*Journal of Science and Technology*
*ISSN: 2456-5660 Volume 8, Issue 12 (Dec -2023)*
*www.jst.org.in* *DOI:https://doi.org/10.46243/jst.2023.v8.i12.pp185-198*

# Machine Learning-based Detection of Malaria Infection through Blood Sample Analysis

N. Teja[1], K. Vyshnavi[2], M. Pavana Sri[2], K. Bharathi[2]
[1]Assistant Professor,[2]UG Students, Department of Information Technology
[1,2]Malla Reddy Engineering College for Women, Maisammaguda, Dhulapally, Kompally,
Secunderabad-500100, Telangana, India.

***To Cite this Article***

N. Teja, K. Vyshnavi, M. Pavana Sri, K. Bharathi**, "Machine Learning-based Detection of Malaria Infection through Blood Sample Analysis "** *Journal of Science and Technology, Vol. 08, Issue 12 - Dec 2023, pp185-194*

***Article Info***

*Received: 13-11-2023 Revised: 23 -11-2023 Accepted: 03-12-2023 Published: 13-12-2023*

## ABSTRACT

Malaria, a life-threatening disease caused by Plasmodium parasites transmitted through infected mosquitoes, remains a significant public health concern in many regions worldwide. Early and accurate detection of malaria infection is crucial for timely treatment and disease management. The automated malaria detection system can be integrated into portable diagnostic devices, enabling healthcare professionals to perform rapid and accurate malaria tests in remote or resource-limited settings. The system can assist researchers and health organizations in tracking malaria prevalence and monitoring its spread, contributing to epidemiological studies and efficient resource allocation. Conventional methods for malaria detection involve manual examination of blood smears under a microscope by trained technicians. Although reliable, this process is time-consuming, labor-intensive, and dependent on the expertise of the microscopist. The regression-based examination of blood smears introduces the potential for errors, leading to false-negative or false-positive results. In recent years, machine learning-based approaches have shown promising results in automating the detection of malaria parasites through blood sample analysis. This work presents an advanced machine learning-based method for the automated detection of malaria infection, leveraging image processing techniques to achieve high accuracy and efficiency.

**Keywords:** malaria infection, blood samples, machine learning, epidemiological.

## 1. INTRODUCTION

Malaria infection is a widespread and potentially deadly disease caused by the Plasmodium parasite, transmitted to humans through the bite of infected female Anopheles mosquitoes. Diagnosis and monitoring of malaria often rely on the analysis of blood samples, which provides crucial insights into the presence and severity of the infection. When a blood sample is obtained from a patient suspected of having malaria, it undergoes a series of laboratory tests to confirm the diagnosis and assess the level of parasitic activity.The primary diagnostic method is the examination of a thin blood smear or a thick blood smear under a microscope. Thin blood smears are used to identify the Plasmodium species responsible for the infection, while thick blood smears are employed to quantify the number of parasites present in the blood. This information is vital for determining the severity of the disease and guiding treatment decisions. Additionally, molecular techniques like polymerase chain reaction (PCR) can be employed to confirm the presence of the parasite and, in some cases, differentiate between species with high accuracy.Blood sample analysis also allows for the evaluation of other important parameters such as hematocrit levels, which help in assessing anemia, a common complication of malaria. Moreover, serological tests can be performed to detect specific antibodies against Plasmodium antigens, providing information about previous exposure to the parasite and aiding in epidemiological studies. The timely and accurate analysis of blood samples is crucial in the management and control of malaria. Rapid and precise diagnosis enables healthcare providers to initiate appropriate treatment promptly, reducing the risk of severe complications and death. Furthermore, monitoring the parasite load in the blood over time allows healthcare professionals to gauge the effectiveness of treatment and make necessary adjustments. Therefore, blood sample analysis remains a cornerstone in the battle against malaria, contributing to both individual patient care and public health efforts to control and ultimately eliminate this devastating disease. The research motivation for developing and improving the analysis of malaria infection through blood sample images is multifaceted and has significant implications for both healthcare and epidemiology. Firstly, malaria remains a major global health concern, particularly in regions with limited access to healthcare resources. It is a leading cause of morbidity and mortality, especially among children and pregnant women. Timely and accurate diagnosis is essential for effective treatment and disease management. Therefore, researchers are motivated to enhance the existing diagnostic methods to ensure early detection and better patient outcomes. Secondly, microscopy-based diagnosis, while reliable, is labor-intensive and requires well-trained technicians, which can be scarce in resource-limited settings. The motivation here is to develop automated image analysis systems and artificial intelligence (AI) algorithms to assist in malaria diagnosis. These technologies have the potential to make diagnosis more accessible, faster, and less dependent on the availability of skilled personnel. This is crucial for expanding healthcare access and improving the reach of malaria diagnosis in remote or underserved areas. Furthermore, monitoring and understanding the epidemiology of malaria is essential for public health interventions, such as vector control and treatment distribution. Analyzing blood samples can provide valuable data on the prevalence of different Plasmodium species, the emergence of drug resistance, and changes in transmission patterns. Researchers are motivated to use image analysis to study the distribution and evolution of malaria parasites, enabling more informed policy decisions and targeted interventions. Lastly, as technology advances, the integration of image analysis and AI in malaria diagnosis aligns with the broader trend of leveraging digital health solutions to improve healthcare delivery. This motivates researchers to explore innovative approaches that combine biology, computer science, and medical expertise to revolutionize how malaria is diagnosed and managed, ultimately contributing to the global effort to eliminate this devastating disease. In summary, the research motivation for malaria infection analysis through blood sample images lies in its potential to enhance diagnosis, expand healthcare access, inform public health strategies, and leverage cutting-edge technology to combat a persistent global health threat.

