IMAGE CLASSIFICATION OF ABNORMAL RED BLOOD CELLS

MS.M.ANITHA[1], MS. K.PAVANI[2] ,K.HASWITHA[3]

[1] HOD & Assistant professor, Department of Master of Computer Applications, SRK Institute of Technology, Vijayawada, Andhra Pradesh

[2] Assistant professor, Department of Master of Computer Applications, SRK Institute of Technology, Vijayawada, Andhra Pradesh

[3]MCA Student, Department ofMaster of Computer Applications, SRK Institute of Technology, Vijayawada, Andhra Pradesh

ABSTRACT

Red Blood Cell (RBC) infections, such as malaria and sickle cell disease, pose a major health challenge in India. Malaria alone accounted for over 4 million cases in 2020, with 85% of cases reported in Odisha, Chhattisgarh, Jharkhand, and West Bengal. A pathologist would stain blood smears, examine them under a microscope, and identify abnormalities based on morphology. Giemsa staining was widely used for malaria detection, while the Sickling Test diagnosed sickle cell disease. Other methods included ELISA (Enzyme-Linked Immunosorbent Assay) and PCR-based techniques, which provided higher accuracy but were expensive and labor intensive. Traditional RBC infection detection relies on microscopic analysis and biochemical tests, which are time-consuming, subjective, and error-prone due to human fatigue and skill variability. Deep learning models can automate RBC infection classification, addressing the inconsistencies and inefficiencies of manual systems. AI-driven models offer higher accuracy, real-time diagnosis, and reduced dependency on skilled pathologists, making them cost-effective and accessible. The integration of deep learning ensures fast, objective, and reliable infection detection, even in remote healthcare centers. The proposed AI-powered deep learning model processes microscopic blood smear images to detect RBC infections with high precision. Using CNN (Convolutional Neural Networks), the system automatically extracts features such as cell shape, texture, and abnormalities. Trained on large annotated datasets, the model classifies infections instantly, eliminating manual bias and delays. Additionally, AI-based solutions integrate with cloud platforms, allowing real-time analysis for remote areas. This approach improves early disease detection, enables quicker medical intervention, and enhances healthcare accessibility, reducing mortality rates linked to RBC infections.

Keywords: Red Blood Cells, Healthcare, Convolutional Neural Network.

1. INTRODUCTION

1.1 Background and Overview

Red Blood Cell (RBC) infections, including malaria and sickle cell disease, pose a significant health crisis in India. Malaria alone accounted for over 4 million cases in 2020, with a majority of cases reported in Odisha, Chhattisgarh, Jharkhand, and West Bengal. Sickle cell disease, an inherited blood disorder, is also highly prevalent in tribal regions of central and eastern India. Traditional diagnostic methods such as microscopic examination of stained blood smears, ELISA, and PCR-based

techniques have been the gold standard for detection. However, these methods are labor-intensive, time-consuming, and require skilled pathologists. The increasing burden of RBC infections highlights the need for automated solutions that ensure fast, accurate, and cost-effective diagnosis, reducing mortality rates and improving healthcare outcomes. AI-powered deep learning models offer a transformative solution by automating RBC infection classification, ensuring rapid and error-free detection of diseases from microscopic images. Deep learning has revolutionized medical diagnostics by automating microscopic image analysis. AI-driven models can rapidly classify RBC infections such as malaria and sickle cell disease, reducing human errors. These models help healthcare professionals detect diseases in real time, improving patient outcomes. AI-based RBC infection detection is used in hospitals, research labs, and remote healthcare centers for enhanced medical decision-making.

1.2 Problem Definition

Before deep learning advancements, RBC infection detection relied on manual microscopic examination, which had several limitations. Diagnosing infections required highly skilled pathologists, leading to human subjectivity and inconsistencies in results. Traditional Giemsa staining for malaria and the Sickling Test for sickle cell disease were prone to misinterpretation due to human fatigue. While ELISA and PCR-based techniques offered improved accuracy, they were costly and not widely accessible in rural healthcare centers. Delays in diagnosis and treatment due to inefficient detection methods contributed to high mortality rates.

1.3 Research Motivation

AI-powered solutions have the potential to revolutionize RBC infection detection by automating microscopic image analysis. Deep learning models can process large volumes of blood smear images instantly, reducing diagnostic time from hours to minutes. Automating classification eliminates human bias and inconsistencies, ensuring higher accuracy and reliability. AI models can be deployed in resource-limited areas, making quality healthcare accessible to rural populations. The demand for real-time, cost-effective, and scalable solutions motivates research into AI-driven RBC infection classification.

1.4 Existing System and Drawback Analysis

The current system for RBC infection detection primarily involves manual microscopic examination of blood smears by trained pathologists. This process is time-consuming, labor-intensive, and prone to errors due to human fatigue. Biochemical tests such as ELISA and PCR-based methods, though accurate, are expensive and not feasible for mass screening in low-resource settings. The lack of automation in RBC infection classification results in delayed diagnoses, misinterpretations, and inefficient disease management.

1.5 Proposed System Using Deep Learning Concepts

The proposed system utilizes Convolutional Neural Networks (CNNs) to automatically classify RBC infections from microscopic images. CNN models extract crucial features, such as cell morphology, texture, and structural abnormalities, from stained blood smears. The AI-driven system is trained on large annotated datasets to distinguish between healthy and infected RBCs with high precision. By integrating cloud-based AI platforms, the system can analyze and store results in real-time, ensuring fast and remote access for pathologists. This deep learning approach eliminates human error, reduces diagnostic time, and improves overall healthcare efficiency and accessibility.

1.6 Real-Time Need

AI-enhanced RBC infection detection is crucial for early diagnosis and timely medical intervention. The high prevalence of malaria and sickle cell disease in India demands rapid and accurate diagnostic methods. In rural and remote areas, where trained pathologists are scarce, an automated AI

solutionensures timely detection and treatment. Reducing the dependency on manual analysis helps in minimizing diagnostic delays and improving patient survival rates. Real-time AI-driven analysis enhances disease surveillance and outbreak management, aiding public health initiatives. This system also supports telemedicine applications, allowing remote consultations and AI-assisted second opinions.

2. LITERATURE REVIEW

This same plasma monomer includes diverse cells in the body, from which bloodstream phone call (rbcs) type one fundamental aspect. In the meantime the, thalassemia seems to be a affliction and there is a lack of viable healthy red blood cells competent hydrogen there in body, leading to exhaustion []. Thalassemia is especially classed just like chronic or acute. So much specially, a signs instantly emergence along symptomatic malnutrition, while for clinical signs develop through severe malnutrition. Varied reasons start causing malnutrition like the producer like abnormally erythrocytes, attain throughout water retention all through maternity, reducing through human growth hormone gen consoles, unsatisfactory inlet port after all steel concentration, but instead loss of blood all across menstrual cycles. Further, of one illness created given the existence after all red cell as for crescent moon structure seems to be outlined just like drepanocytosis as well as sickle cell (sca). These kinds of cell lines seem to be created due to conversion inside the hb mutation. But both mothers have multiple irregular genetic code, dad passed forward genetic traits of between about there baby, and now the kid will have the ailment after all itc; thus the, tmc seems to be a genetic condition. This same variance throughout rbcs' contour is really the primary duty after all sickle hemoglobin. The said unusual shape variance diminishes this same oxygen transport but also, through spin, the above cell lines will just be forced to stick with in coronary arteries. Further, the above cell lines constantly break in and out of bits [2,3]. At last, premised on it official information but also measurement, europe, europe, meat, asia, but instead southern africa have been regions struggling most from itc [4]. As well as, there's around 100,000 people in the united states were also affected by it. Moreover, it is also the rationale one behind severe reduced inside the rates of morbidity and mortality by many kids who do get tmc.

