BSCS-F24-019

03-134212-098: ZOHRA ARSHAD

03-134212-027: HAFIZ AHSAN JAVED

Automated analysis of blood-borne diseases in third-world countries

In partial fulfilment of the requirements for the degree of

Bachelor of Science in Computer Science

Supervisor: TAHIR IQBAL

Department of Computer Sciences

Bahria University, Lahore Campus

May 2025

C e r t i f i c a t e

We accept the work contained in the report titled

"Automated analysis of blood-borne diseases in third-world countries"

written by

Zohra Arshad

Hafiz Ahsan Javed

as a confirmation to the required standard for the partial fulfilment of the degree of Bachelor of Science in Computer Science.

Approved by:

Supervisor: Tahir Iqbal

_____(Signature)

13-May-2025

DECLARATION

We hereby declare that this project report is based on our original work except for citations and quotations which have been duly acknowledged. We also declare that it has not been previously and concurrently submitted for any other degree or award at Bahria University or other institutions.

Enrolment	Name	Signature
03-134212-098	Zohra Arshad	
03-134212-027	Hafiz Ahsan Javed	

Date : 13-May-2025

Specially dedicated to

my beloved grandmother, mother and father

Zohra Arshad

my beloved grandmother, mother and father

Hafiz Ahsan Javed

ACKNOWLEDGEMENTS

We would like to thank everyone who had contributed to the successful completion of this project. We would like to express our gratitude to my research supervisor,
Mr Tahir Iqbal for his invaluable advice, guidance and his enormous patience throughout the development of the research.
In addition, we would also like to express my gratitude to our loving parent and friends who had helped and given me encouragement.

Zohra Arshad
Hafiz Ahsan Javed

Automated Analysis of blood borne diseases in third world countries

ABSTRACT

Malaria is among the most important world public health issues, with considerable morbidity and mortality, particularly in World third countries. Therapy and disease control rely on fast and precise diagnosis. Traditional methods like microscopic observation of blood smears are time consuming and skill demanding, and these may not be easily available for poor-resource countries.

The proposed project is focused on automated malaria infected cell detection with the implementation of a deep learning model. We utilized convolutional neural network ANN, (CNN), VIT and the hybrid CNN+VIT model implementation with the support of transfer learning and scratch modelling for increased performance. These models were trained and tested using a custom dataset which was constructed by merging images of the NLM Malaria Dataset and the Malaria Cell Images Dataset on Kaggle, totalling 27,500 images divided into four malaria classes—Plasmodium Falciparum, Plasmodium Vivax, Plasmodium Ovale, Plasmodium Malariae—and one class for non-infected. The model was executed efficiently with high precision, accuracy, and recall and hence is a reliable system to support malaria diagnosis.

Also, a web interface was developed using React.js for uploading cell images and obtaining the malaria infection predictions, thereby making the technology convenient and accessible. The article reflects a good account of method, experiment, and result and demonstrates the effectiveness of deep learning in medical image processing.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS / ABBREVIATIONS

CNN	Convolutional Neural Network
ANN	Artificial Neural Network
ViT	Vision Transformer
AI	Artificial Intelligence
Open cv	Open-source computer vision
TF	TensorFlow
MD5	Message Digest Algorithm 5

CHAPTER 1

1. INTRODUCTION

1.1 Background

Diagnosis of malaria by microscopic visualization consists of finding Plasmodium parasites within stained blood films. This procedure, while being extremely specific and sensitive
while conducted by well-trained technicians, is not always possible in those areas with poor access to well-trained technicians and good laboratory facilities. The difference in the quality of microscopy and subjective nature of the test can also cause variable results [1].

In the last decade, medical image analysis has improved significantly with the help of machine learning and deep learning. Both Convolutional Neural Networks (CNNs) and Vision Transformers (ViTs) are utilized extensively to build automated systems to diagnose diseases such as cancer, diabetic retinopathy, and pneumonia. They are capable of learning intricate patterns from large sets of data and tend to work as well as, or sometimes even better than, human clinicians.

CNNs are good at identifying patterns in images and are thus beneficial for medical diagnosis. ViTs, however, break down images into small patches and treat them as a sequence, enabling them to pick up significant details that CNNs may not. Through the integration of these two methods, the model can perform more accurately and better in medical image analysis.

1.2 Problem Statements

Malaria remains a severe global public health problem, impacting an estimated millions of people across the world annually, primarily in the tropics and subtropics of the world. It is crucial that effective diagnosis and effective treatment and control be achieved. Developments within artificial intelligence (AI) and deep learning provide potential solutions for computerized malaria detection. Machine learning algorithms, especially Convolutional Neural Networks (CNNs) and Vision Transformers (ViTs), have been found to possess valuable applications in medical image analysis, with performance on par with human experts [2]. Conventional microscopy-based diagnosis of malaria is time-consuming, technician-trained, and prone to human error and results in delay and misdiagnosis. The drawback of the traditional microscopy-based diagnosis calls for an excellent and automated diagnostic system.

The study seeks to develop an AI-driven malaria detection system using Convolutional Neural Networks (CNNs) and Vision Transformers (ViTs) with enhanced diagnostic efficiency and accuracy. It is trained on in-house data of 27,500 images of the NLM Malaria Datasheet and the Malaria Cell Images Dataset on Kaggle, which is categorized into four malaria types—Plasmodium Falciparum, Plasmodium Vivax, Plasmodium Ovale, Plasmodium Malariae—and a single healthy class.

1. Limitations of Traditional Diagnosis: Highly trained microscopists are required for traditional techniques, and screening is problematic at scales larger than individual slides.

2. Blood Smear Images Differ: Variation in cell morphology, staining procedure, and image quality complicates computer-assisted classification.

3. High Speed and Accuracy Required: Automated systems require high recall and precision and rapid reporting to support real-time diagnosis.

4. Generalization Across Malaria Species: Most of the current models only distinguish infected and non-infected cells, whereas the current research aims to distinguish between four species of malaria for better diagnosis.

Using deep learning strategies, the work herein is an attempt to devise a cost-efficient and scalable machine for malaria detection that will be used to support clinicians in minimization of misdiagnosis errors, enhance detection early enough, and provide accessibility to accurate malaria screening in environments of poor resource settings.

1.3 Aims and Objectives

The objectives are listed as following:

To design an AI-based malaria diagnosis system using Convolutional Neural Networks (CNNs) and Vision Transformers (ViTs), with web frontend application, for enhancing accuracy, efficiency, and accessibility of malaria diagnosis.

1. Develop and Train the Model – Design and train deep learning models (CNNs and ViTs) on a private dataset of five classes
   (four types of malaria+ non-infected).

2. Improve Performance – Obtain the best possible precision, recall, and accuracy and prevent issues such as class imbalance and image variability.

3. Develop a Web-Based Interface – Develop a readily accessible web-based interface through which users can easily upload and scan for malaria blood smears.

4. Compare the CNN and ViT Models – Identify the limitations and strengths of both models to determine the most appropriate method for the malaria classification.

5. To enhance diagnostic efficacy and precision through a specific dataset of 27,500 blood smear images.

1.4 Scope of Project

This project seeks to automate malaria detection using deep learning. The scope includes:

1. Constructing CNN, ANN, and ViT models for malaria classification.

2. Training the models on a custom dataset of 27,500 blood smear images from the NLM Malaria Dataset and Kaggle.

3. Creating a web application for user interaction.

4. Testing the model's performance on the dataset.

CHAPTER 2

LITERATURE REVIEW (and/or SRS)

2.1 Background

Malaria is still one of the most deadly and prevalent vector-borne disease, especially in the third world like sub-Saharan Africa and South Asia. It is caused by parasites of the genus Plasmodium, transmitted to human populations by bites from infected Anopheles mosquitoes. Proper and early diagnosis is crucial so that treatment can be rendered effective as well as minimize mortality. Conventional diagnostic techniques, including microscopic inspection of blood smears, are time-consuming and need skilled personnel, and this can result in delays in diagnosis and treatment, particularly in resource-constrained environments [1].

New advances in computer vision and deep learning have shown tremendous potential in the automation of medical image analysis.
Convolutional Neural Networks (CNNs), for example, have established new standards in image classification, segmentation, and object detection [4]. More recent innovations such as Vision Transformers (ViT) have ensued, with greater generalization capability via better capture of global image relations compared to previous CNNs [5].

This project employs deep learning, i.e., CNN and ViT models, to develop an automated malaria diagnostic system. It aims to introduce a fast, efficient, and cost-effective blood smear image analysis tool that can substantially assist healthcare professionals in remote or resource-poor areas.