## 2. LITERATURE SURVEY

According to the World Health Organization (WHO), malaria case rates (i.e., cases per 1000 population) fell from 82 in 2000 to 57 in 2019 but rose to 59 in 2020. The WHO reported that this unusual 2020 increase in malaria case rates was related to service supply disruptions during the

COVID-19 pandemic [1]. In fact, the number of malaria cases increased from 227 million in 2019 to 241 million in 2020, and the number of malaria deaths in 2020 was estimated at 627,000, a 12% increase from 2019 [2].

Moreover, in the case of malaria, the more severe problem is that the existing malaria diagnosis method relies on direct human observation, which takes much time for diagnosis, making it difficult to test many patients simultaneously. Additionally, there is a limitation in that diagnostic accuracy is greatly affected by variability between observers. In other words, the effectiveness of the conventional microscopic diagnosis is highly dependent on the expertise of parasitologists. Besides, it is common for parasitologists to work in resource-constrained environments without stringent systems to maintain their know-how or diagnostic quality [3]. This can often lead to erroneous diagnoses and inappropriate treatment, which can have fatal consequences [3-5].

There are several promising prior studies on the capabilities of ML-based techniques in detecting infectious diseases. For instance, using a machine learning framework, Colubri et al. [6] introduced an application that can predict the outcome of Ebola patients from early clinical symptoms. Smith and Kirby [7] described ML applications for analyzing different types of microbial image data, particularly progress in smear and plate interpretation.

Another notable study on ML-based infectious disease diagnosis is that of Das et al. [8], who developed a computer-aided malaria parasite characterization and classification based on light microscopy images of peripheral blood smears collected from 600 patients using an ML approach. Their proposed ML scheme applying the Bayesian approach provides 84.0% accuracy and 98.1% sensitivity by selecting the 19 most significant features, and the support vector machine (SVM) achieved 83.5% screening accuracy and 96.6% sensitivity with the 9 most significant features [8].

Similarly, there are other studies that have applied various machine learning methods to detect malaria parasites. Bibin et al. [9] proposed a deep belief network (DBN)-based trained model to classify 4100 peripheral blood smear images into parasitic or nonparasitic classes. The proposed method showed an F-score of 89.66%, a sensitivity of 97.60%, and a specificity of 95.92% [9]. Gopakumar et al. [10] used a customized CNN model operating on a focus stack of images for automated quantitative detection of Plasmodium falciparum malaria in blood smears. The detection accuracy of the CNN model was 97.06% sensitivity and 98.50% specificity [10].

Yang et al. [3] developed a method using a deep learning algorithm to detect malaria parasites in thick blood smear images, run on a smartphone. They trained and tested a deep learning method using 1819 thick smear images from 150 patients [3]. The study results showed the effectiveness of the CNN model in distinguishing positive (parasitic) image patches from negative image patches, with performance metrics of accuracy (93.46% ± 0.32%), precision (94.25% ± 1.13%), and negative predictive value (92.74% ± 1.09%) [3].

Especially in the case of the COVID-19 pandemic, Dandekar et al. [11] applied the neural network module of ML to develop a globally applicable COVID-19 diagnosis model to analyze and compare the role of quarantine control policies globally across the continents of Europe, North America, South America, and Asia. Dandekar et al. [11] also hosted quarantine diagnosis results from 70 countries around the world on a public platform: https://covid19ml.org/ (accessed on 15 March 2023).

One example of a notable literature review source for ML-based infectious disease diagnosis is the work of Baldominos et al. [12]. The study performed a computer-based systematic literature review in order to investigate where and how computational intelligence (i.e., different types of machine learning techniques) is being utilized to predict patient infection [12].

Deep learning, a specific subset of machine learning, is a computational processing system composed of artificial neural networks, heavily inspired by how biological nervous systems process information and make decisions [13]. Deep learning allows for incrementally learning complex input data features by going through the architecture's hidden layers [14]. That is, as the input data pass through hidden layers, the complexity of the input data is computed as a simpler and less abstract concept for the final output, which is the so-called nested hierarchical approach [14,15,16].

For more information on deep learning, see the work by Alzubaidi et al. [17], which presents essential overall aspects of deep learning and provides a clear image of deep learning in one review paper.

## 3. PROPOSED SYSTEM

The methodology leverages image processing and machine learning techniques to automate the detection of malaria parasites in blood sample images. It is a promising approach to improve the efficiency and accuracy of malaria diagnosis, particularly in resource-limited settings where access to skilled technicians may be limited. However, it's important to note that developing and fine-tuning the RFC model typically requires a substantial amount of labeled data and expertise in machine learning and image analysis. Additionally, the performance of the model should be rigorously evaluated to ensure its accuracy and reliability in real-world healthcare applications. Figure 4.1 shows the proposed system model. The detailed operation illustrated as follows:

Step 1: Image Processing: This is the initial step where you process the blood sample images. Image processing techniques may include preprocessing steps such as noise reduction, contrast enhancement, and image segmentation to isolate the relevant features (in this case, malaria parasites) from the background and other elements in the image. This step is essential for preparing the images for further analysis.