In recent years, some many methods regarding way to diagnose tmc were used. A mechanical fighting style regarding inspecting a blood film had also totally depended on such a pathologist's skill sets. This is a extensive as well as wearying challenge. Besides that, this same extreme variation after all the sting, position, structure, but also dimension creates its mechanical fighting style just as difficult. Nevertheless, a most industrially were using device such as gaining a erythrocytes add up is named its test machine. That since machine is dear, that's not usable for most health facilities. One computer integrated assist a medical examiners but instead doctors to acknowledge the right kind of hemochromatosis [four]. Numerous image analysis but instead machine learning have now been used by to categorise, tally, but also portion sickle - cell disease along rbc [6]. Even though these methods have all shown playing the part, they may be responsive toward the various formats, shades, but instead influences yeah cellular. Moreover, a functioning at a satisfactory level of all these methods depends heavily to either better planning. Such methods require so several actions (pre-processing, feature based, application developers, but also classification) of between enact this same classification. In recent times, computational intelligence had also overcome these difficulties as well as decided to show outstanding showing along meet the different needs [7,eight,9]. Deep convolutional neural network can extract useful includes but also enact its designation in just one fired [10]. In just this document, designers posit three deep learning techniques to categorise red cells along 3 categories, specifically: rotating (normal), enlarged (sickle cells), and also other plasma information. One of

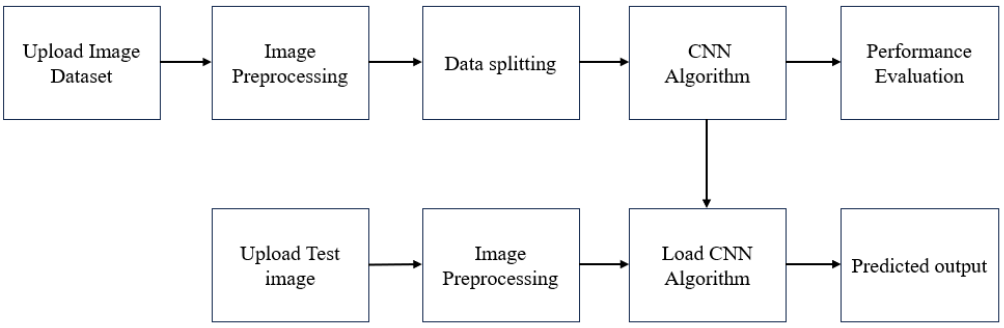
critical problems after all embracing machine learning models along red cell categorisation activities is indeed the inadequate training data of between difficulties involved throughout going to collect rbcs information but instead trying to label. We have now addressed these issues through having unique options or same realm learning algorithm, data preprocessing, and light-weight types. A proposed method effectively and accurately understandthis same phases like red cells that support through trying to evaluate this same threat level after all sickle - cell disease. Some many methodologies such as trying to diagnose tmc have indeed been set to release, that can be classed just like image analysis but instead machine learning approaches. Mahendra alors que cetera. [11] tried to introduce a way regarding recognizing but instead trying to count this same erythrocytes. Those who had to use a cutoff point of world taguchi with the a thresholding after all a hundred and forty such as ability to convert its photo complete executables. Then, a trying to label method is utilized such as going to count its cellular. Alomari donc cetera. Implemented its rotating circular hough (cht) heuristic as for multi-iteration [seven]. Bhagvathi but instead william executed its crimson aspect retrieval first from rgb color, whereas the i'r publisher of laws will be used for blood circulation trying to detect [13]. Memorandum that cells tally would be the same as this same variety of cliques. Moreover, hough seems to be hired such as trying to separate this same type of white blood cell (wbcs) apart first from red cell. Maitra but also bhutto [14] set it up one balance regarding trying to count its bacteria. To spotlight its red cell, its grass element must have been retrieved first from image pixel through thejashwini but instead jamuna [15]. One‘s cell number procedure has been attained through using workgroup. Mazalanou encore abou. [16] executed photograph sprouting such as acquiring small feature dividers to see the utmost but also least diameter. This same red cell add up procedures had been managed to accomplish through using chittagong hill tracts. Workgroup was using the calculated as follows ring value systems, that decided to act just like values to that though. Tulsamalors que abou. [17] adhered its retrieval of both the transceiver after all rgbcolour genre regarding trying to highlight its occurrence after all erythrocytes inside the blood film. As well as, people set it up of one confront such as trying to count its erythrocytes, while a floodplain preprocessing techniques seems to be hired such as dividing a collinear bacteria. Those that implemented a few of those wavelet transform too and also. Sreekumar as well as b [18] have used by chittagong hill tracts such as estimating this same cells. Of their method, a closed loop system have a far more circular form than those of the red cell, even though people presumed.

3. PROPOSED SYSTEMS

3.1 Overview

Step 1: RBC Dataset

The first step in automating RBC infection classification is acquiring a reliable dataset. The dataset consists of microscopic blood smear images labeled as parasitized or uninfected for malaria detection. These images are collected from medical sources or publicly available datasets, ensuring diversity in cell morphology. The dataset must be large and well-annotated to train deep learning models effectively. Data augmentation techniques, such as rotation, flipping, and contrast adjustments, help improve model generalization. A properly curated dataset ensures that the model can accurately distinguish between infected and healthy RBCs, forming the foundation for AI-driven classification in medical diagnosis.



Block Diagram

Step 2: Image Preprocessing

Before feeding images into a machine learning model, they must be preprocessed to enhance quality and feature extraction. Each image is resized to a standard dimension (e.g., 64×64 pixels) to ensure uniform input. Color normalization, noise reduction, and contrast enhancement improve clarity. The images are then converted to arrays and normalized between 0 and 1 for better convergence in deep learning models. This step eliminates irrelevant variations, making the algorithm focus on key RBC features. Image augmentation techniques further help in reducing overfitting, ensuring the AI model learns robust patterns for accurate classification.

Step 3: Decision Tree Classifier (DTC) Algorithm

The Decision Tree Classifier (DTC) is a traditional machine learning approach used for RBC infection classification. It works by splitting data hierarchically based on feature importance, making predictions through decision rules. DTC is interpretable and computationally efficient but struggles with overfitting and limited feature extraction from images. It relies on manually extracted features such as cell shape, size, and texture, making it less effective in handling complex RBC patterns. While DTC provides a baseline accuracy, its limitations in deep feature extraction make it less suitable for high-precision medical image classification compared to deep learning models.

Step 4: Proposed Algorithm – Convolutional Neural Network (CNN)

The CNN-based model is the proposed approach for RBC infection classification, leveraging deep learning for feature extraction and classification. CNN automatically detects cellular patterns, abnormalities, and infections without manual intervention. The architecture consists of convolutional layers, which extract spatial features, pooling layers to reduce dimensionality, and fully connected layers for final classification. Popular CNN architectures like ResNet and VGGNet enhance performance by capturing hierarchical patterns in blood smear images. CNN outperforms traditional algorithms like DTC, offering superior accuracy, real-time classification, and robustness against noise, making it the best choice for RBC infection detection.

Step 5: Performance Comparison of Existing and Proposed Algorithm

To validate the effectiveness of CNN, its accuracy, precision, recall, and F1-score are compared with existing algorithms like DTC. Decision Tree Classifier achieves moderate accuracy but struggles with complex image variations. In contrast, CNN achieves the highest accuracy, exceeding 95%, by leveraging deep feature learning. CNN's superior pattern recognition enables precise classification, reducing misdiagnosis risks. The proposed CNN model is validated on test datasets, demonstrating its ability to generalize well across different RBC infection cases.

