use of data augmentation improved the model performance and yielded positive results. Models of VGG-19 and ImageNet were derived from the initial concept using the combination of transfer learning and parameter tuning. To determine the key features, this paper has focused on the network architecture. This was intended to enhance the model's performance by modifying the network architecture and hyper-tuning the features to achieve a better-performing model.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

References

[1] C. Boschi-Pinto, T. R. Dilip, and A. Costello, "Association between community management of pneumonia and diarrhoea in high-burden countries and the decline in under-five

encouraging outcomes. The basic VGG-19 model obtains 85% accuracy, but after fine-tuning the model and applying the data augmentation technique to the training dataset, it can attain 97.14%, as shown in Table 4. Transfer learning is a

mortality rates: an ecological analysis," *BMJ Open*, vol. 7, no. 2, Article ID e012639, Feb 2017.

[2] M. Poostchi, K. Silamut, R. J. Maude, S. Jaeger, and G. Thoma, "Image analysis and machine learning for detecting malaria," *Translational Research: The Journal of Laboratory and Clinical Medicine*, vol. 194, pp. 36–55, Jan 2018.

[3] D. K. Das, M. Ghosh, M. Pal, A. K. Maiti, and C. Chakraborty, "Machine learning approach for automated screening of malaria parasite using light microscopic images," *Micron*, vol. 45, pp. 97–106, 2013.

[4] V. V. Makkapati and R. M. Rao, "Segmentation of malaria parasites in peripheral blood smear images," in *Proceedings of the 2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1361–1364, Taipei, Taiwan, April 2009.

[5] N. Mustare, V. Kaveri Sreelathareddy, and V. Sreelathareddy, "Development of automatic identification and classification system for malaria parasite in thin blood smears based on morphological techniques," in *Proceedings of the 2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI), 2017 Development of Automatic Identification and Classification System for Malaria Parasite in Thin Blood Smears Based on Morphological Techniques*, pp. 3006–3011, Chennai, India, September 2017.

[6] S. Raviraja, S. S. Osman, and Kardman, "A novel technique for malaria diagnosis using invariant moments and by image compression," in *Proceedings of the 4th Kuala Lumpur International Conference on Biomedical Engineering 2008*, N. A. Abu Osman, F. Ibrahim, W. A. B. Wan Abas, H. S. Abdul Rahman, and H. N. Ting, Eds., pp. 730–733, Springer Berlin Heidelberg, Kuala Lumpur, Malaysia, January 2008.

[7] J. Somasekar, A. Rama Mohan Reddy, and L. Sreenivasulu Reddy, "An efficient algorithm for automatic malaria detection in microscopic blood images," in *Global Trends in Information Systems and Software Applications*, P. V. Krishna, M. R. Babu, and E. Ariwa, Eds., Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 431–440, 2012.

[8] V. Nair and G. E. Hinton, "3d object recognition with deep belief nets," in *Proceedings of the 22nd International Conference on Neural Information Processing Systems*, pp. 1339–1347, NIPS'09, Curran Associates Inc, Red Hook, NY, USA, 2009.

[9] A. Prathyakshini and C. V. Akshaya, "Classification and clustering of infected leaf plant using K-means algorithm," in *Cognitive Computing and Information Processing*, T. Nagabhushan, V. N. M. Aradhya, P. Jagadeesh, S. Shukla, and M. L. Chayadevi, Eds., Springer Singapore, Singapore, pp. 468–474, 2018.

[10] I. Arel, D. C. Rose, and T. P. Karnowski, "Deep machine learning - a new Frontier in artificial intelligence research [research Frontier]," *IEEE Computational Intelligence Magazine*, vol. 5, no. 4, pp. 13–18, 2010.

[11] Y. Purwar, S. L. Shah, G. Clarke, A. Almugairi, and A. Muehlenbachs, "Automated and unsupervised detection of malarial parasites in microscopic images," *Malaria Journal*, vol. 10, no. 1, p. 364, 2011.

[12] C. Di Ruberto, A. Dempster, S. Khan, and B. Jarra, "Automatic thresholding of infected blood images using granulometry and regional extrema," in *Proceedings of the 15th International Conference on Pattern Recognition. ICPR-2000*, vol. 3, p. 441p. 441, September 2000.

[13] N. Ritter and J. Cooper, "Segmentation and border identification of cells in images of peripheral blood smear slides,"vol. 62, pp. 161–169, in *Proceedings of the Thirtieth Australasian Conference on Computer Science. Author-Anonymous, 2007*, vol. 62, pp. 161–169, Australian Computer Society, Inc, Australia, August 2007.

[14] G. Díaz, F. A. González, and E. Romero, "A semi-automatic method for quantification and classification of erythrocytes infected with malaria parasites in microscopic images," *Journal of Biomedical Informatics*, vol. 42, no. 2, pp. 296–307, 2009.