Step 2: Random Forest Classifier (RFC) Building: After image processing, the next step involves training a machine learning model, specifically a Random Forest Classifier (RFC). In this step, you would typically use a labeled dataset of blood sample images, where each image is associated with a known diagnosis (e.g., whether it contains malaria parasites or not). The RFC is trained to learn patterns and features in the images that distinguish between infected and uninfected samples. This classifier can handle complex relationships in the data and is capable of making predictions based on these learned patterns.

Step 3: RFC Prediction: Once the RFC model is trained, it can be used to predict whether new, unseen blood sample images contain malaria parasites or not. When a new blood sample image is input into the trained RFC, the model evaluates the image based on the patterns it has learned during training and produces a prediction. This prediction can help automate the process of diagnosing malaria from blood sample images, reducing the need for manual examination and potentially increasing the speed and accuracy of diagnosis.

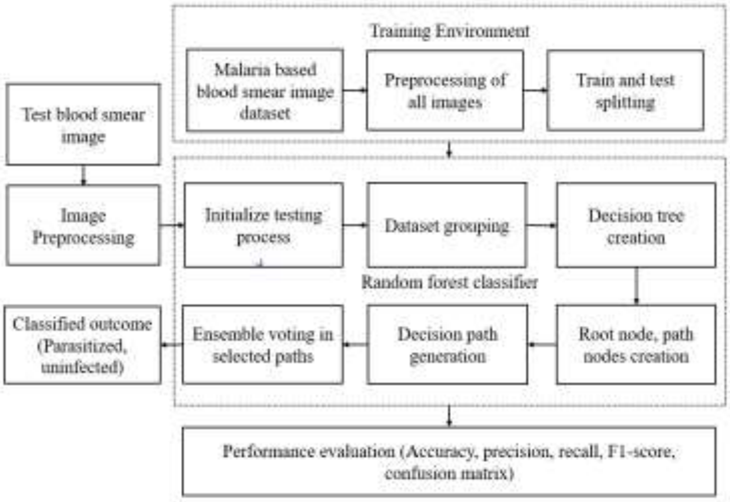


Figure 1: Proposed methodology

### 3.2 Image preprocessing

Image preprocessing is a critical step in computer vision and image analysis tasks. It involves a series of operations to prepare raw images for further processing by algorithms or neural networks. Here's an explanation of each step-in image preprocessing:

**Step 0. Image Read:** The first step in image preprocessing is reading the raw image from a source, typically a file on disk. Images can be in various formats, such as JPEG, PNG, BMP, or others. Image reading is performed using libraries or functions specific to the chosen programming environment or framework. The result of this step is a digital representation of the image that can be manipulated programmatically.

**1. Image Resize:** Image resize is a common preprocessing step, especially when working with machine learning models or deep neural networks. It involves changing the dimensions (width and height) of the image. Resizing can be necessary for several reasons:

- Ensuring uniform input size: Many machine learning models, especially convolutional neural networks (CNNs), require input images to have the same dimensions. Resizing allows you to standardize input sizes.
- Reducing computational complexity: Smaller images require fewer computations, which can be beneficial for faster training and inference.
- Managing memory constraints: In some cases, images need to be resized to fit within available memory constraints.

When resizing, it's essential to maintain the aspect ratio to prevent image distortion. Typically, libraries like OpenCV or Pillow provide convenient functions for resizing images.

**2. Image to Array:** In this step, the image is converted into a numerical representation in the form of a multidimensional array or tensor. Each pixel in the image corresponds to a value in the array. The array is usually structured with dimensions representing height, width, and color channels (if applicable).

For grayscale images, the array is 2D, with each element representing the intensity of a pixel. For color images, it's a 3D or 4D array, with dimensions for height, width, color channels (e.g., Red, Green, Blue), and potentially batch size (if processing multiple images simultaneously).

The conversion from an image to an array allows for numerical manipulation and analysis, making it compatible with various data processing libraries and deep learning frameworks like NumPy or TensorFlow.

**3. Image to Float32:** Most machine learning and computer vision algorithms expect input data to be in a specific data type, often 32-bit floating-point numbers (float32). Converting the image array to float32 ensures that the pixel values can represent a wide range of intensities between 0.0 (black) and 1.0 (white) or sometimes between -1.0 and 1.0, depending on the specific normalization used.

This step is essential for maintaining consistency in data types and enabling compatibility with various machine learning frameworks and libraries. It's typically performed by dividing the pixel values by the maximum intensity value (e.g., 255 for an 8-bit image) to scale them to the [0.0, 1.0] range.

**4. Image to Binary:** Image binarization is a process of converting a grayscale image into a binary image, where each pixel is represented by either 0 (black) or 1 (white) based on a specified threshold. Binarization is commonly used for tasks like image segmentation, where you want to separate objects from the background.

The process involves setting a threshold value, and then for each pixel in the grayscale image, if the pixel value is greater than or equal to the threshold, it is set to 1; otherwise, it is set to 0.