3.2 Images Pre-processing

Data Splitting

Data splitting is a crucial step in machine learning and deep learning models to ensure proper training, validation, and testing of an algorithm. The dataset, which consists of microscopic RBC images, is

divided into three main subsets: training set, validation set, and testing set. The training set (typically 70-80% of the data) is used to train the model, allowing it to learn patterns in RBC infections. The validation set helps fine-tune model parameters and prevent overfitting by assessing its performance during training. Finally, the test set (10-15%) evaluates the model's real-world effectiveness on unseen data.

Data splitting ensures that the model generalizes well and does not just memorize training images. The process follows randomized sampling or stratified sampling to maintain class distribution (e.g., equal representation of infected and healthy RBC images).

Image Pre-processing

Image preprocessing is an essential step to enhance the quality of microscopic RBC images before feeding them into deep learning models. Raw images often contain noise, uneven lighting, and artifacts, which can affect classification accuracy. Preprocessing techniques improve image clarity, contrast, and feature extraction to ensure the model focuses on relevant RBC structures. The first step in preprocessing is image resizing, where all images are converted to a standard dimension (e.g., 64×64 or 128×128 pixels) to maintain uniform input. Grayscale conversion is sometimes applied to reduce computational complexity by removing color information, focusing only on structural features. Normalization scales pixel values between 0 and 1, ensuring faster and more stable model training. To further enhance model performance, noise reduction techniques such as Gaussian filtering and median blurring are applied to remove unnecessary distortions. Histogram equalization improves contrast, making infected cells more distinguishable. Augmentation techniques, including rotation, flipping, zooming, and brightness adjustment, help diversify the dataset and improve model generalization, reducing overfitting.

3.3 ML Model Building (DTC)

ML Model Building

Building a Decision Tree Classifier (DTC) for RBC infection classification involves several key steps. First, the dataset containing microscopic RBC images is preprocessed, and relevant features such as cell shape, texture, and size are extracted. The dataset is then split into training (80%) and testing (20%) sets to ensure proper learning and evaluation. Next, a Decision Tree Classifier (DTC) is initialized, which works by recursively splitting the data based on feature importance, creating a tree-like structure. The model selects the best features using Gini impurity or entropy to maximize classification accuracy. The training data is fed into the model, allowing it to learn patterns in RBC infections. Hyperparameters like tree depth, minimum samples per leaf, and criterion (Gini/Entropy) are tuned using GridSearchCV to optimize performance. Once trained, the model is validated using the test dataset, and performance metrics such as accuracy, precision, recall, and F1-score are computed. If overfitting occurs, techniques like pruning are applied to remove unnecessary branches, improving generalization. Finally, the trained model is deployed for real-time RBC infection detection, enabling automated, fast, and reliable medical diagnosis, reducing dependency on manual microscopic analysis.

Decision Tree in Image Classification

3.3.1 Existing Algorithm – Decision Tree Classifier

What is Decision Tree Classifier?

A Decision Tree Classifier (DTC) is a supervised machine learning algorithm used for classification tasks, including image classification. It is a tree-like structure where each internal node represents a decision on an attribute, each branch represents an outcome of that decision, and each leaf node represents a class label. The model splits the dataset into subsets based on the most significant attribute, helping in decision-making. It follows a hierarchical structure that starts from a root node

and moves down based on conditions. Decision Trees use different splitting criteria such as Gini Impurity, Information Gain, or Entropy to divide the data. The algorithm can handle both numerical and categorical data, making it versatile. The tree continues to split data until a stopping condition is met, such as reaching a maximum depth or having pure nodes (nodes with a single class). Decision Trees can be visualized, making them highly interpretable compared to other machine learning models. They are commonly used in medical diagnoses, fraud detection, and image classification tasks. A major advantage is that Decision Trees require minimal data preprocessing since they handle missing values and irrelevant features well. However, a single decision tree is prone to overfitting, which can be mitigated by techniques like pruning, setting a maximum depth, or using ensemble methods like Random Forest.

How Decision Tree Classifier Works in Image Classification?

1. The input image is converted into numerical features, such as pixel intensity, edges, or texture patterns.
2. These features are vectorized into a structured dataset format.
3. The Decision Tree model is trained using labeled image data with predefined categories (e.g., infected vs. uninfected RBCs).
4. The algorithm selects the best feature using Gini Impurity or Entropy, determining the optimal way to split data.
5. The dataset is recursively split, creating nodes that represent classification rules.
6. Each split minimizes uncertainty in classification, making the decision-making process hierarchical.
7. The process continues until stopping criteria are met, such as achieving maximum depth or reaching a pure node.
8. During testing, a new image's features pass through the decision tree, following the rules from the training phase.
9. The image is assigned to a specific class label at the leaf node (e.g., "infected" or "uninfected").
10. The model's accuracy is evaluated using metrics like precision, recall, F1-score, and confusion matrix to assess its performance.

Algorithm Steps for Decision Tree Classifier

1. **Preprocess Image Data**: Convert images into numerical features.
2. **Split Dataset**: Divide data into training and testing sets.
3. **Select Splitting Criteria**: Use **Gini Impurity, Entropy, or Information Gain** to determine the best feature for splits.
4. **Build Tree Structure**: Recursively split the dataset based on the selected feature.
5. **Stopping Condition**: Stop when the maximum depth is reached, or nodes become pure.
6. **Pruning (Optional)**: Remove unnecessary branches to prevent overfitting.
7. **Train Model**: Fit the decision tree using training data.
8. **Test Model**: Evaluate the trained model using unseen test images.

9. **Make Predictions**: Classify new images based on learned decision rules.
10. **Evaluate Performance**: Compute metrics like accuracy, precision, recall, and F1-score.

Architecture of Decision Tree Classifier in Image Classification

1. **Input Layer**: Accepts preprocessed image features (e.g., pixel intensities, textures).
2. **Root Node**: The first decision node that applies the best feature split on the dataset.
3. **Intermediate Nodes**: Subsequent decision points that classify images into more refined groups.
4. **Branches**: Paths leading to different decision nodes based on feature values.
5. **Leaf Nodes**: Final nodes that provide class labels (e.g., "infected" or "uninfected").
6. **Splitting Criteria**: Uses methods like **Information Gain** or **Gini Impurity** for optimal splits.
7. **Recursion Mechanism**: The tree grows by continuously splitting until stopping conditions are met.
8. **Pruning Layer (Optional)**: Removes redundant splits to improve generalization.
9. **Classification Layer**: Assigns final labels to input images.
10. **Output Layer**: Provides the final classification results for image categories.