[15] G. Díaz, F. Gonzalez, and E. Romero, "Infected cell identification in thin blood images based on color pixel classification: comparison and analysis," in *Proceedings of the Congress on Pattern Recognition, 12th Iberoamerican Conference on Progress in Pattern Recognition, Image Analysis and Applications*, pp. 812–821, Springer-Verlag, Berlin, Heidelberg, 2007.

[16] S. S. Savkare, A. S. Narote, and S. P. Narote, "Automatic blood cell segmentation using k-mean clustering from microscopic thin blood images," in *Proceedings of the Third International Symposium on Computer Vision and the Internet*, pp. 8–11, VisionNet'16, Association for Computing Machinery, New York, NY, USA, 2016.

[17] N. E. Ross, C. J. Pritchard, D. M. Rubin, and A. G. Dusé, "Automated image processing method for the diagnosis and classification of malaria on thin blood smears," *Medical, & Biological Engineering & Computing*, vol. 44, no. 5, pp. 427–436, Apr 2006.

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, May 2017.

[19] J. A. Quinn, R. Nakasi, P. K. B. Mugagga, P. Byanyima, W. Lubega, and A. Andama, "Deep convolutional neural networks for microscopy-based point-of-care diagnostics," 2016, http://arxiv.org/abs/1608.02989.

[20] A. B. Chowdhury, J. Roberson, A. Hukkoo, S. Bodapati, and D. J. Cappelleri, "Automated complete blood cell count and malaria pathogen detection using convolution neural network," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1047–1054, 2020.

[21] L. Meng, C. V. Aravinda, K. R. Uday Kumar Reddy, T. Izumi, and K. Yamazaki, "Ancient Asian character recognition for literature preservation and understanding," in *Digital Heritage. Progress in Cultural Heritage: Documentation, Preservation, and Protection*, M. Ioannides, E. Fink, R. Brumana et al., Eds., Springer International Publishing, Cham, pp. 741–751, 2018.

[22] H. Shalini and C. V. Aravinda, "An IoT-based predictive analytics for estimation of rainfall for irrigation," in *Advances in Artificial Intelligence and Data Engineering*, N. N. Chiplunkar and T. Fukao, Eds., Springer Singapore, Singapore, pp. 1399–1413, 2021.

[23] X Yue, H Li, K Saho, K Uemura, C V Aravinda, and L Meng, "Machine learning based apathy classification on Doppler radar image for the elderly person," *Procedia Computer Science*, vol. 187, pp. 146–151, 2021, https://www.sciencedirect.com/science/article/pii/S1877050921008255.

[24] A. C V, L. Meng, A. Masahiko, U. Kumar, and A. Prabhu, "A complete methodology for k historical character recognition using multiple features approach and deep learning model," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 8, 2020.

[25] D. Jie, G. Zheng, Y. Zhang, X. Ding, and L. Wang, "Spectral kurtosis based on evolutionary digital filter in the application

of rolling element bearing fault diagnosis," *International Journal of Hydromechatronics*, vol. 4, no. 1, pp. 27–42, 2021.

[26] Xu, C. Li, and C. Li, "Electric window regulator based on intelligent control," *Journal of Artificial Intelligence and Technology*, vol. 1, pp. 198–206, 2021.

[27] M. Kaur, D. Singh, V. Kumar, B. B. Gupta, and A. A. Abd El-Latif, "Secure and energy efficient-based E-health care framework for green internet of things," *IEEE Transactions on Green Communications and Networking*, vol. 5, no. 3, pp. 1223–1231, 2021.

[28] S. C. Mondal, P. l. C. Marquez, and M. O. Tokhi, "Analysis of mechanical adhesion climbing robot design for wind tower inspection," *Journal of Artificial Intelligence and Technology*, vol. 1, pp. 219–227, 2021.

[29] M. Kaur and D. Singh, "Multi-modality medical image fusion technique using multi-objective differential evolution based deep neural networks," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 2, pp. 2483–2493, 2021.

[30] A. Balakrishna and P. K. Mishra, "Modelling and analysis of static and modal responses of leaf spring used in automobiles," *International Journal of Hydromechatronics*, vol. 4, no. 4, pp. 350–367, 2021.

[31] P. K Singh, "Data with non-Euclidean geometry and its characterization," *J. Artif. Intell. Technol.* vol. 2, pp. 3–8, 2022.

[32] D. Singh, V. Kumar, M. Kaur, M. Y. Jabarulla, and H.-N. Lee, "Screening of COVID-19 suspected subjects using multi-crossover genetic algorithm based dense convolutional neural network," *IEEE Access*, vol. 9, no. 2021, pp. 142566–142580, 2021.

[33] T. V. Hahn and C. K. Mechefske, "Self-supervised learning for tool wear monitoring with a disentangled-variational-autoencoder," *International Journal of Hydromechatronics*, vol. 4, no. 1, pp. 69–98, 2021.