Table 2.1 Literature Review

Sr	Name	Year	Accuracy	Model used	Methodology	Gaps
1.	Recent Advancements in Detection and Quantification of Malaria Using Artificial Intelligence	2024	97%	VGG19	Analysis of AI methods, particularly convolutional neural networks (CNNs), for malaria detection.	-Lack of datasets -The robustness of algorithms -Non-interpretability -Poor integration into clinical practice.
2.	Machine and deep learning methods in identifying malaria through microscopic blood smear: A systematic review	2024	99.67%	CNN	Systematic review applying the PRISMA method to assess and condense pertinent literature related to automated malaria diagnosis	- Inadequate validation techniques - Lack of generalization - Insufficient data variability - Low classification accuracy
3.	Deep Learning Enhanced Extended Depth-of-Field for	June 2019	73%	CNN	Evaluated image quality by comparing using the Structural Similarity Index (SSIM)	-Limited Field of View (FOV) -Lack of exploration in using CNNs

	Thick Blood-Film Malaria High-Throughput Microscopy				and parasite detection accuracy by testing with a CNN-based object detector.	-Ineffective analysis of multiple focal planes -Insufficient robustness of performance metrics
4.	Metrics to guide development of machine learning algorithms for malaria diagnosis	2022	81%	CNN	The article talks about the use of machine learning algorithms in automated malaria diagnosis by microscopy on Giemsa-stained blood smears. It highlights the use of clinical expertise and task-specific performance measures to make the algorithms more appropriate.	-Neglect of Clinical Needs -Fitted Performance Measure -Patient Level Priority -Embedment of Domain Expertise -Practice Performance Problems
5.	Efficient deep learning-based approach for malaria detection using red	2024	97.57%	Efficient Net	The research uses a deep learning model (EfficientNet) to classify red blood cell images for detecting	-Inefficiency and time taken by malaria detection processes - Delays in diagnosis - Low precision -Misdiagnosis using traditional methods

	blood cell smears				malaria, with k-fold cross-validation used to validate performance.	- Inadequate usage of sophisticated calculation
6.	Trials and tribulations: Developing an artificial intelligence for screening malaria parasite from peripheral blood smears	2023	94%	Inception V3	Deep learning techniques (such as convolutional neural networks, watershed segmentation, Inception V3 transfer learning, and multi-class classifiers) that are used in stained blood smear images for detection of parasites. The method encompasses segmentation, classification, and assessment on microscopic images.	-Limited automation for whole slide images. -Time-consuming and skill-dependent manual methods. -Moderate sensitivity and specificity. -Lack of datasets with diverse cell types and parasite stages. -Dependence on internet and manual photo capture in mobile apps.
7.	Developing an Artificial Intelligence for Screening Malaria	2023	94%	Inception V3	The research employed convolutional neural networks (CNNs), watershed transformation for	-Previous datasets were limited to manually cropped RBCs that didn't include WBCs and

	Parasite from Peripheral Blood Smears				segmentation, and combined models (multi-class and binary classifiers) that were trained on Giemsa-stained blood smear images. Segmentation of WBCs and RBCs was followed by classification of infected cells.	platelets. -Most of the earlier models considered only thick smears and did not operate with whole slide images. -Previous tools needed manual microphotography and internet access, which made them infrastructure- and observer-dependent. -No current models sufficiently included detection of various parasite stages, like gametocytes.
8.	Developing an artificial intelligence for screening malaria parasite from peripheral blood smears	2023	94%	Naive DCNN	Designed five deep learning models (A, B, C, D, E) for identifying malaria parasites through image segmentation and classification methods.	-Earlier models did not successfully strike a Balance between sensitivity and specificity in detecting malaria. -Current techniques were hampered by differences in image quality and resolution.

						-Absence of real-time detection efficacy in earlier studies. -Earlier models tended not to generalize effectively across different datasets.
9.	Deep Learning Based Automatic Malaria Parasite Detection from Blood Smear and Its Smartphone Based Application	2020	99.23%	8-layer CNN	Implemented a model of malaria parasite detection based on a publicly available dataset with emphasis on accuracy and computational costs	-Earlier models were inefficient for use in power-restricted devices. -Previous approaches failed to be computationally efficient with high accuracy. -Ineffective diagnostic tools being poorly accessible in resource-poor environments.
10.	Smartphone-enabled personalized diagnostics: current status and future	2021, June	94.40%	Faster R-CNN	Statistical comparison (t-test) of performance of CNN models using precision, recall, F-score, and mAP0.5 as metrics.	-Inadequate sensitivity and specificity testing of the complete diagnostic system. -Poor integration of smartphone features into customized diagnostics. -Lack of rigorous

						testing of diverse populations and conditions.
11.	Advances and Challenges in Automating Malaria Diagnosis Using Digital Image Analysis and AI	November 2022	90.00%	(CNNs), including YOLO variants	Review of recent AI-based methods with emphasis on digital image analysis, deep learning (CNN models), and smartphone incorporation for malaria detection.	- Insufficiency of universal protocols for image acquisition and diagnosis in various settings. -Requirement of further optimization of automated microscopy and hardware automation to be used independently.

CHAPTER 3

DESIGN AND METHODOLOGY

3.1 Methodology

3.1.1 Dataset Acquisition

The project starts off with the initial step of acquiring an appropriate dataset. Luckily, there are many freely accessible datasets, and these are a suitable starting point for most research projects. For the best possible result for the project, a careful selection process was undertaken. Some of the leading datasets available on Kaggle were selected for consideration, such as the "Malaria Cell Images Dataset", "Malaria Detection", and "Malaria Parasite Detection". After a thorough review, the final dataset was generated by combining NLM Malaria Datasheet and Malaria Cell Images Dataset images. The gathered dataset provides a diverse and broad corpus of blood smear images, necessary to train highly effective and potent deep learning models.

3.1.2 Pre-processing

Once the dataset is received, it undergoes a high-level pre-processing process before analysis. The dataset contains 27,500 images of four malaria blood smear species—Plasmodium Falciparum, Plasmodium Vivax, Plasmodium Ovale, and Plasmodium Malariae—and one extra class, non-infected or healthy. Invalid images, i.e., faulty images (unavailable file or processable file format), are discarded, thus high-quality images alone are used for training and testing. Library such as Python's PIL (Pillow) or OpenCV may be utilized to identify corrupted images by trying to open all the images and then raising an exception. The same images are also labelled and deleted so as to avoid dataset duplication of data. It is done by identifying hash values (such

as MD5 or perceptual hashing) of all the images and comparing them to identify duplicates. Duplicate removal prevents the model from learning on duplicate images so that the model can generalize. All images are resized to a standard resolution of 224x224 pixels for ease of compatibility and conformity with deep learning architecture. These are methods like rotation, flip, zoom, and brightness applied to enhance diversity in the training data and avoid overfitting. The database is also properly balanced to cover all classes without a preferred class. These are oversampling minority classes or under sampling majority classes. Lastly, the data are divided into training, validation, and test sets (e.g., 70% training, 15% validation, and 15% test) in order to have sufficient model performance evaluation. The dataset is hence sanitized and cleaned for future modelling and analysis and input to deep learning models is guaranteed.

3.1.3 Feature Engineering

Once the dataset is cleaned and pre-processed, the subsequent step is featuring engineering, where useful patterns are derived from the blood smear images to enhance model performance. For this task, CNN, ANN, and ViT models were coded from scratch to learn features. CNNs learned spatial features such as edges and textures, ANNs learned flattened image data for easy patterns, and ViTs split images into patches to learn global relationships. Merging these characteristics enriched the dataset, allowing the model to learn well and attain high accuracy in detecting malaria.

3.1.4 Model building

Following feature engineering, **CNN**, **ANN**, and **ViT** models were implemented **from scratch** and tested for malaria detection. CNNs extracted spatial features like edges and textures, ANNs processed flattened image data for simpler patterns, and ViTs captured global relationships by treating images as patch sequences. Each model was evaluated individually, and their performances were compared to identify the most effective approach, ensuring high accuracy and robustness for the project.

3.1.5 Evaluation

Once the model is built, the evaluation process begins. We measure the performance of the model through several evaluation metrics such as Training Accuracy, Testing Accuracy, Validation Accuracy, Precision, Recall, and F1 Score. This must be done to ensure the quality of the model and whether it meets the goal of the project or not.

3.1.6 Web Deployment

In the end, the trained model is deployed onto a web application.