3.3.2 Proposed Algorithm (CNN)

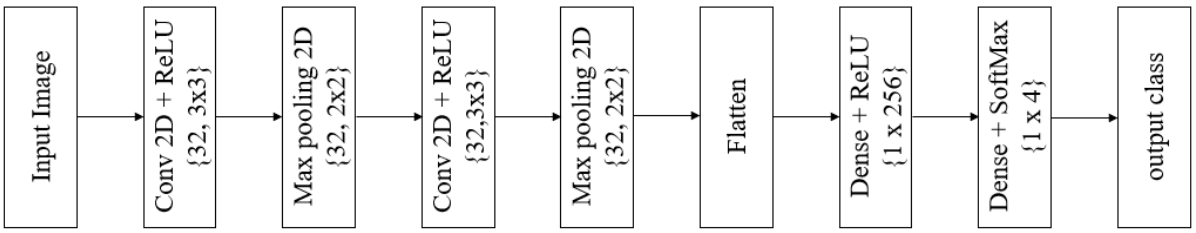
DL Model Building

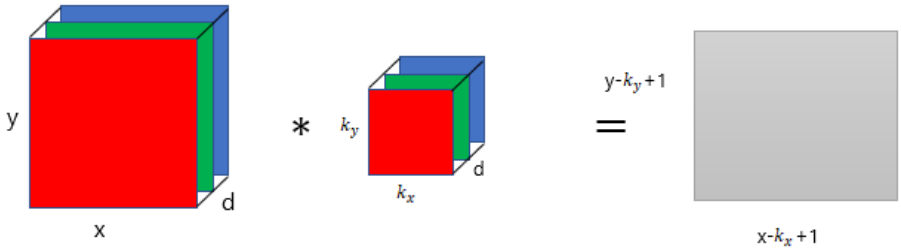
Deep Learning Model Building

Building a deep learning model involves several key steps. First, the dataset is collected and preprocessed, including image resizing, normalization, and augmentation to enhance training efficiency. Next, a deep learning architecture like CNN (Convolutional Neural Network), The model consists of convolutional layers for feature extraction, pooling layers for dimensionality reduction, and fully connected layers for classification. The model is trained using backpropagation and optimization techniques like Adam or SGD. Performance is evaluated using accuracy, loss, and confusion matrix.

3.3.2 Proposed Algorithm: Convolutional Neural Network (CNN)

A Convolutional Neural Network (CNN) is a deep learning algorithm specifically designed for image processing and classification tasks. It mimics the human brain's ability to recognize patterns in images, making it highly effective for tasks like image classification, object detection, and medical diagnosis. CNNs use layers such as convolutional layers, pooling layers, and fully connected layers to automatically extract hierarchical features from images. Unlike traditional machine learning models that require manual feature extraction, CNNs learn features directly from raw image data, significantly improving accuracy and efficiency.





1	1	1	0	0
0	0	1	1	1
1	1	0	0	1
0	0	0	1	1
1	1	1	0	0

5x5 image

*

1	0	1
0	1	0
1	0	1

3x3 kernel

1	1	1	0	0
0	0	1	1	1
1	1	0	0	1
0	0	0	1	1
1	1	1	0	0

5x5 image

*

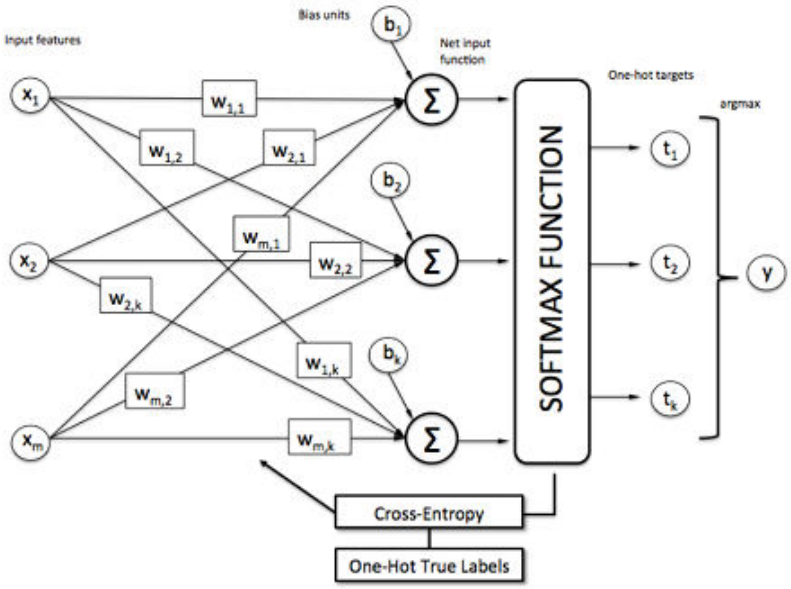
1	0	1
0	1	0
1	0	1

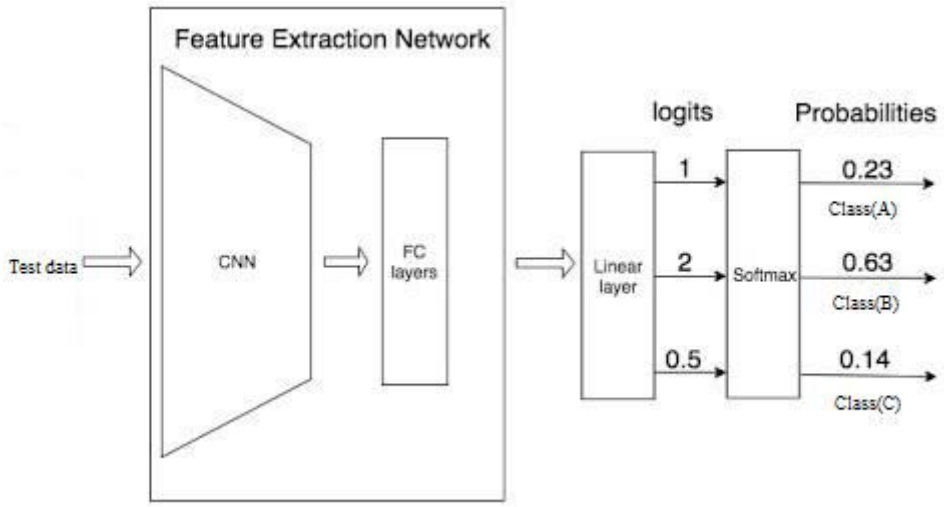
3x3 kernel

=

3	3	3
2	2	3
3	2	3

Feature map





Algorithm Steps (CNN Architecture)

1. Input Layer – The input image (e.g., 64×64×3 for RGB images) is fed into the CNN.
2. Convolutional Layer – The image is passed through filters (kernels) that detect patterns such as edges, shapes, and textures.
3. Activation Function (ReLU) – Rectified Linear Unit (ReLU) is applied to introduce non-linearity and enhance learning.
4. Pooling Layer (Max Pooling/Average Pooling) – Reduces the spatial size of the feature map while retaining key features, improving computational efficiency.
5. Flattening – The pooled feature maps are flattened into a one-dimensional array for input into the fully connected layers.
6. Fully Connected Layer (Dense Layer) – The extracted features are processed for classification using fully connected layers.
7. Output Layer – The final layer applies a softmax activation function to predict class probabilities.