Binarization simplifies the image and reduces it to essential information, which can be particularly useful in applications like character recognition or object tracking, where you need to isolate regions of interest.

### 3.3 Dataset Splitting

In machine learning data pre-processing, we divide our dataset into a training set and test set. This is one of the crucial steps of data pre-processing as by doing this, we can enhance the performance of our machine learning model. Suppose if we have given training to our machine learning model by a dataset and we test it by a completely different dataset. Then, it will create difficulties for our model to understand the correlations between the models. If we train our model very well and its training accuracy is also very high, but we provide a new dataset to it, then it will decrease the performance. So we always try to make a machine learning model which performs well with the training set and also with the test dataset. Here, we can define these datasets as:

**Training Set**: A subset of dataset to train the machine learning model, and we already know the output.

**Test set**: A subset of dataset to test the machine learning model, and by using the test set, model predicts the output.

### 3.4 Random Forest Algorithm

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model. As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output. The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

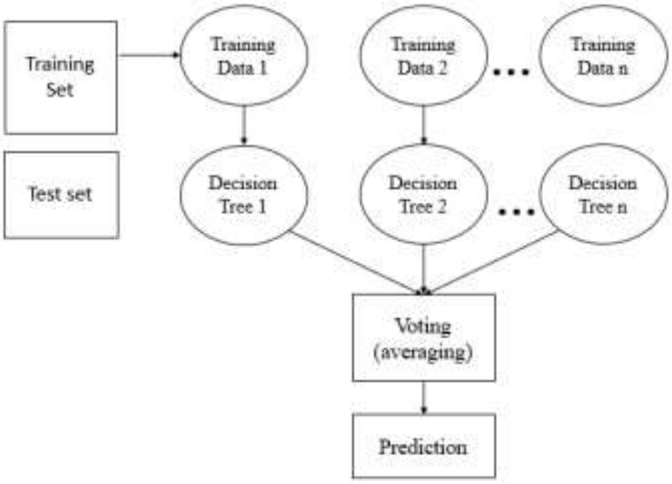


Figure 2: Random Forest algorithm.

### 3.4.1 Random Forest algorithm

Step 1: In Random Forest n number of random records are taken from the data set having k number of records.

Step 2: Individual decision trees are constructed for each sample.

Step 3: Each decision tree will generate an output.

Step 4: Final output is considered based on Majority Voting or Averaging for Classification and regression respectively.

### 3.4.2 Important Features of Random Forest

- **Diversity-** Not all attributes/variables/features are considered while making an individual tree, each tree is different.
- **Immune to the curse of dimensionality-** Since each tree does not consider all the features, the feature space is reduced.
- **Parallelization-**Each tree is created independently out of different data and attributes. This means that we can make full use of the CPU to build random forests.
- **Train-Test split-** In a random forest we don’t have to segregate the data for train and test as there will always be 30% of the data which is not seen by the decision tree.
- **Stability-** Stability arises because the result is based on majority voting/ averaging.

### 3.4.3 Assumptions for Random Forest

Since the random forest combines multiple trees to predict the class of the dataset, it is possible that some decision trees may predict the correct output, while others may not. But together, all the trees predict the correct output. Therefore, below are two assumptions for a better Random Forest classifier:

- There should be some actual values in the feature variable of the dataset so that the classifier can predict accurate results rather than a guessed result.
- The predictions from each tree must have very low correlations.

Below are some points that explain why we should use the Random Forest algorithm

- It takes less training time as compared to other algorithms.
- It predicts output with high accuracy, even for the large dataset it runs efficiently.
- It can also maintain accuracy when a large proportion of data is missing.

### 3.4.4 Types of Ensembles

Before understanding the working of the random forest, we must look into the ensemble technique. Ensemble simply means combining multiple models. Thus, a collection of models is used to make predictions rather than an individual model. Ensemble uses two types of methods:

**Bagging**– It creates a different training subset from sample training data with replacement & the final output is based on majority voting. For example, Random Forest. Bagging, also known as Bootstrap Aggregation is the ensemble technique used by random forest. Bagging chooses a random sample from the data set. Hence each model is generated from the samples (Bootstrap Samples) provided by the Original Data with replacement known as row sampling. This step of row sampling with replacement is called bootstrap. Now each model is trained independently which generates results. The final output is based on majority voting after combining the results of all models. This step which involves combining all the results and generating output based on majority voting is known as aggregation.

**Boosting**– It combines weak learners into strong learners by creating sequential models such that the final model has the highest accuracy. For example, ADA BOOST, XG BOOST.

## 4. RESULTS AND DISCUSSION

### 4.1 Dataset

Table 1 provides the dataset description. The dataset contains total of 1047 images with 530 images in uninfected class and 517 images in parasitized class.

Table 1: Dataset description.

| S. No. | Number of images | Class type |
|---|---|---|
| 1 | 530 | Uninfected |
| 2 | 517 | Parasitized |

### 4.2 Results analysis

Figure 3 shows a selection of images from the dataset that are classified as belonging to the "parasitized" class. These images likely exhibit characteristics associated with parasitized in blood smear samples

Figure 4 displays sample images from the dataset categorized as "uninfected." These images are likely examples of blood smear samples with no signs of parasitized or abnormalities.

Figure 5 represents the numerical data of the input images after they've undergone preprocessing, which involve tasks such as resizing, normalization, and flattening to prepare the images for input to the machine learning model.