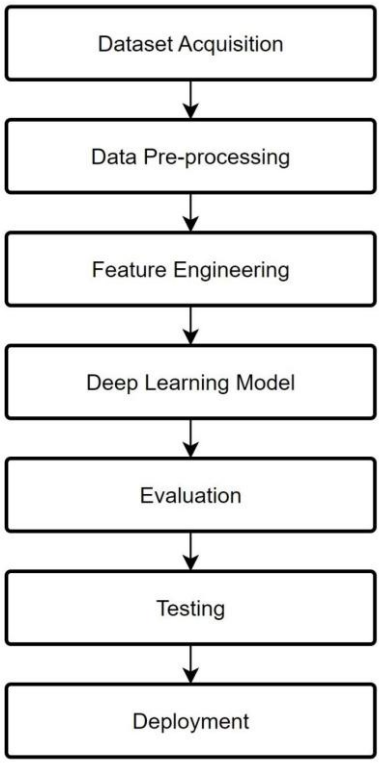

Figure 3.1: Methodology Flow Diagram

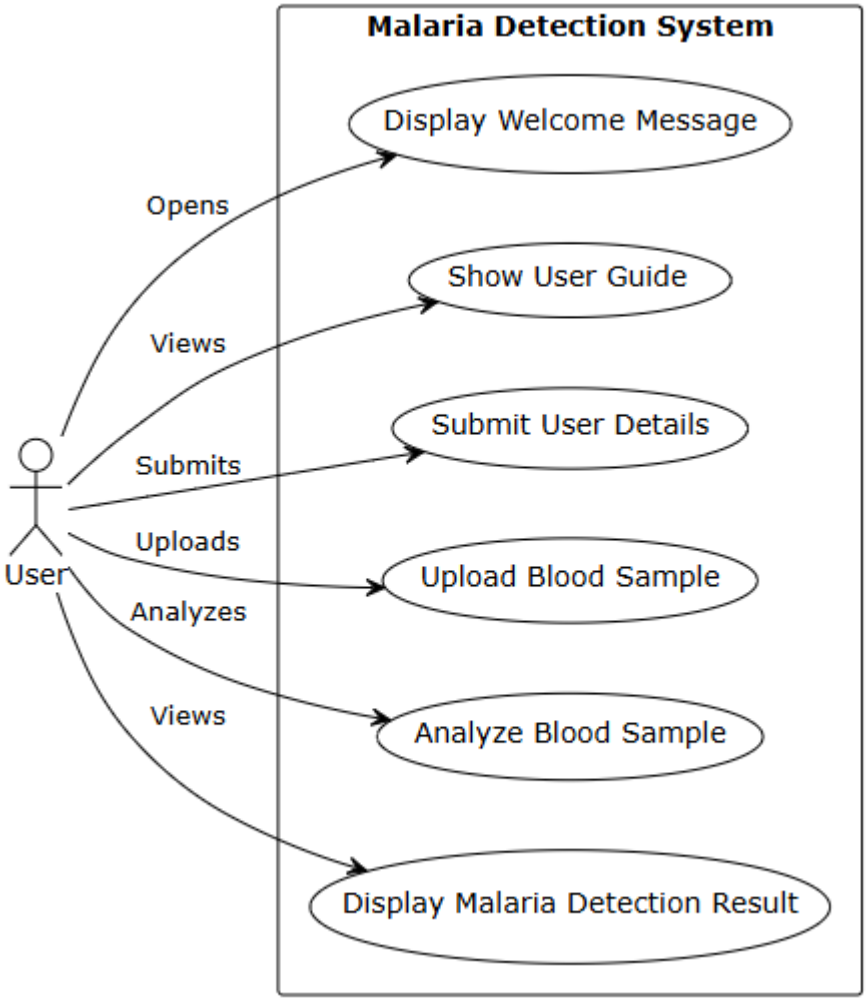

Figure 3.2 Use Case Diagram

3.1.7 Use Cases

Table 3.1 Use Cases

Use Case ID	Use Case Title	Actors	Preconditions	Postconditions	Description
UC-01	Display Patient Form	Lab technician or doctor	User is on the Home page.	The form is visible with all required fields.	The system shows a form to enter patient's details like name, age, and email.
UC-02	Enter and Validate Patient Details	Lab technician or doctor	Form is displayed.	All fields pass validation.	User inputs valid patient data into the form.
UC-03	Data for Report	System	All fields are valid.	Data is available on /upload and /report pages.	After submission, patient data is available for use in the final report.
UC-04	Display Data on Diagnostic Report	System	Prediction has been made.	Report displays name, age, and email of the patient along with test result.	Stored patient details are fetched and shown on

					the final report PDF or UI.
UC-05	Upload Blood Smear Image	User	Access to web application	Image uploaded	User selects and uploads a blood smear image.
UC-06	Receive Malaria Prediction	User	Image uploaded	Prediction results received	System processes the image and returns prediction.

3.1.8 Test Cases

Table 3.2 Test Cases

Test Case ID	Test Case Title	Test Steps	Expected Result
TC-01	Verify Image Upload Functionality	1. Navigate to upload page. 2. Select valid image. 3. Click 'Upload.'	Image uploads successfully.
TC-02	Validate Prediction Accuracy	1. Upload known infected image. 2. Wait for prediction.	Prediction matches known result.
TC-03	Check Report Download Functionality	1. Upload image. 2. Click 'Download Report.'	Report downloaded with prediction details.
TC-04	Submit Valid Patient Data	1. Open Home page. 2. Enter valid name, age, and email. 3. Click 'Submit.'	Data is saved and user is redirected to /upload.
TC-05	Missing Required Field	1. Open form. 2. Leave one field (e.g., email) blank. 3. Click 'Submit.'	Error message appears; form is not submitted.
TC-06	Invalid Email Format	1. Enter incorrect email (e.g., zohra@com). 2. Click 'Submit.'	Validation error is shown.

TC-07	Data Appears in Report	1. Submit form. 2. Upload image. 3. Download report.	Report includes patient name, age, email, and diagnosis result.
TC-08	Form Submission Button Disabled During Submission	1. Fill form correctly. 2. Click 'Submit' and observe button state.	Button is disabled and shows 'Processing…' to prevent resubmission.

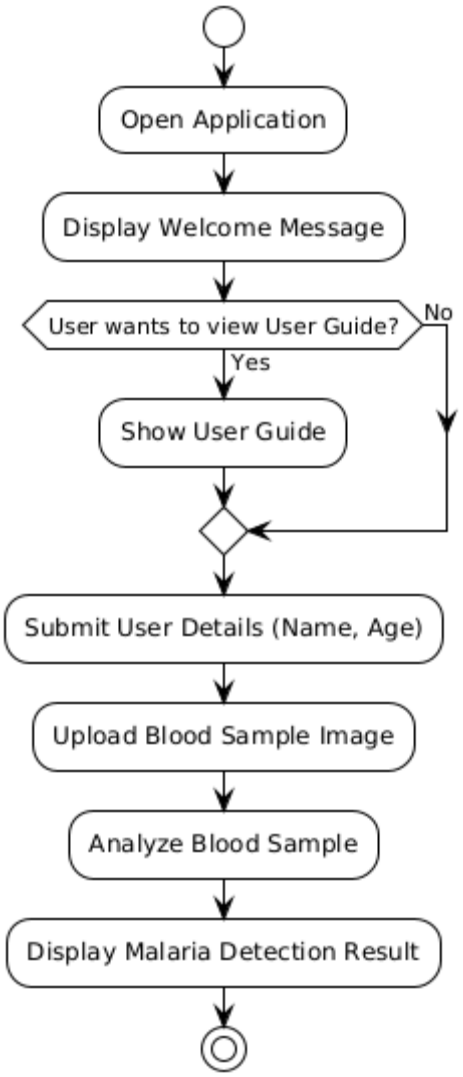

Figure 3.3 Activity Diagram

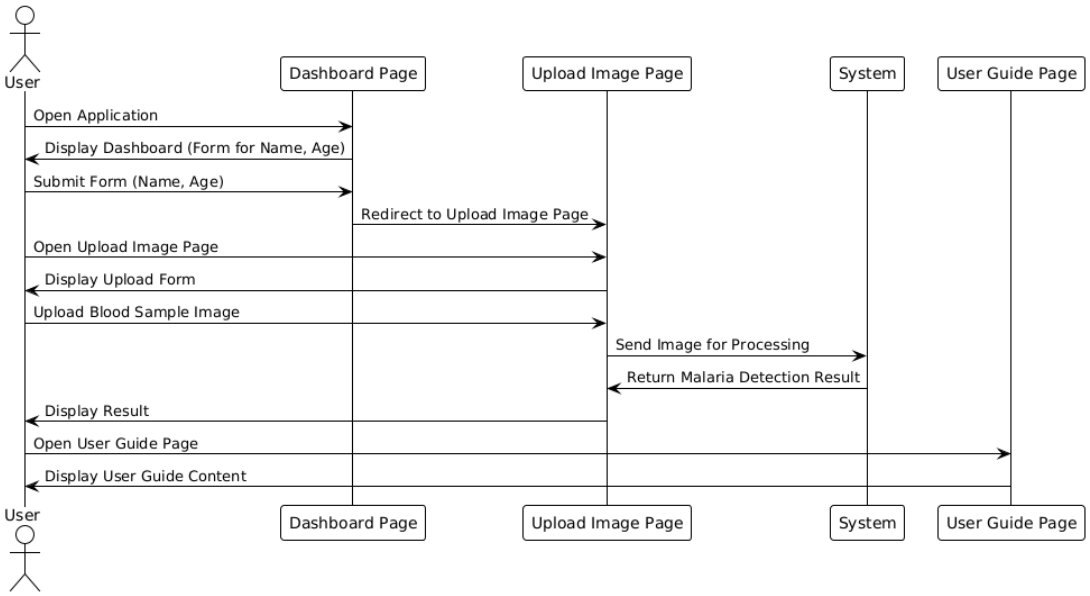

Figure 3.4 Sequence diagram

CHAPTER 4

DATA AND EXPERIMENTS (and/or IMPLEMENTATION)

4.1 Data Pre-processing and Exploration

The process of data preparation before the in-depth explanation of feature engineering and model building process. This section is about preparation and overview of the dataset such as data acquisition, data cleaning and the process of preparing the organized data set into three parts i.e. train data set, validation data set and test data set. These initial steps are critical to ensure that the data to be used in the modelling are appropriate and that, by extension, the models derived from the data are learnable, testable, and estimable.