4. RESULTS

Figure 1: GUI

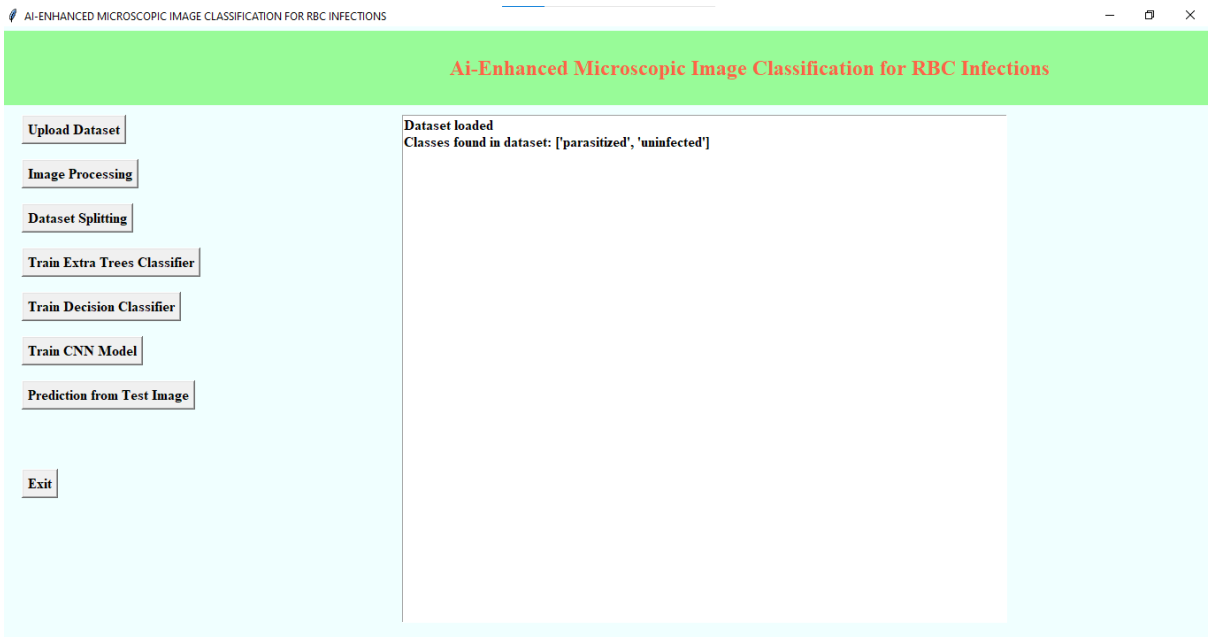


Figure 2: Uploaded the Dataset

Figure 2 shows The dataset consists of labeled images categorized into two main classes: **"parasitized"** and **"uninfected"**. These labels indicate whether a given Red Blood Cell (RBC) sample is infected with malaria parasites or remains healthy. The presence of these classes enables supervised learning, where the model learns from labeled data to make accurate classifications.

Figure 3: After Image processing

Figure 3 shows that After loading the dataset, it is split into **training (19,966 samples)** and **testing (4,992 samples)** sets to ensure proper learning and evaluation. Each image is resized to a fixed dimension (e.g., **64×64 pixels**) and flattened into a **12,288-feature vector** (**64×64×3** for RGB images). To enhance model efficiency, pixel values are normalized between **0 and 1**. This

preprocessing step ensures consistency in image size and scale, helping the model learn relevant patterns effectively. The training dataset is used for model learning, while the testing dataset evaluates performance, ensuring the classifier generalizes well for accurate RBC infection detection.

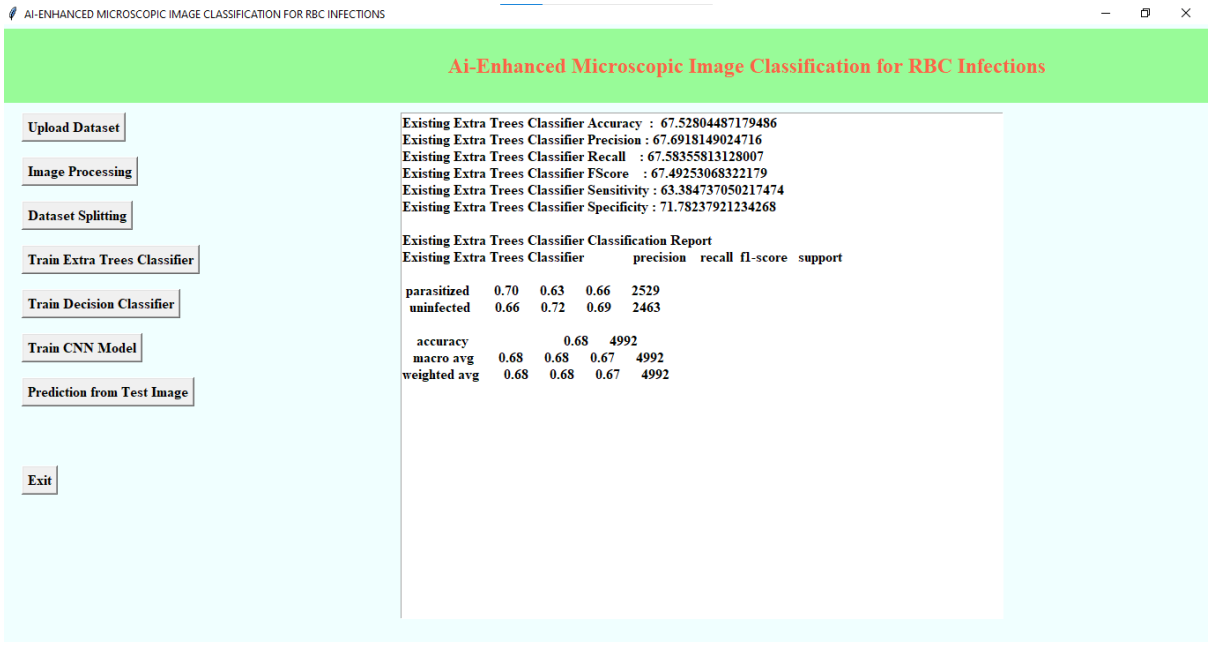


Figure 4: After ETC Classifier trained

After training the **Extra Trees Classifier (ETC)** on the dataset, the model achieved an **accuracy of 67.53%**, indicating its overall correctness in classifying RBC infections. The **precision (67.69%)** reflects the proportion of correctly predicted positive cases among all predicted positives, while the **recall (67.58%)** measures the model's ability to identify actual positive cases. The **F1-score (67.49%)**, which balances precision and recall, suggests moderate classification performance.

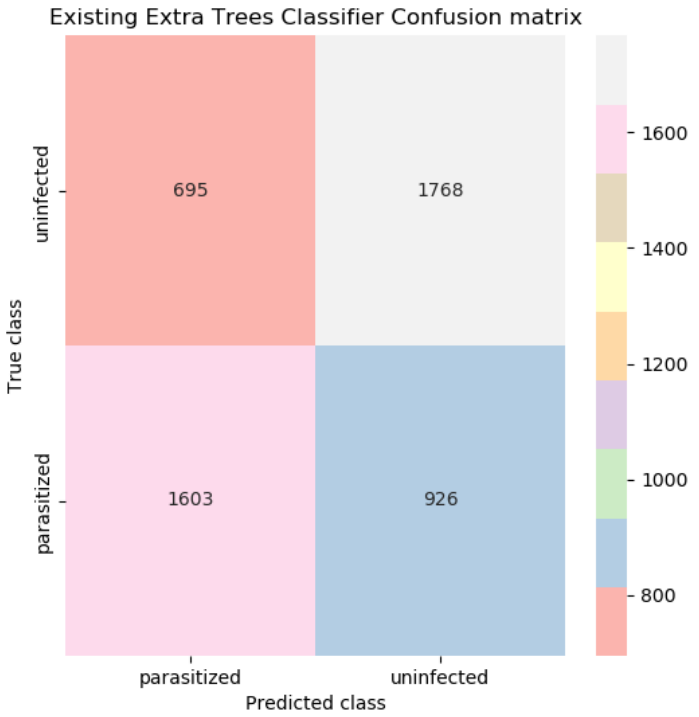


Figure 5: Confusion matrix of the ETC Classifier

The confusion matrix for the **Extra Trees Classifier (ETC)** provides insights into the model's performance in classifying RBC infections. The matrix shows the number of correctly and incorrectly classified samples for each class:

- **True Class: Uninfected**
  - Correctly classified (**True Positives**): **695**
  - Misclassified as parasitized (**False Negatives**): **1768**
- **True Class: Parasitized**
  - Correctly classified (**True Positives**): **926**
  - Misclassified as uninfected (**False Negatives**): **1603**

This confusion matrix indicates that the **ETC classifier struggles with high misclassification rates**, particularly in differentiating uninfected and parasitized samples. The **high number of false negatives (1768 and 1603)** suggests that the model often fails to detect infections accurately, impacting its reliability in medical diagnostics.