In Figure 6, the target array data is depicted. Each value in this array corresponds to the classification of the corresponding input image as "uninfected" (0) or "parasitized" (1).

Figure 7 demonstrates the results of making predictions using the proposed machine learning (ML) model on a set of test data. It shows a few test images, and the predicted class labels.

Figure 8 contains the classification report generated for the random forest model, which provides the quality metrics such as precision, recall, and F1-score for each class, allowing us to assess the model's performance on different metrics

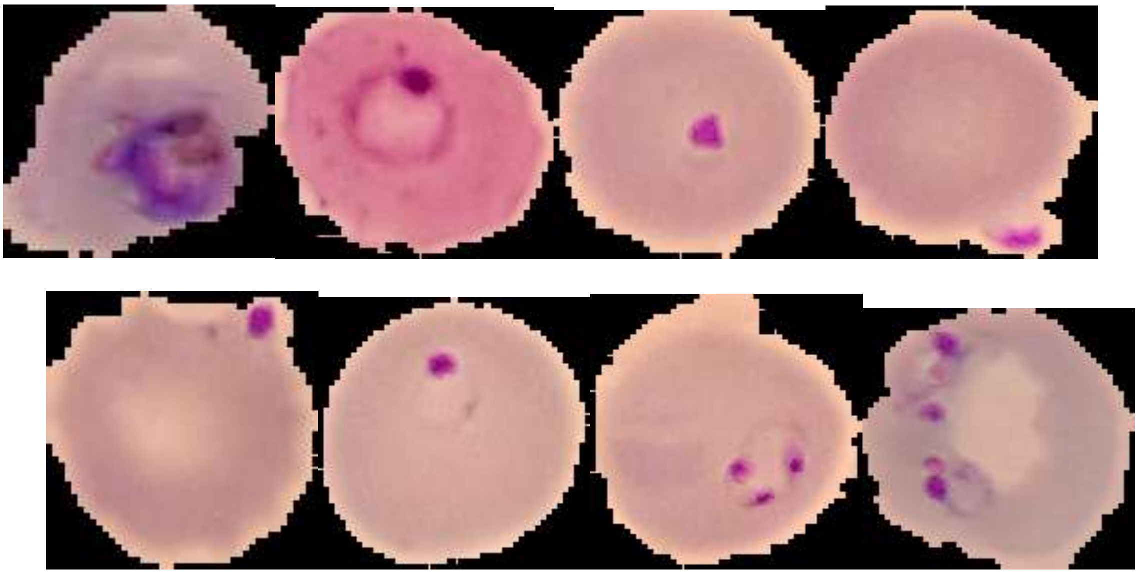


Figure 3: Sample images of dataset with parasitized class.

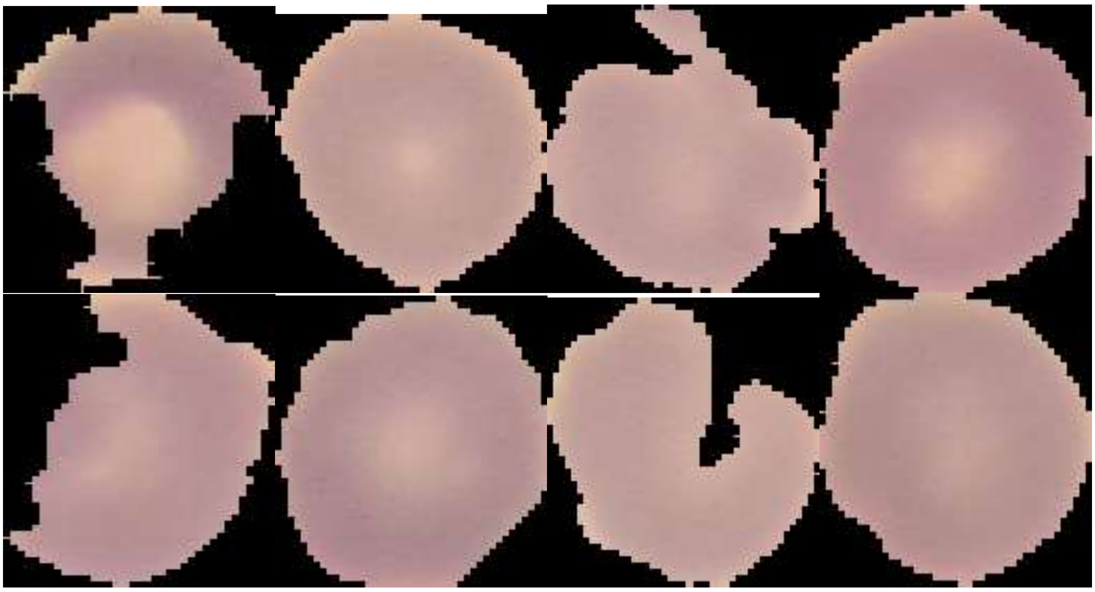
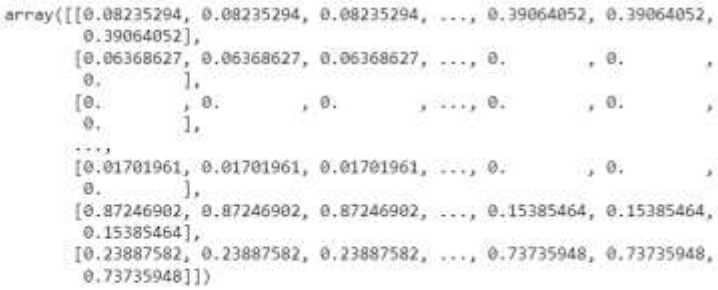
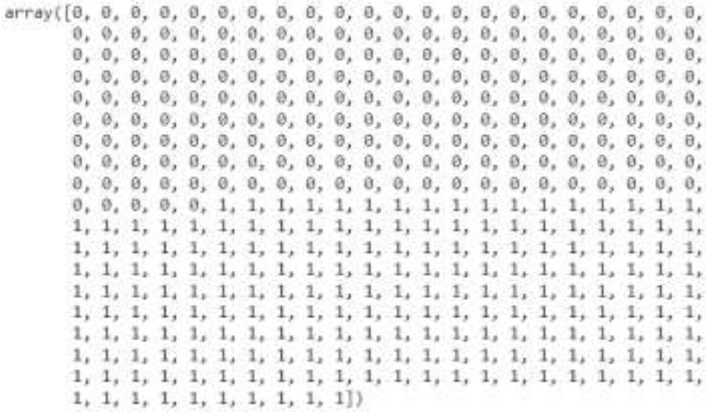