4.1.1 Data Acquisition

The system learns from two most important sources: the NLM Malaria Dataset (clinically confirmed samples annotated by experts) and Kaggle repositories (representative field-quality images). Combined, this training ensures both high diagnostic accuracy and real-world practicability. All images passed quality screening for excluding poor quality smears but preserving required variability in staining and magnification.

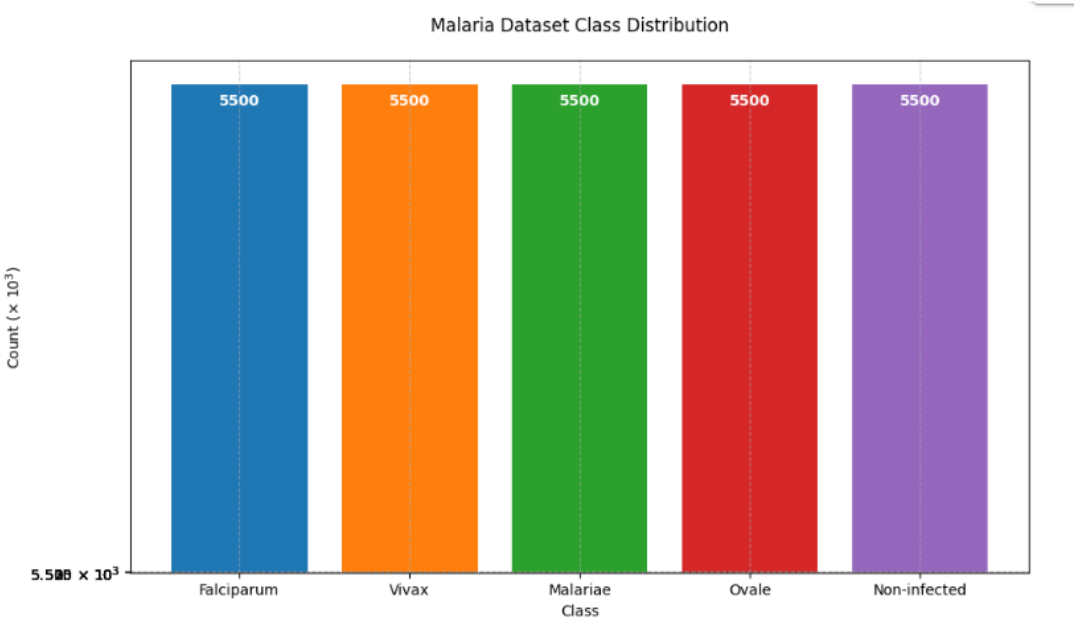

Figure 4.1 Balanced dataset

4.1.2 Data Cleaning and Pre-processing

Malaria blood smear images were under stringent preprocessing to make them meet both consistency and quality criteria. Poor quality samples (defocused, low-quality staining, or artefact-ridden images) were filtered out by means of automatic quality evaluation processes and visual screening. The remaining images were normalized compared to staining in order to reduce colour distribution heterogeneity among different laboratories. Each image was cropped to contain only relevant cellular areas and resized into a consistent resolution. Rotation, flipping, and contrast changes were used as data augmentation methods to enhance dataset diversity without losing biological characteristics. The images were then annotated from source dataset professional annotations and the parasite-infected cells distinctly outlined for supervised learning. This diverse preprocessing pipeline ensured the stability of the dataset while ensuring the preservation of the required biological variations for effective model training.

4.1.3 Dataset Split

Most of the data is available in the training set, which is used by the machine learning models (CNN ANN, Vits) to learn from the patterns and features in the data.

Depending on the dataset size and complexity of the model to be learned, the training set is 70% of the dataset and the same quantity of images in each class.

The training, validation, and test sets comprise the custom dataset. The CNN, ANN, and Vit models are trained using patterns and characteristics found in the training set, which is made up of most of the data. To avoid overfitting, Early Stopping and Patience parameters are used.

4.1.3.1 Training Set

Most of the dataset is included in the training set, which is what the machine learning models (CNN ANN, Vits) utilize to learn from patterns and characteristics in the data. Depending on the size of the dataset and the complexity of the models being trained, the training set comprises 70% of the dataset, equal number of images from each class.

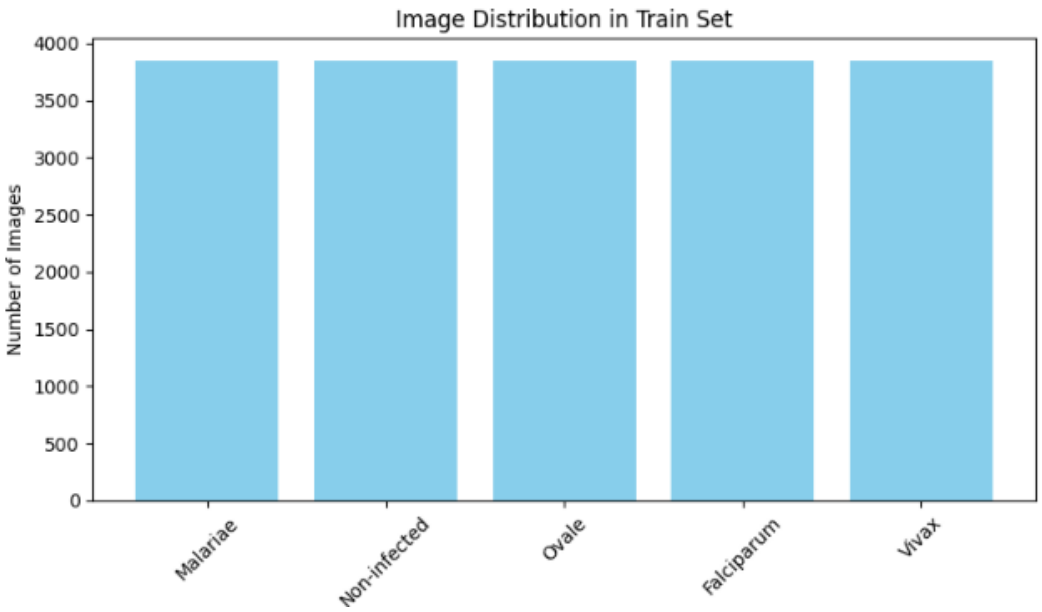

Figure 4.2 Equal Images Distribution in Training Set

4.1.3.2 Test set

The test set is used to measure the trained models' ultimate performance and their capacity to predict new data. The test set receives 15% of the total dataset.

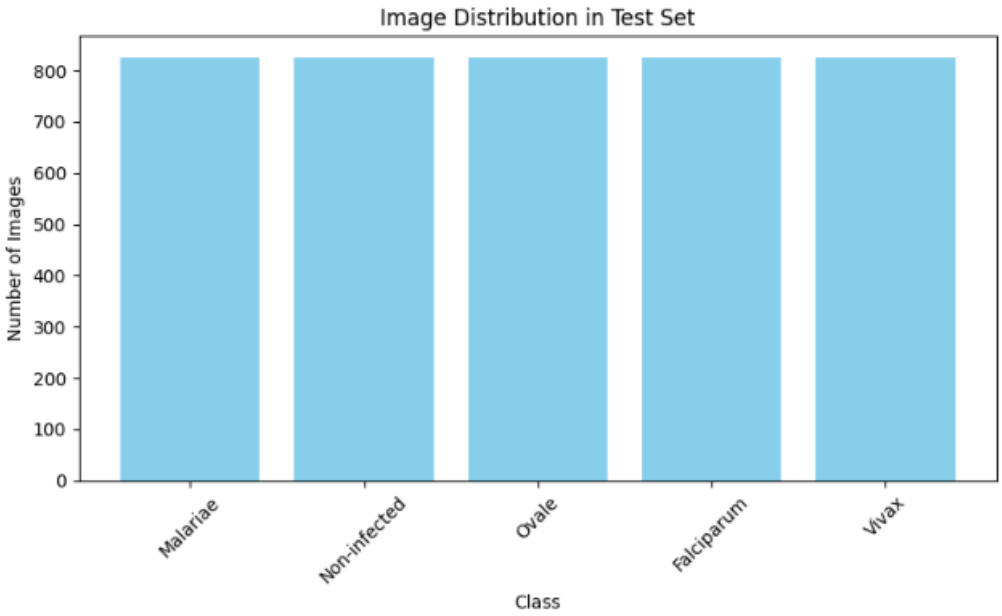

Figure 4.3 Equal Images Distribution in Test Set

4.1.3.3 Validation set

Validation data is used to evaluate the model's performance during training and help tune hyperparameters. It ensures the model generalizes well and prevents overfitting before testing on unseen data. The validation set receives 15% of the total dataset.