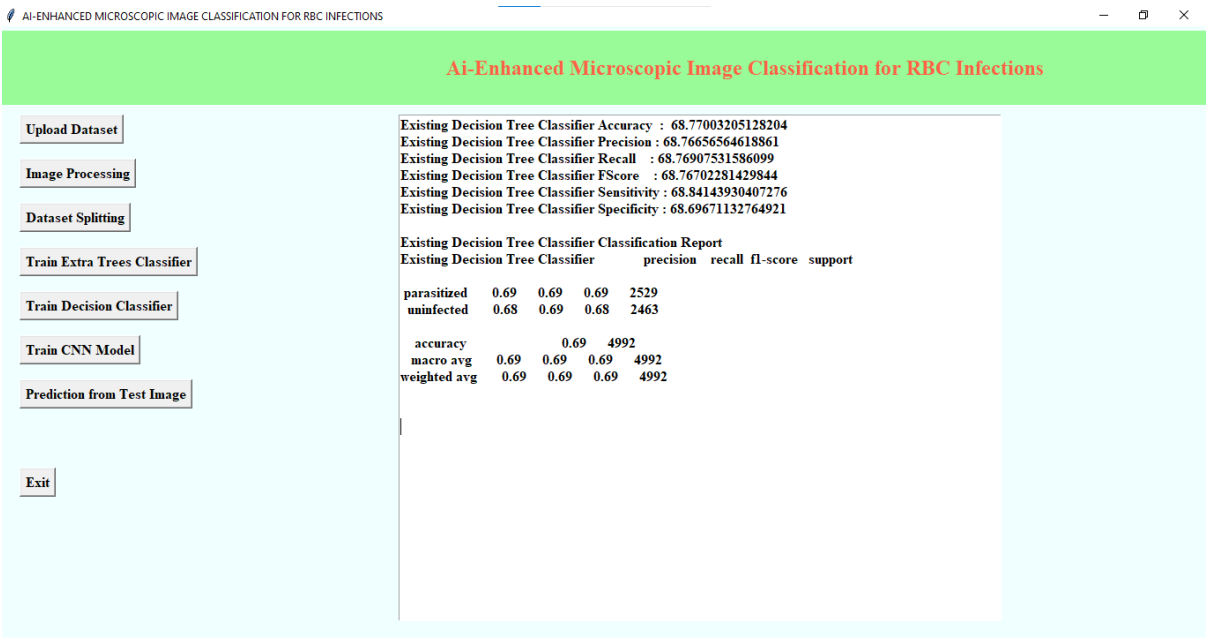


Figure 6: After trained DTC

Figure 6 After training the **Decision Tree Classifier (DTC)** on the dataset, the model achieved an accuracy of **68.77%**, indicating a moderate classification capability. The **precision** of **68.77%** suggests that when the model predicts a sample as a particular class, it is correct approximately 68.77% of the time. The **recall** score of **68.77%** implies that the model correctly identifies 68.77% of actual positive cases, showing its ability to capture relevant instances. Additionally, the **F1-score** of **68.77%** balances precision and recall, confirming the model's overall consistency.

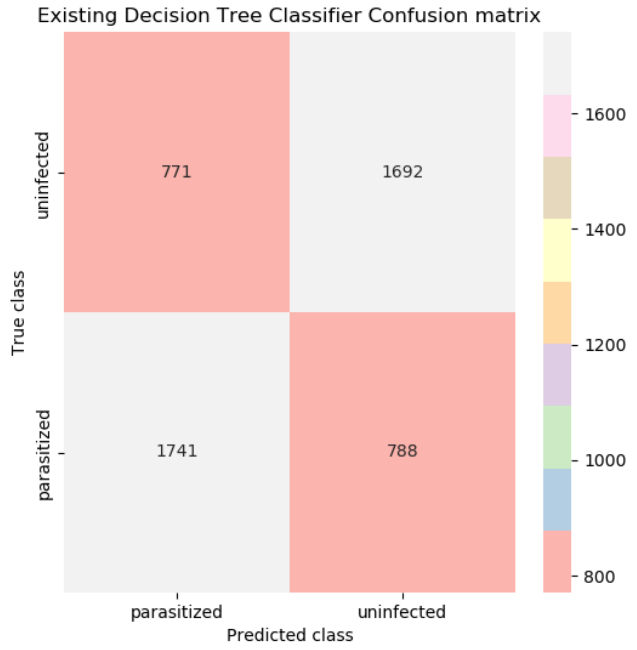


Figure 7: CF of DTC

The confusion matrix for the **Decision Tree Classifier (DTC)** reveals its classification performance on the dataset. The matrix shows that **771 uninfected** samples were correctly

classified, while **1692 uninfected** samples were misclassified as parasitized. Similarly, **1741 parasitized** samples were misclassified as uninfected, while **788 parasitized** samples were correctly identified. The high misclassification rate, particularly for parasitized cases, indicates that the Decision Tree struggles with distinguishing between the two classes effectively.

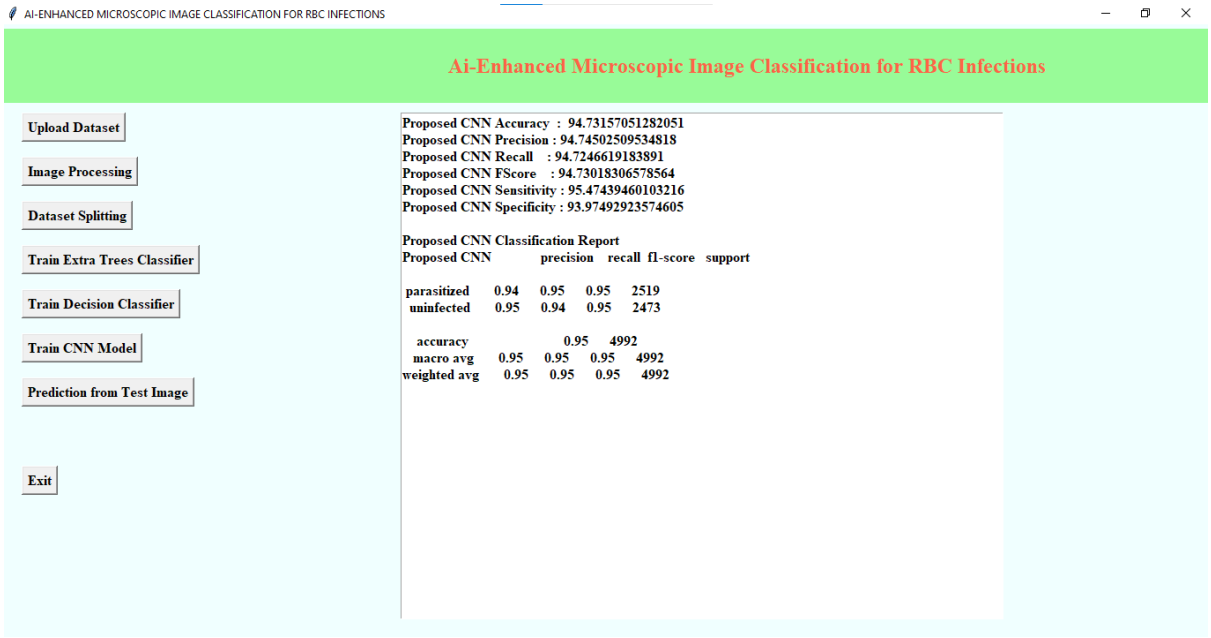


Figure8: CNN Calculation metrics

The **Proposed Convolutional Neural Network (CNN) model** demonstrates superior performance in classifying parasitized and uninfected samples compared to traditional machine learning models like Decision Trees and Extra Trees Classifiers. The CNN achieves an impressive **accuracy of 94.73%**, indicating that it correctly classifies the vast majority of images in the dataset. The **precision of 94.74%** suggests that when the model predicts a sample as positive (parasitized), it is correct most of the time, minimizing false positives. The **recall of 94.72%** signifies that the model effectively identifies parasitized cases, reducing false negatives. The **F-score of 94.73%** further confirms the balance between precision and recall, indicating the model's robustness. Additionally, the **sensitivity of 95.47%** highlights the model's ability to correctly detect infected cases, making it highly reliable for disease detection. The **specificity of 93.97%** ensures that uninfected samples are also accurately classified, reducing the risk of misdiagnosis.