Figure 4: Sample images from dataset with uninfected class.



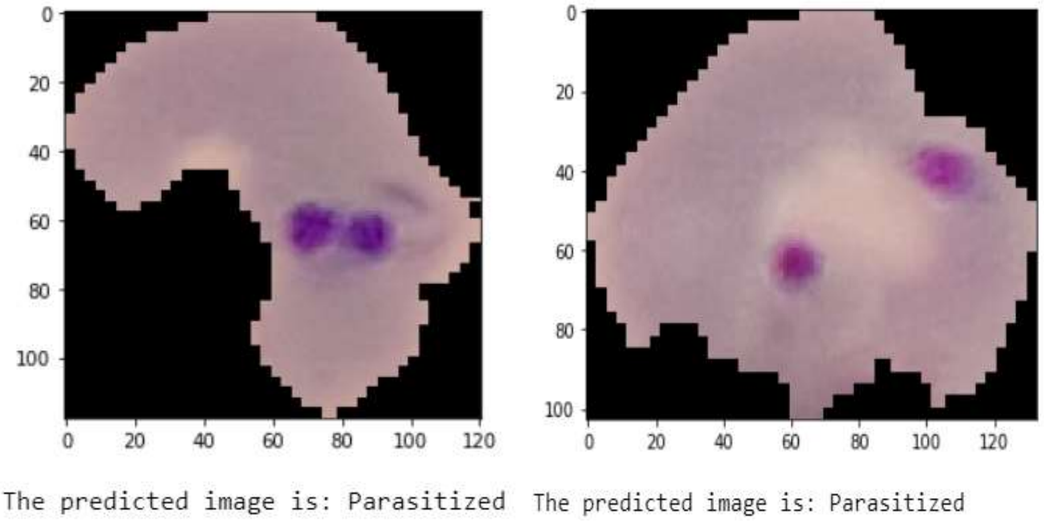
```
array([[0.08235294, 0.08235294, 0.08235294, ..., 0.39064052, 0.39064052,
        0.39064052],
       [0.06368627, 0.06368627, 0.06368627, ..., 0.        , 0.        ,
        0.        ],
       [0.        , 0.        , 0.        , ..., 0.        , 0.        ,
        0.        ],
       ...,
       [0.01701961, 0.01701961, 0.01701961, ..., 0.        , 0.        ,
        0.        ],
       [0.87246902, 0.87246902, 0.87246902, ..., 0.15385464, 0.15385464,
        0.15385464],
       [0.23887582, 0.23887582, 0.23887582, ..., 0.73735948, 0.73735948,
        0.73735948]])
```

Figure 5: Array data of input images after preprocessing.



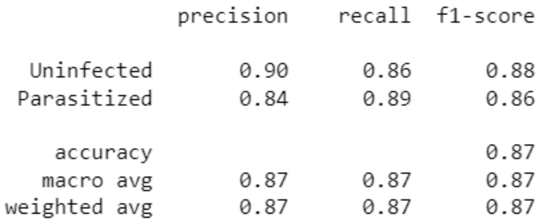
```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1])
```

Figure 6: Target array data (uninfected = 0, and parasitized = 1).



```
The predicted image is: Parasitized  The predicted image is: Parasitized
```

Figure 7: Sample prediction on test data using proposed ML model.



```
              precision    recall  f1-score

  Uninfected       0.90      0.86      0.88
 Parasitized       0.84      0.89      0.86

    accuracy                           0.87
   macro avg       0.87      0.87      0.87
weighted avg       0.87      0.87      0.87
```

Figure 8: Classification report of random forest model.