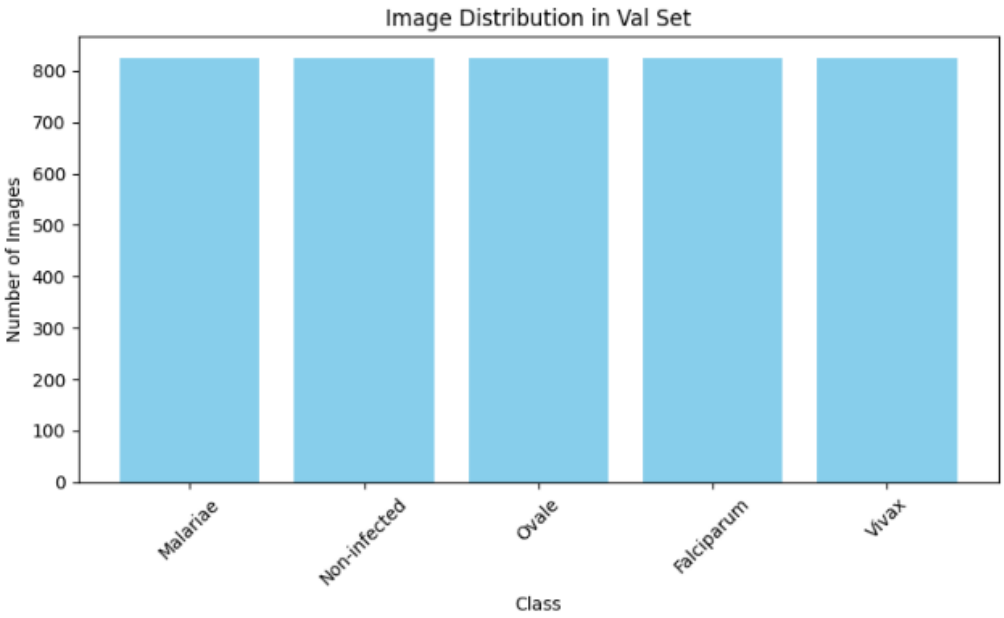

Figure 4.4 Equal Images Distribution in Validation Set

4.2 Experimentation and Model Development

The subsequent section lays down the phases in model development and experimentation, like feature engineering, model structure, model selection and training, and model selection. All of these are clearly detailed to present an exhaustive detail about the procedure.

4.2.1 Feature Extraction

is an inescapable process step of machine learning involving selection, transformation, and determination of pertinent features out of unmodified data towards the best model performance. It helps more robust learning capability among algorithms by nailing down most salient information about the data and redundancy elimination as well as noise removal. In image classification tasks, such as detecting malaria, feature engineering may include resampling images, pixel intensity normalization, data augmentation (rotation, flip), or deep feature extraction using pre-trained models. All the above approaches enhance generalization capability of the model and training time

and reduce overfitting. Efficient feature engineering closes the gap between raw data and informative information, thus leading to good and stable prediction models.

CHAPTER 5

RESULTS, DISCUSSIONS & USER MANUAL

5.1 ANN

The number of samples in the dataset was 27500. First, we requested the ANN model for training. We employed various parameters to tune the model (LR, BS, no of layers). The history of our training with metrics (loss, accuracy) is below:

Table 5.1 Sample for ANN

Total No of samples	27500
No of training samples	19245
No of testing samples	4130
No validation samples	4125

1) Hyper Parameters: Epochs = 5, Batch Size = 32, LR = 0.001

Table 5.2 ANN 1st Training History

No of Epochs	Accuracy in %	Loss (Training)	Accuracy of Validation	Loss of Validation
1	0.2891	9.1088	0.5439	1.2870
2	0.4375	1.5591	0.4817	1.4090
3	0.5266	1.2760	0.4883	1.1710
4	0.5000	1.2089	0.5693	0.5693
5	0.6531	0.9603	0.5046	1.2370

2) Hyper Parameters: Epochs = 10, Batch Size = 32, LR = 0.001

Table 5.3 ANN 2nd Training History

No of Epochs	Accuracy in %	Loss (Training)	Accuracy of Validation	Loss of Validation
1	0.6983	0.8405	0.5098	1.7565
2	0.5312	2.0812	0.4106	1.7085
3	0.7070	0.8051	0.8042	0.5750
4	0.9062	0.3911	0.7961	0.5660
5	0.7777	0.6382	0.8120	0.5430
6	0.8438	0.5223	0.8088	0.5558
7	0.7693	0.6473	0.7825	0.5715
8	0.6875	0.6488	0.8020	0.5267
9	0.7961	0.5867	0.8533	0.4783
10	0.9062	0.4178	0.8650	0.4297

3) Hyper Parameters: Epochs = 15, Batch Size = 32, LR = 0.001

Table 5.4 ANN 3rd Training History

No of Epochs	Accuracy in %	Loss (Training)	Accuracy of Validation	Loss of Validation
1	0.8177	0.5325	0.8030	0.5683
2	0.8438	0.4905	0.8142	0.5316
3	0.8134	0.5446	0.8083	0.5221
4	0.7500	0.7074	0.8391	0.4665
5	0.8136	0.5364	0.8633	0.3986
6	0.8750	0.2881	0.8689	0.3827
7	0.8225	0.5064	0.7888	0.5991

8	0.6875	0.7859	0.7947	0.5836
9	0.8248	0.5182	0.7905	0.5974
10	0.9375	0.3561	0.7971	0.5358
11	0.8281	0.5097	0.8335	0.4970
12	0.8125	0.5759	0.8259	0.4952
13	0.8388	0.4796	0.8145	0.4899
14	0.8438	0.5624	0.8374	0.4429
15	0.8398	0.4652	0.8364	0.4673

4) Hyper Parameters: Epochs = 10, Batch Size = 32, LR = 0.001, Patience = 4

Table 5.5 ANN 4th Training History

No of Epochs	Accuracy in %	Loss (Training)	Accuracy of Validation	Loss of Validation
1	0.8235	0.5134	0.8704	0.3947
2	0.9688	0.2597	0.8855	0.3703
3	0.8327	0.4899	0.8298	0.5130
4	0.8750	0.4551	0.8367	0.4618
5	0.8330	0.4887	0.8635	0.3712
6	0.9062	0.3857	0.8743	0.3471
7	0.8444	0.4666	0.8369	0.4669
8	0.7500	0.6093	0.8486	0.4160
9	0.8450	0.4540	0.8447	0.4567
10	0.9062	0.2892	0.8503	0.4577
11	0.8309	0.4994	0.8164	0.5095
12	0.7188	0.7796	0.8088	0.5246
13	0.8602	0.4181	0.8848	0.3641
14	0.7812	0.6616	0.8936	0.3503
15	0.8493	0.4423	0.8562	0.4438

16	0.7188	0.5657	0.8628	0.4172
17	0.8544	0.4234	0.8652	0.3839
18	0.9062	0.3642	0.8708	0.3573
19	0.8547	0.4224	0.8833	0.3493
20	0.8750	0.3382	0.8792	0.3688

5.2 CNN

The number of samples in the dataset was 27500. We after applying ANN to CNN model used it for training. We applied various parameters for fine tuning the model (LR, BS, no of layers). The training history of us including metrics (loss, accuracy) is displayed below:

Table 5.6 Sample for CNN model

Total No of samples	27500
No of training samples	19245
No of testing samples	4130
No validation samples	4125

1) Hyper Parameters: Epochs = 5, Batch Size = 32, LR = 0.001, Patience = 2

Table 5.7 CNN 1st training history

No of Epochs	Accuracy in %	Loss (Training)	Accuracy of Validation	Loss of Validation
1	0.7778	0.6669	0.8761	0.3520
2	0.8864	0.3321	0.9008	0.2736
3	0.9105	0.2661	0.9195	0.2432

4	0.9259	0.2204	0.9251	0.2286
5	0.9326	0.2042	0.9442	0.1893

2) Hyper Parameters: Epochs = 10, Batch Size = 32, LR = 0.001, Patience = 4

Table 5.8 CNN 2nd training history

No of Epochs	Accuracy in %	Loss (Training)	Accuracy of Validation	Loss of Validation
1	0.8081	0.6112	0.8958	0.3565
2	0.9199	0.2725	0.9055	0.2710
3	0.9320	0.2191	0.9476	0.1558
4	0.9484	0.1661	0.9566	0.1383
5	0.9487	0.1613	0.9627	0.1148
6	0.9580	0.1242	0.9619	0.1238
7	0.9596	0.1226	0.9556	0.1476

- Early stopping triggered here since the validation loss didn't improve for 3 consecutive epochs.