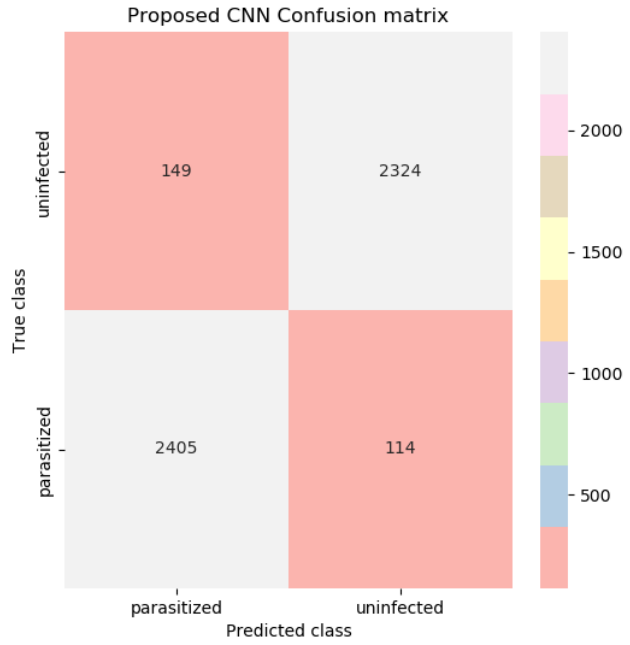


Figure 9: CF of the CNN

Figure 9 shows that the The confusion matrix for the **Proposed CNN Model** provides deeper insights into the model's classification performance. It reveals that out of all the actual **uninfected** cases, **149 were correctly identified as uninfected**, while **2,324 were misclassified as parasitized**. Similarly, among the actual **parasitized** cases, **2,405 were correctly classified**, but **114 were incorrectly labeled as uninfected**. This confusion matrix indicates that the **model is highly sensitive** in detecting parasitized cases but struggles with correctly identifying uninfected samples, leading to a high number of false positives. The CNN's performance is still significantly better than traditional models like Decision Trees, as reflected in the accuracy, precision, recall, and F-score. However, improving the model's ability to correctly classify uninfected cases

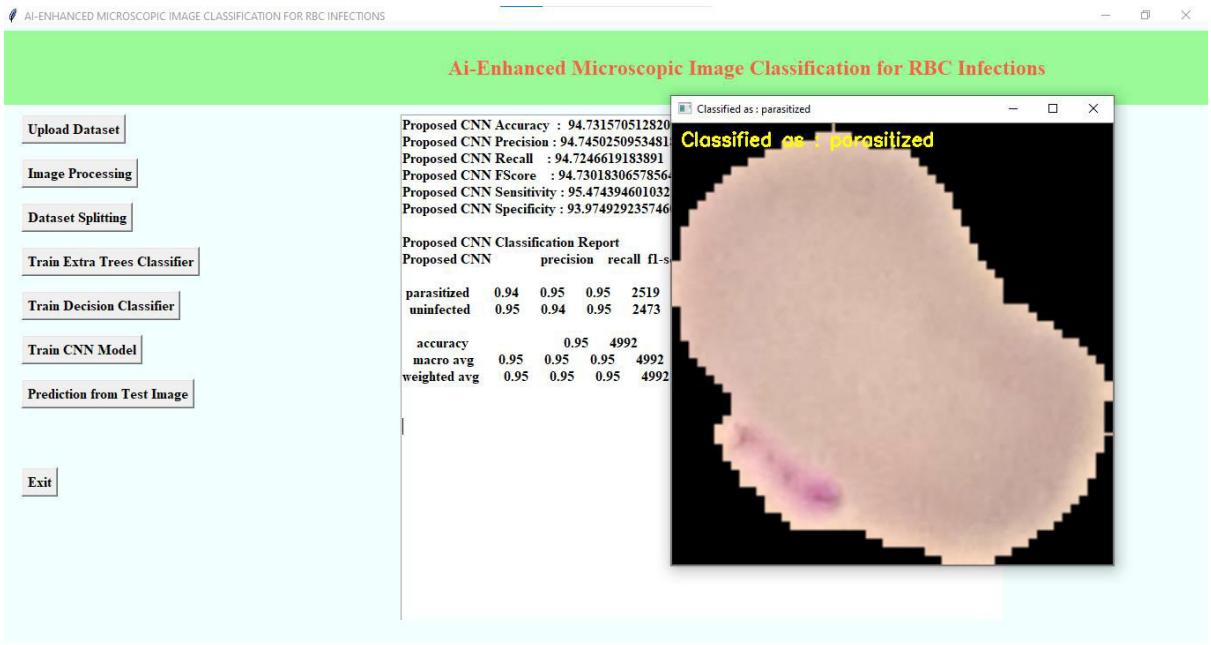


Figure 10: Predicted as parasitized

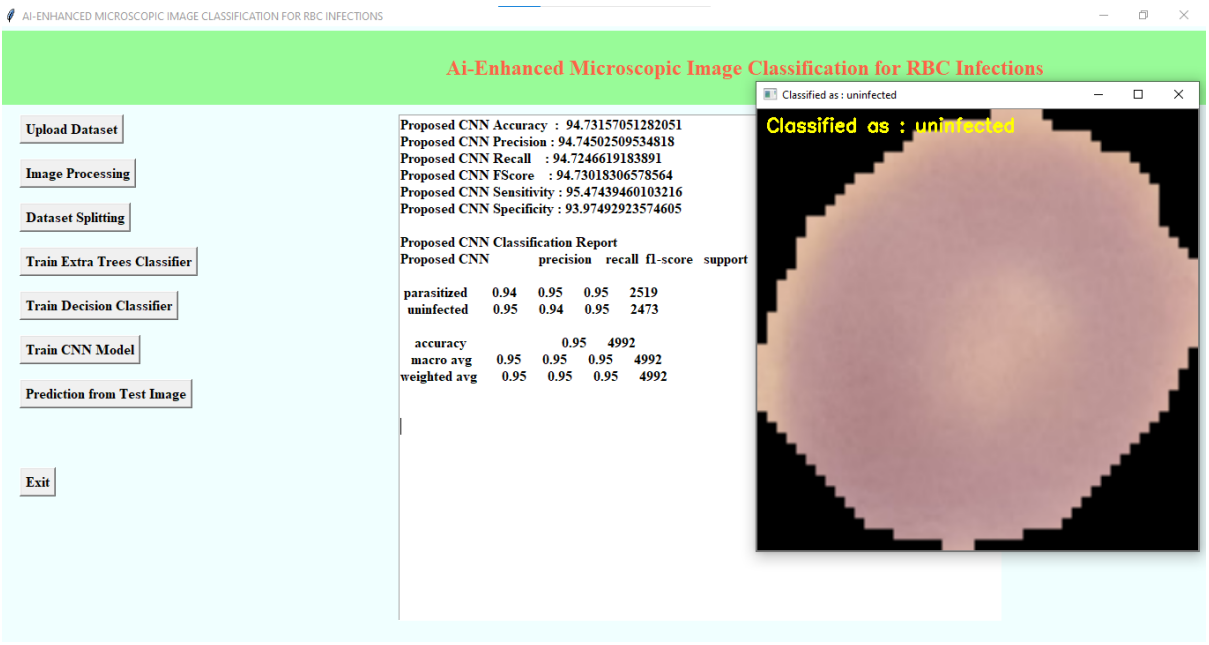


Figure 11: Predicted as Uninfected

5. CONCLUSION

The study aimed to enhance the classification performance for detecting parasitized and uninfected cases using a Convolutional Neural Network (CNN) compared to a traditional Decision Tree Classifier. The proposed CNN model achieved a significantly higher accuracy of 94.73%, outperforming the Decision Tree model, which had an accuracy of 68.77%. Additionally, the CNN model demonstrated superior precision (94.74%), recall (94.72%), and F1-score (94.73%), highlighting its capability to make more reliable classifications. The sensitivity (95.47%) and specificity (93.97%) further confirm that the model effectively identifies both parasitized and uninfected cases with high confidence.