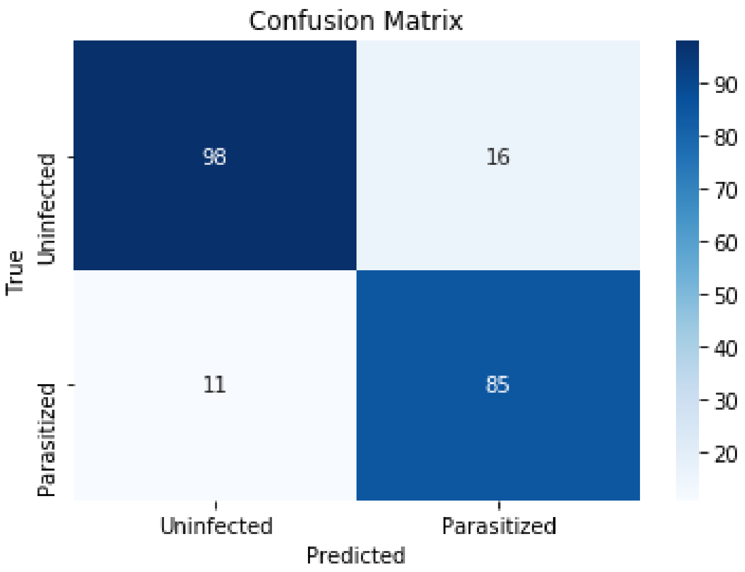
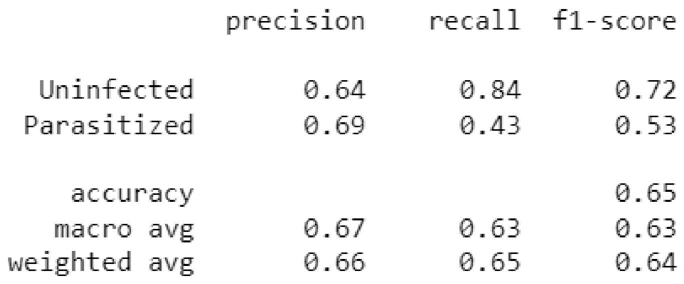


Figure 9: Obtained confusion matrix with actual and predicted labels using random forest model.

In Figure 9, a confusion matrix visualizes the performance of a classification model. This presents a heatmap-style confusion matrix showing the relationship between actual labels and predicted labels from the random forest model.



```
              precision    recall  f1-score

  Uninfected       0.64      0.84      0.72
 Parasitized       0.69      0.43      0.53

    accuracy                           0.65
   macro avg       0.67      0.63      0.63
weighted avg       0.66      0.65      0.64
```

Figure 10: Classification report of proposed KNN model.

Figure 10 displays the classification report for the proposed KNN model, which has improved performance over random forest model. Figure 9 shows the confusion matrix for the KNN model. It illustrates how well the KNN model correctly classified images into different classes (uninfected or parasitized).

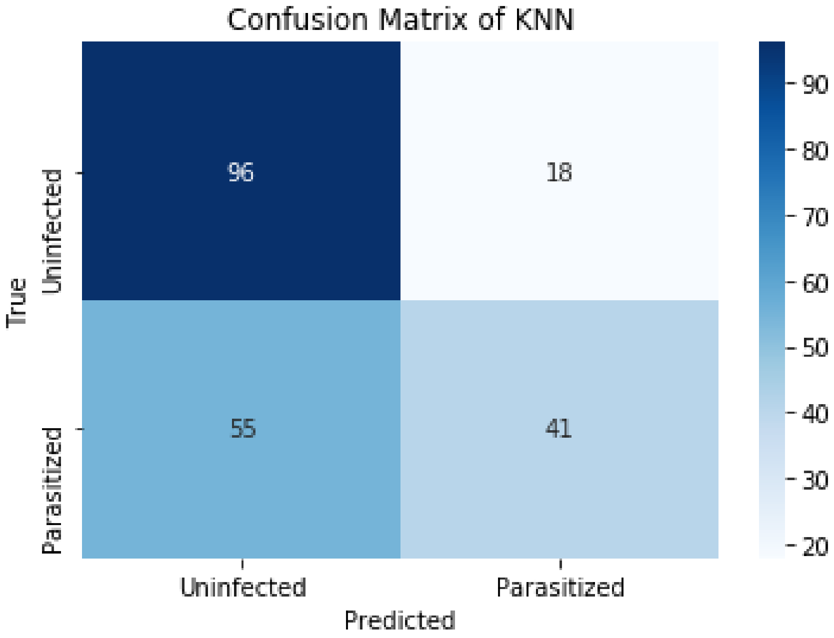


Figure 11: Confusion matrix of proposed KNN model for detection and classification of CXR images.

Table 2: Overall performance comparison of proposed ML models.

| Model name | Accuracy (%) | Precision (%) | Recall (%) | F1-score |
|---|---|---|---|---|
| Random Forest | 87.14 | 87 | 87 | 87 |
| KNN classifier | 65.23 | 66 | 65 | 64 |

Table 2 presents a comprehensive comparison of the overall performance of two proposed ML models used for parasitized detection and classification in blood smear samples. The models under consideration are the "Random Forest" model and the "KNN classifier." The evaluation is based on key performance metrics, including "Accuracy," "Precision," "Recall," and "F1-score." Accuracy, which signifies the ratio of correctly predicted labels to the total predictions, is an essential metric. The "Random Forest" model achieves an accuracy of 87.14%, while the "KNN classifier" demonstrates very less accuracy of 65.23%. Precision, indicating the correctness of predicted positive instances, is 87% for the "Random Forest" model and 66% for the "KNN classifier." Recall, also known as sensitivity, reveals the capability to correctly identify actual positive instances. Both models showcase comparable recall values, with the "Random Forest" and "KNN classifier" achieving 87% and 65% respectively. F1-score, a balance between precision and recall, harmonizes the trade-offs between false positives and false negatives. Notably, the F1-scores mirror the accuracy and precision values for both models, with 87% for "Random Forest" and 64% for the "KNN classifier."