5.3 ViT

The no of samples in the dataset were 27500. Following ANN, CNN we used to ViT model for training. We used various parameters to fine tune the model (LR, BS, no of layers, patches). The history of our training including metrics (loss, accuracy) is given below:

Table 5.9 Samples for ViT model

Total No of samples	27500
No of training samples	19245
No of testing samples	4130
No validation samples	4125

1) Hyper Parameters: Epochs = 5, Batch Size = 32, LR = 0.001

Table 5.10 ViT 1st Training History

No of Epochs	Accuracy in %	Loss (Training)	Accuracy of Validation	Loss of Validation
1	0.2416	2.9713	0.3985	0.8086
2	0.3677	1.8420	0.5583	1.3146
3	0.5257	1.2794	0.6679	0.8382
4	0.6444	0.9483	0.7368	0.6354
5	0.7467	0.7220	0.8086	0.5497

2) Hyper Parameters: Epochs = 10, Batch Size = 32, LR = 0.001

Table 5.11 ViT 2nd Training History

No of Epochs	Accuracy in %	Loss (Training)	Accuracy of Validation	Loss of Validation
1	0.6845	1.0141	0.8354	0.5240
2	0.8076	0.5857	0.8788	0.4020
3	0.8347	0.4903	0.8987	0.3771
4	0.8700	0.4037	0.9072	0.2754

5	0.8957	0.2926	0.8567	0.4379
6	0.8899	0.3280	0.9132	0.2679
7	0.9225	0.2327	0.9404	0.1712
8	0.9210	0.2383	0.9383	0.1935
9	0.9451	0.1727	0.9521	0.1653
10	0.9269	0.2315	0.9388	0.2010

3) Hyper Parameters: Epochs = 20, Batch Size = 32, LR = 0.001

Table 5.12 ViT 3rd Training History

No of Epochs	Accuracy in %	Loss (Training)	Accuracy of Validation	Loss of Validation
1	0.9524	0.1607	0.9736	0.0833
2	0.9687	0.1075	0.9775	0.0771
3	0.9767	0.0741	0.9821	0.0556
4	0.9782	0.0752	0.9779	0.0686
5	0.9835	0.0553	0.9874	0.0439
6	0.9843	0.0502	0.9840	0.0546
7	0.9865	0.0518	0.9864	0.0504
8	0.9872	0.0386	0.9864	0.0453
9	0.9904	0.0353	0.9879	0.0457
10	0.9882	0.0369	0.9869	0.0461
11	0.9885	0.0370	0.9881	0.0441
12	0.9889	0.0382	0.9886	0.0445
13	0.9907	0.0321	0.9874	0.0452
14	0.9888	0.0359	0.9876	0.0449
15	0.9891	0.0353	0.9879	0.0449
16	0.9886	0.0370	0.9876	0.0446
17	0.9894	0.0370	0.9876	0.0458

18	0.9897	0.0357	0.9881	0.0438
19	0.9886	0.0366	0.9879	0.0447
20	0.9893	0.0369	0.9881	0.0451

4) Hyper Parameters: Epochs = 25, Batch Size = 32, LR = 0.001

Table 5.13 ViT model 4th Training History

No of Epochs	Accuracy in %	Loss (Training)	Accuracy of Validation	Loss of Validation
1	0.9498	0.1677	0.9733	0.0850
2	0.9677	0.1072	0.9770	0.0755
3	0.9763	0.0810	0.9808	0.0856
4	0.9804	0.0693	0.9813	0.0626
5	0.9827	0.0612	0.9850	0.0517
6	0.9848	0.0508	0.9821	0.0568
7	0.9856	0.0476	0.9850	0.0513
8	0.9868	0.0460	0.9886	0.0396
9	0.9866	0.0419	0.9864	0.0498
10	0.9881	0.0417	0.9879	0.0469
11	0.9878	0.0403	0.9872	0.0508
12	0.9888	0.0378	0.9884	0.0435
13	0.9900	0.0347	0.9888	0.0416
14	0.9892	0.0365	0.9886	0.0444
15	0.9890	0.0359	0.9881	0.0434
16	0.9905	0.0344	0.9886	0.0430
17	0.9886	0.0361	0.9881	0.0433
18	0.9895	0.0344	0.9884	0.0447
19	0.9897	0.0335	0.9884	0.0424

20	0.9888	0.0372	0.9884	0.0441
21	0.9903	0.0327	0.9884	0.0440
22	0.9891	0.0350	0.9884	0.0440

- Early stopping triggered validation accuracy stopped improving for a set of consecutive epochs.

5.4 Hybrid model (CNN &ViT)

The number of samples in the dataset were 27500. Following ANN, CNN, ViT we use hybrid model CNN and ViT for training. We have used various parameters to tune the model (LR, BS, no of layers, patches). The history of our training along with metrics (loss, accuracy) is below:

Table 5.14 Hybrid model Samples

Total No of samples	27500
No of training samples	19245
No of testing samples	4130
No validation samples	4125

1) Hyper Parameters: Epochs = 5, Batch Size = 32, LR = 0.0001

Table 5.15 Hybrid model 1st Training Samples

No of Epochs	Accuracy in %	Loss (Training)	Accuracy of Validation	Loss of Validation
1	0.7516	0.6462	0.8705	0.3608
2	0.8870	0.3306	0.9198	0.2276
3	0.9234	0.2273	0.9202	0.2154

4	0.9438	0.1648	0.9583	0.1214
5	0.9489	0.1514	0.9535	0.1679

2) Hyper Parameters: Epochs = 5, Batch Size = 32, LR = 0.0001

Table 5.16 Hybrid model 2nd Training Samples

No of Epochs	Accuracy in %	Loss (Training)	Accuracy of Validation	Loss of Validation
1	0.9596	0.1285	0.9486	0.0044
2	0.9602	0.1236	0.9639	0.0035
3	0.9692	0.0980	0.9716	0.0027
4	0.9752	0.0787	0.9697	0.0029
5	0.9744	0.0788	0.9789	0.0024
6	0.9766	0.0735	0.9782	0.0020
7	0.9790	0.0652	0.9731	0.0028
8	0.9782	0.0720	0.9850	0.0016
9	0.9811	0.0562	0.9833	0.0019
10	0.9835	0.0505	0.9714	0.0028

3) Hyper Parameters: Epochs = 15, Batch Size = 32, LR = 0.0001

Table 5.17 Hybrid model 3rd Training Samples

No of Epochs	Accuracy in %	Loss (Training)	Accuracy of Validation	Loss of Validation
1	0.9888	0.0345	0.9738	0.0032
2	0.9902	0.0315	0.9724	0.0038

3	0.9935	0.0196	0.9821	0.0018
4	0.9921	0.0254	0.9833	0.0020
5	0.9954	0.0151	0.9748	0.0032
6	0.9910	0.0315	0.9833	0.0019

- After epoch 6, as validation loss didn't get better for 3 consecutive epochs (epochs 4, 5, 6) so early stopping triggered.

4) Hyper Parameters: Epochs = 20, Batch Size = 32, LR = 0.0001

Table 5.18 Hybrid model 4th training history

No of Epochs	Accuracy in %	Loss (Training)	Accuracy of Validation	Loss of Validation
1	0.9456	0.1627	0.9808	0.0596
2	0.9651	0.1113	0.9821	0.0531
3	0.9678	0.1019	0.9830	0.0520
4	0.9740	0.0876	0.9842	0.0474
5	0.9733	0.0840	0.9845	0.0450
6	0.9756	0.0759	0.9838	0.0450
7	0.9783	0.0728	0.9850	0.0436
8	0.9790	0.0708	0.9855	0.0430
9	0.9807	0.0655	0.9857	0.0408
10	0.9803	0.0660	0.9857	0.0410
11	0.9782	0.0661	0.9862	0.0396
12	0.9815	0.0579	0.9869	0.0397
13	0.9800	0.0644	0.9867	0.0394
14	0.9803	0.0614	0.9862	0.0383
15	0.9807	0.0636	0.9869	0.0378
16	0.9819	0.0594	0.9872	0.0360
17	0.9821	0.0631	0.9879	0.0360

18	0.9811	0.0622	0.9884	0.0344
19	0.9822	0.0540	0.9881	0.0348
20	0.9823	0.0579	0.9884	0.0349