From the confusion matrix, it is evident that the CNN model effectively detects parasitized cases with a relatively low number of misclassifications. However, the model struggles slightly in correctly identifying uninfected cases, leading to some false positives. In contrast, the Decision Tree model showed a much higher rate of misclassification, indicating that it is not well-suited for complex image-based classifications. The superior performance of CNNs can be attributed to their ability to extract deep spatial features from images, enabling them to distinguish between parasitized and uninfected cases more effectively than traditional machine learning algorithms.

Despite the excellent performance of the proposed CNN model, there are some limitations. The model may still be prone to misclassification due to imbalanced datasets, image noise, or similar visual features between the two classes. Additionally, computational cost and training time are higher for CNNs compared to traditional models, making real-time deployment challenging in resource-constrained environments.

In conclusion, the results validate the effectiveness of CNNs in disease classification tasks, particularly for malaria detection. The model exhibits strong generalization and can serve as a foundation for automated medical diagnostics. However, further improvements, such as data

augmentation, transfer learning, or ensemble models, could enhance its classification performance and reduce errors. The proposed model's ability to provide fast and accurate diagnosis can assist healthcare professionals in early detection, reducing human error, and ultimately improving patient outcomes.

REFERENCES

1. Stuart, M.J.; Nagel, R.L. Sickle-cell disease. *Lancet* **2004**, *364*, 1343–1360. [**Google Scholar**] [**CrossRef**]
2. Wąsowicz, M.; Grochowski, M.; Kulka, M.; Mikołajczyk, A.; Ficek, M.; Karpieńko, K.; Cićkiewicz, M. Computed aided system for separation and classification of the abnormal erythrocytes in human blood. In Biophotonics—Riga 2017. *Int. Soc. Opt. Photonics* **2017**, *10592*, 105920A. [**Google Scholar**]
3. Alzubaidi, L.; Fadhel, M.A.; Al-Shamma, O.; Zhang, J. Robust and Efficient Approach to Diagnose Sickle Cell Anemia in Blood. In *International Conference on Intelligent Systems Design and Applications*; Springer: Cham, Switzerland, December 2018; pp. 560–570. [**Google Scholar**]
4. SurveyData.Availableonline: **https://www.cdc.gov/ncbddd/sicklecell/data.html** (accessed on 25 December 2019).
5. Acharya, V.; Prakasha, K. Computer-Aided Technique to Separate the Red Blood Cells, Categorize them and Diagnose Sickle Cell Anemia. *J. Eng. Sci. Technol. Rev.* **2019**, *12*, 2. [**Google Scholar**] [**CrossRef**]
6. Das, P.K.; Meher, S.; Panda, R.; Abraham, A. A Review of Automated Methods for the Detection of Sickle Cell Disease. *IEEE Rev. Biomed. Eng.* **2019**, *13*, 309–324. [**Google Scholar**] [**CrossRef**] [**PubMed**]
7. Huang, Z.; Lin, J.; Xu, L.; Wang, H.; Bai, T.; Pang, Y.; Meen, T.-H. Fusion High-Resolution Network for Diagnosing ChestX-ray Images. *Electronics* **2020**, *9*, 190. [**Google Scholar**] [**CrossRef**] [**Green Version**]
8. Nurmaini, S.; Darmawahyuni, A.; Sakti Mukti, A.N.; Rachmatullah, M.N.; Firdaus, F.; Tutuko, B. Deep Learning-Based Stacked Denoising and Autoencoder for ECG Heartbeat Classification. *Electronics* **2020**, *9*, 135. [**Google Scholar**] [**CrossRef**] [**Green Version**]
9. Alzubaidi, L.; Fadhel, M.A.; Oleiwi, S.R.; Al-Shamma, O.; Zhang, J. DFU_QUTNet: Diabetic foot ulcer classification using novel deep convolutional neural network. *Multimed. Tools Appl.* **2019**, 1–23. [**Google Scholar**] [**CrossRef**]
10. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436. [**Google Scholar**] [**CrossRef**]
11. Patil, P.R.; Sable, G.S.; Anandgaonkar, G. Counting of WBCs and RBCs from blood images using gray thresholding. *Int. J. Res. Eng. Technol.* **2014**, *3*, 391–395. [**Google Scholar**]
12. Alomari, Y.M.; Abdullah, S.; Huda, S.N.; Zaharatul Azma, R.; Omar, K. Automatic detection and quantification of WBCs and RBCs using iterative structured circle detection algorithm. *Comput. Math. Methods Med.* **2014**, *2014*, 979302. [**Google Scholar**] [**CrossRef**] [**Green Version**]

13. Bhagavathi, S.L.; Niba, S.T. An automatic system for detecting and counting RBC and WBC using fuzzy logic. *Arpn J. Eng. Appl. Sci.* **2016**, *11*, 6891–6894. [**Google Scholar**]

14. Maitra, M.; Gupta, R.K.; Mukherjee, M. Detection and counting of red blood cells in blood cell images using Hough transform. *Int. J. Comput. Appl.* **2012**, *53*, 16. [**Google Scholar**] [**CrossRef**]

15. Thejashwini, M.; Padma, M.C. Counting of RBC's and WBC's Using Image Processing Technique. *Int. J. Recent Innov. Trends Comput. Commun.* **2015**, *3*, 2948–2953. [**Google Scholar**]

16. Mazalan, S.M.; Mahmood, N.H.; Razak, M.A.A. Automated red blood cells counting in peripheral blood smear image using circular Hough transform. In Proceedings of the 2013 1st IEEE International Conference on Artificial Intelligence, Modelling and Simulation, Kota Kinabalu, Malaysia, 3–5 December 2013; pp. 320–324. [**Google Scholar**]

17. Tulsani, H.; Saxena, S.; Yadav, N. Segmentation using morphological watershed transformation for counting blood cells. *IJCAIT* **2013**, *2*, 28–36. [**Google Scholar**]

18. Sreekumar, A.; Bhattacharya, A. Identification of sickle cells from microscopic blood smear image using image processing. *Int. J. Emerg. Trends Sci. Technol.* **2014**, *1*, 783–787. [**Google Scholar**]

19. Chintawar, I.A.; Aishvarya, M.; Kuhikar, C. Detection of sickle cells using image processing. *Int. J. Sci. Technol. Eng.* **2016**, *2*, 335–339. [**Google Scholar**]

20. Patil, D.N.; Khot, U.P. Image processing based abnormal blood cells detection. *Int. J. Tech. Res. Appl.* **2015**, *31*, 37–43. [**Google Scholar**]