## 5. CONCLUSION

In conclusion, the methodology involving image processing followed by Random Forest Classifier (RFC) building and prediction for malaria diagnosis from blood sample images represents a significant advancement in the field of healthcare and disease management. This approach addresses critical challenges related to the efficiency, accuracy, and accessibility of malaria diagnosis. By automating the analysis of blood sample images, it streamlines the diagnostic process, reducing the time required for diagnosis and treatment initiation. Additionally, it enhances diagnostic consistency, reduces the potential for human error, and offers scalability, making it suitable for both routine diagnostics and large-scale screening efforts. The integration of machine learning and image analysis technologies into healthcare systems holds promise for improving malaria control, early detection of outbreaks, and enhancing overall healthcare access. While there may be initial development costs, the long-term benefits in terms of improved healthcare delivery, reduced costs, and better disease surveillance make this methodology a valuable addition to the fight against malaria.

## REFERENCES

[1]. WHO? World Malaria Report 2022. Available online: https://www.who.int/teams/global-malaria-programme/reports/world-malaria-report-2022 (accessed on 1 March 2023).

[2]. WHO? World Malaria Report 2021: An In-Depth Update on Global and Regional Malaria Data and Trends. Available online: https://www.who.int/teams/global-malaria-programme/reports/world-malaria-report-2021 (accessed on 1 September 2022).

[3]. Yang, F.; Poostchi, M.; Yu, H.; Zhou, Z.; Silamut, K.; Yu, J.; Maude, R.J.; Jaeger, S.; Antani, S. Deep learning for smartphone-based malaria parasite detection in thick blood smears. IEEE J. Biomed. Health Inform. 2019, 24, 1427–1438.

[4]. World Health Organization. Malaria Microscopy Quality Assurance Manual, 2nd ed.; World Health Organization: Geneva, Switzerland, 2016; Available online: https://www.who.int/docs/default-source/documents/publications/gmp/malaria-microscopy-quality-assurance-manual.pdf (accessed on 2 March 2023).

[5]. Rawat, W.; Wang, Z. Deep convolutional neural networks for image classification: A comprehensive review. Neural Comput. 2017, 29, 2352–2449.

[6]. Colubri, A.; Silver, T.; Fradet, T.; Retzepi, K.; Fry, B.; Sabeti, P. Transforming clinical data into actionable prognosis models: Machine-learning framework and field-deployable app to predict outcome of Ebola patients. PLoS Negl. Trop. Dis. 2016, 10, e0004549. [Green Version]

[7]. Smith, K.P.; Kirby, J.E. Image analysis and artificial intelligence in infectious disease diagnostics. Clin. Microbiol. Infect. 2020, 26, 1318–1323.

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

[9]. Bibin, D.; Nair, M.S.; Punitha, P. Malaria parasite detection from peripheral blood smear images using deep belief networks. IEEE Access 2017, 5, 9099–9108.

[10]. Gopakumar, G.P.; Swetha, M.; Sai Siva, G.; Sai Subrahmanyam, G.R.K. Convolutional neural network-based malaria diagnosis from focus stack of blood smear images acquired using custom-built slide scanner. J. Biophotonics 2018, 11, e201700003.

[11]. Dandekar, R.; Rackauckas, C.; Barbastathis, G. A machine learning-aided global diagnostic and comparative tool to assess effect of quarantine control in COVID-19 spread. Patterns 2020, 1, 100145.

[12]. Baldominos, A.; Puello, A.; Oğul, H.; Aşuroğlu, T.; Colomo-Palacios, R. Predicting infections using computational intelligence–a systematic review. IEEE Access 2020, 8, 31083–31102.

[13]. O'Shea, K.; Nash, R. An introduction to convolutional neural networks. arXiv 2015, arXiv:1511.08458.

[14]. Sadeghi-Tehran, P.; Angelov, P.; Virlet, N.; Hawkesford, M.J. Scalable database indexing and fast image retrieval based on deep learning and hierarchically nested structure applied to remote sensing and plant biology. J. Imaging 2019, 5, 33. [Green Version]

[15]. Wu, J. Introduction to Convolutional Neural Networks; National Key Lab for Novel Software Technology, Nanjing University: Nanjing, China, 2017; Volume 5, p. 495. [Google Scholar]

[16]. Sen, O.; Keles, H.Y. A Hierarchical Approach to Remote Sensing Scene Classification. PFG–J. Photogramm. Remote Sens. Geoinf. Sci. 2020, 90, 161–175.

[17]. Alzubaidi, L.; Zhang, J.; Humaidi, A.J.; Al-Dujaili, A.; Duan, Y.; Al-Shamma, O.; Santamaría, J.; Fadhel, M.A.; Al-Amidie, M.; Farhan, L. Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. J. Big Data 2021, 8, 53.