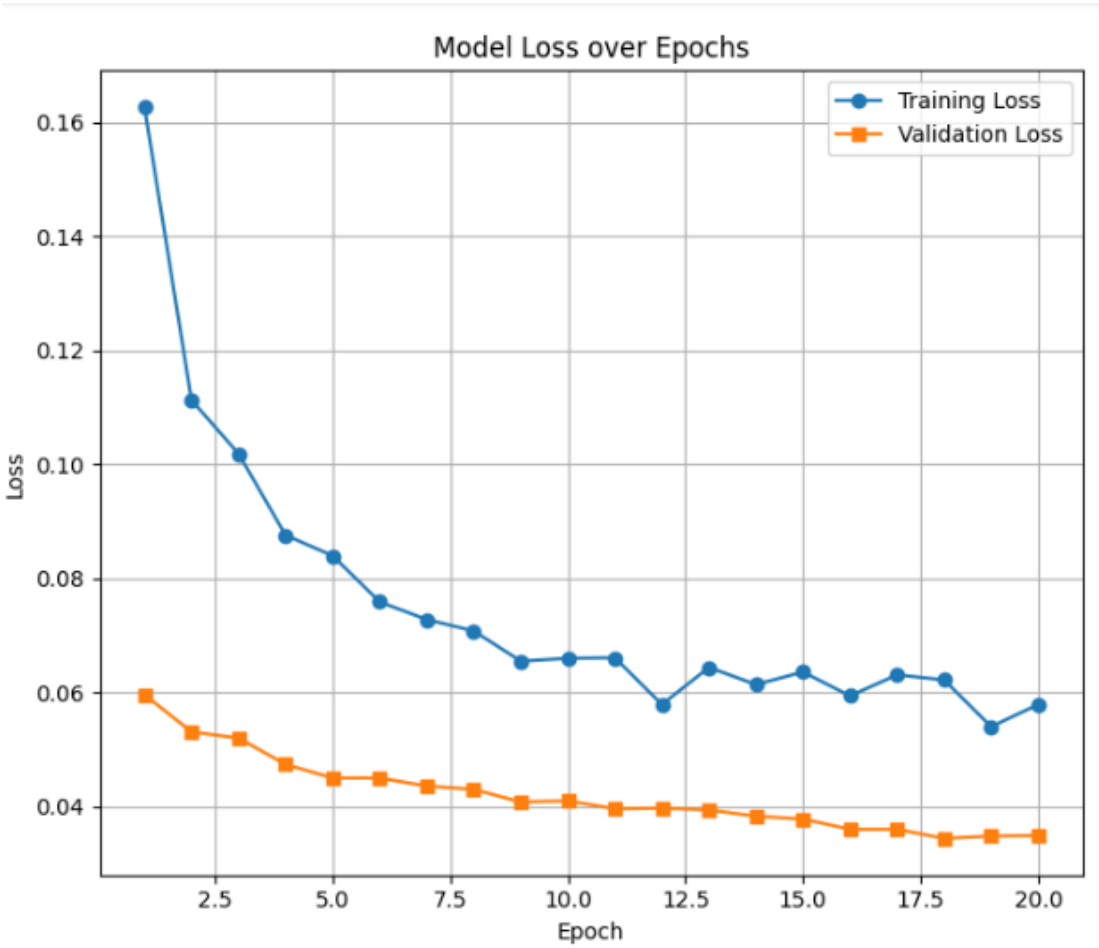

Figure 5.1 Training and Validation Loss

- Loss Graph: Training and validation loss decline across epochs, with validation loss consistently lower.

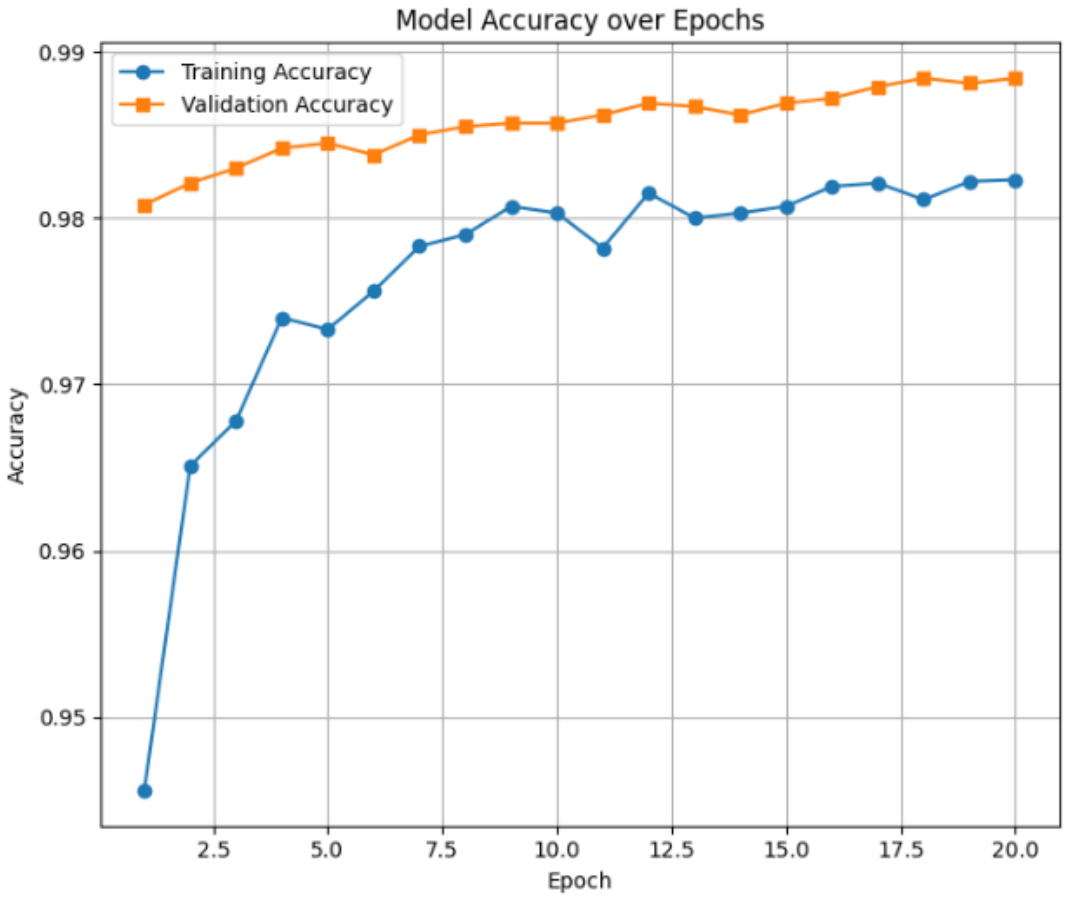

Figure 5.2 Training and Validation Accuracy

- Accuracy Graph: Both validation and training accuracy increase consistently, with validation accuracy very slightly better than training.

Table 5.19 Classification Report

Index	Precision	Recall	F1-Score	Support
0	0.9939	0.9818	0.9878	826
1	0.9891	0.9903	0.9897	826
2	0.9637	0.9952	0.9792	826
3	0.9928	0.9988	0.9958	826
4	0.9851	0.9576	0.9711	826

- The classification report displays good performance for all five classes. Class 3 records the best F1-score (0.9958), reflecting perfect precision and recall. Class 2 demonstrates lower precision compared to others but an exceptionally high recall. Generally, the model demonstrates strong and balanced classification accuracy.

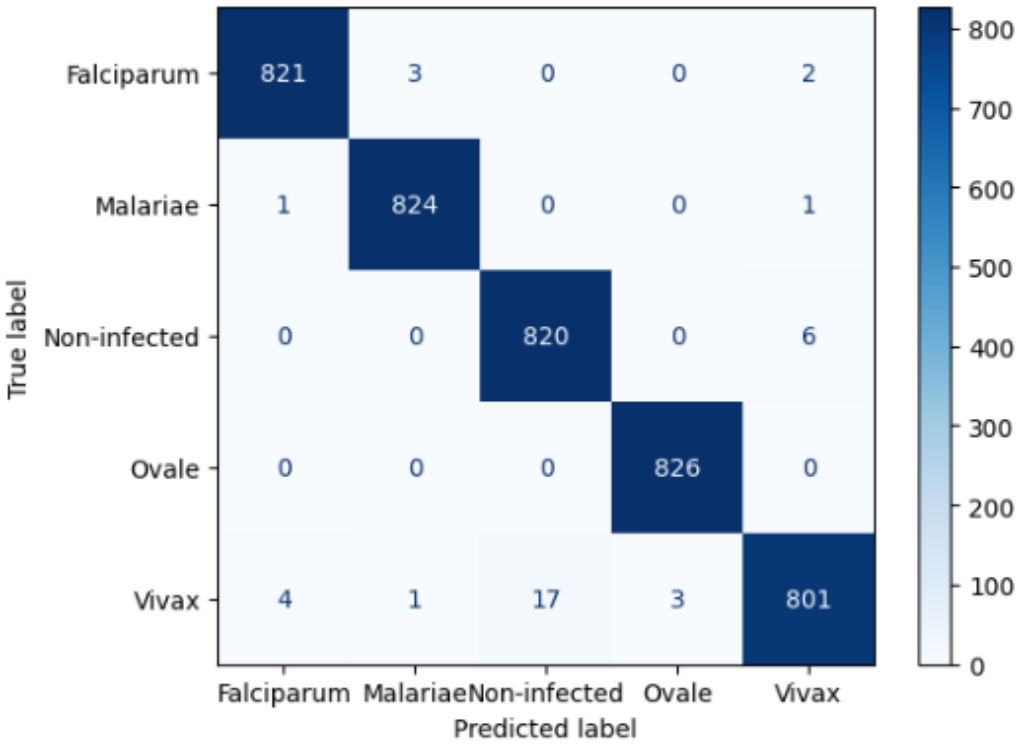

Figure 5.3 Confusion matrix

- The confusion matrix suggests high classification performance in all types of malaria and the non-infected class. The majority of predictions match the actual labels with little misclassifications. Highest confusion between Vivax and Non-infected samples but overall model performance is still good.

5.1 Training Overview of Hybrid Model (CNN & ViT Combined)

The model had excellent and stable learning curve over the 20 training epochs. Training accuracy improved from **94.56% to 98.23%,** and validation accuracy improved from **98.08% to 98.84%,** with outstanding generalization. Concurrently, training loss dropped from **0.1627 to 0.0579,** while validation loss dropped from **0.0596 to 0.0349**, indicating error prediction minimization successfully.

After training, the model has a test accuracy of **99.08%** and a very low-test loss of 0.0204 with good generalization on unseen data. This proves that the hybrid CNN-ViT architecture, which was trained from scratch, can perform accurate weapon detection through the provided dataset.

5.5 Deployment

Deployment was the process in our malaria detection application in which machine learning models were being incorporated into a responsive web application. The intention was to provide engaging user experience and accurate predictions as well as accuracy.

5.5.1 Application Architecture

The front-end for the application was created using React.js to look after user interface and back-end development for computing model inference. The use of React is a result of the component-based property enabling the user interface to be dynamic and scalable in nature. Front-end to back-end communication was done using RESTful APIs to push images and receive prediction output in response form.

5.5.2 Navigation Menu

Rather than a sidebar, our application utilizes a top navigation bar with the following sections:

Home: Short introduction and description of the system used for malaria detection.

Upload: Top page feature where patient information is entered, and images are uploaded to be classified.

Guide: User guide and FAQs on the best use of the app.

5.5.3 Page Details

Home Page: It's the first page and sets the context regarding the reason the Web was built to combat malaria through the power of automated testing also a form to enter patient details.

Upload Page: The interaction hub page where the user loads an image and gets live predictions from the hybrid CNN-ViT model.

Guide Page: Presents a quick guide to system use, i.e., file formats accepted, interpretation of results, and support contact information.

5.5.4 Model Combination

A ViT-CNN-based hybrid model was employed in the Flask backend. The model was trained on a balanced dataset of malaria cells at a 224×224 resolution. Image preprocessing is performed on image upload, and it is supplied to the model. The backend returns:

- Predicted class label

- Confidence score

5.2 User Guide

Here is the user guide to execute the malaria detection web:

Step 1: Execute the web in any web browser.
Step 2: Input a patient's name, age, and email address hit submit.
Step 3: Navigate to the Upload page from the top menu.
Step 4: Click the "Upload Image" button to upload a blood smear image.
Step 5: Click "Predict" to run classification.
Step 6: Click "Download Report" to download a PDF report of the result

Home Page:

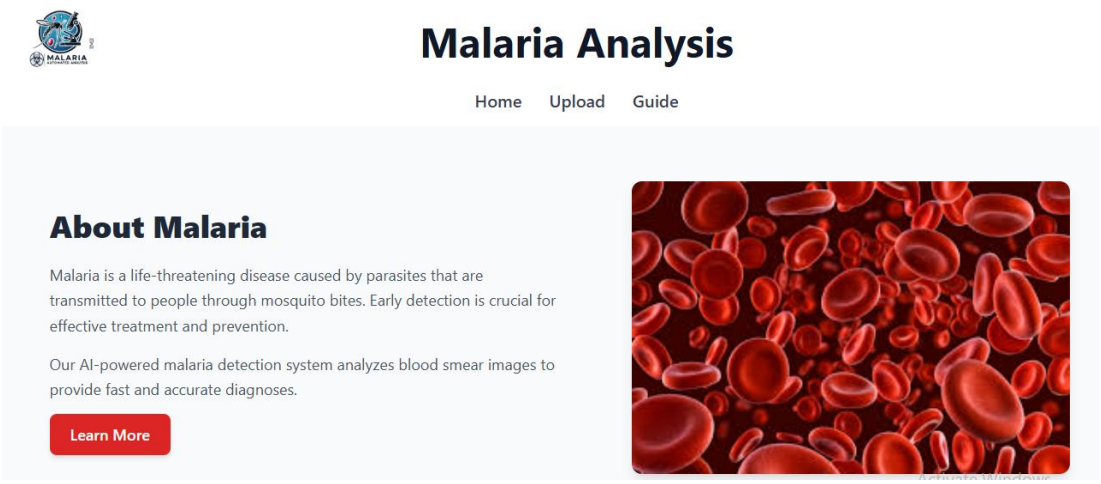

Figure 5.4 User manual Home Page

Upload page:

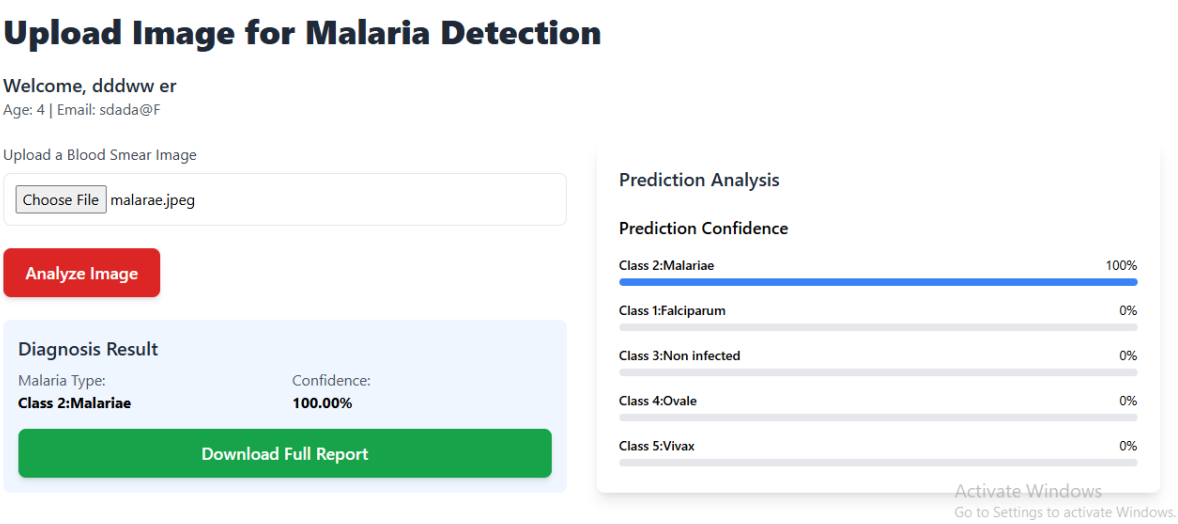

Figure 5.5 User manual Upload Page

CHAPTER 6

CONCLUSION AND RECOMMENDATIONS

6.1 Conclusion

Malaria Detection System project successfully created, developed, and implemented an automatic malaria diagnosis system, using images of blood samples. The system utilized sophisticated image processing methods and machine learning models [3] to obtain high accuracy in identifying malaria parasites, therefore offering a quick and efficient solution to the conventional manual microscopy. The simple-to-use web-based system enables medical professionals to upload images of blood samples and receive real-time diagnostic results, which makes the system an efficient system for use in malaria-endemic areas.

While successful, the system has the disadvantage of not being able to ensure data quality and availability of resources. Future development will involve an increase in what the system can do to enable it to detect other blood-borne diseases, work towards creating a smartphone app to facilitate ease of use and working in conjunction with health practitioners to test on a large scale. Malaria Detection System is a huge breakthrough in malaria detection and can enhance healthcare across the globe.

6.2 Recommendations

To improve the system to identify malaria, the highest priority is to optimize the AI model for a wide range of image qualities by the influence of state-of-the-art preprocessing methods and hybrid models and ensure strong performance in field deployments. The system should be deployed in a light and offline mobile app in low-resource settings. Clinical validation through multi-centre clinical trials needs to be conducted to meet WHO certification standards, and clinician feedback loops must be integrated to facilitate maximum usability. Public health program

partnerships will allow integration with current healthcare infrastructure to facilitate a sustainable model of implementation.

6.3 Future work

The capability of the system to identify co-infections such as dengue and typhoid through multi-disease AI models can greatly enhance differential diagnosis in high-endemic areas. Assessment of treatment efficacy with time and effects on the technology's impact on the healthcare process will be helpful to optimize. Research on integration of telemedicine and adaptive learning to allow on-going adaptation of models to varying regional data variation is promising areas. Finally, incorporation into national surveillance networks would make the system an asset to real-time surveillance of malaria as well as managing outbreaks.

REFERENCES

[1] Sawant, S., & Singh, A. (2024). Malaria Cell Detection Using Deep Neural Networks. *arXiv preprint arXiv:2406.20005*

[2] Dong, Y., Jiang, Z., Shen, H., & Wang, S. (2020). Classification of malaria-infected cells with deep convolutional neural networks. *PeerJ*, 8, e10217.

[3] Rajaraman, S., Antani, S. K., Poostchi, M., Silamut, K., Hossain, M. A., Maude, R. J., ... & Thoma, G. R. (2018). Pre-trained convolutional neural networks as feature extractors toward improved malaria parasite detection in thin blood smear images. *PeerJ*, *6*, e4568.

[4] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, *25*.

[5] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... & Houlsby, N. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.