**Malaria Screener: AI-Powered Malaria Diagnosis System with Mobile, Web, and Chatbot Integration**

**Introduction**

**Background & Problem Statement**

Malaria remains one of the deadliest diseases globally, especially in Sub-Saharan Africa, with children and pregnant women being the most vulnerable. According to the World Health Organization (WHO), over 240 million cases of malaria are reported annually, resulting in nearly 600,000 deaths. Timely and accurate diagnosis is critical for effective treatment, but many remote or low-resource areas lack trained personnel and diagnostic tools.

To bridge this gap, our team developed **Malaria Screener**, an **AI-powered diagnostic system** integrated with **mobile apps, a web dashboard, and a chatbot** to support health workers and communities in identifying malaria cases efficiently and reliably.

This project contributes directly to **Sustainable Development Goal 3 (Good Health and Well-being)** and **SDG 9 (Industry, Innovation and Infrastructure)** by harnessing the power of artificial intelligence and digital technologies to support accessible healthcare.

A person holding a cellphone

AI-generated content may be incorrect.

**Purpose of the Blog**

This blog aims to share the development journey of Malaria Screener -from idea to implementation and inspire others in the field of digital health innovation. We’ll walk you through the challenges, methodology, results, and future possibilities.

**The Problem: Lack of Reliable and Scalable Diagnosis**

In many health posts across rural Africa and Asia, diagnosis is still done manually through microscopy by untrained eyes, which often leads to misdiagnosis. Our team witnessed real cases where patients were either underdiagnosed or over-treated, both of which can be fatal or cause drug resistance. We aimed to solve this with a scalable AI solution.

**Methodology**

**Dataset**

We used the **NIH Malaria Dataset** and additional annotated RBC microscopy images collected from clinical partners. The dataset included both **thin and thick smear images** of infected and uninfected cells.

**Pre-trained CNN Models**

To ensure efficiency on mobile devices, we experimented with:

* **MobileNetV2 (TensorFlow Lite optimized)**
* **EfficientNet Lite**
* **ResNet50 (quantized for mobile)**
* **YOLOv5 (for future detection/segmentation tasks)**

Final Model: **MobileNetV2** for its lightweight size and acceptable trade-off between accuracy and latency.

**Preprocessing & Optimization**

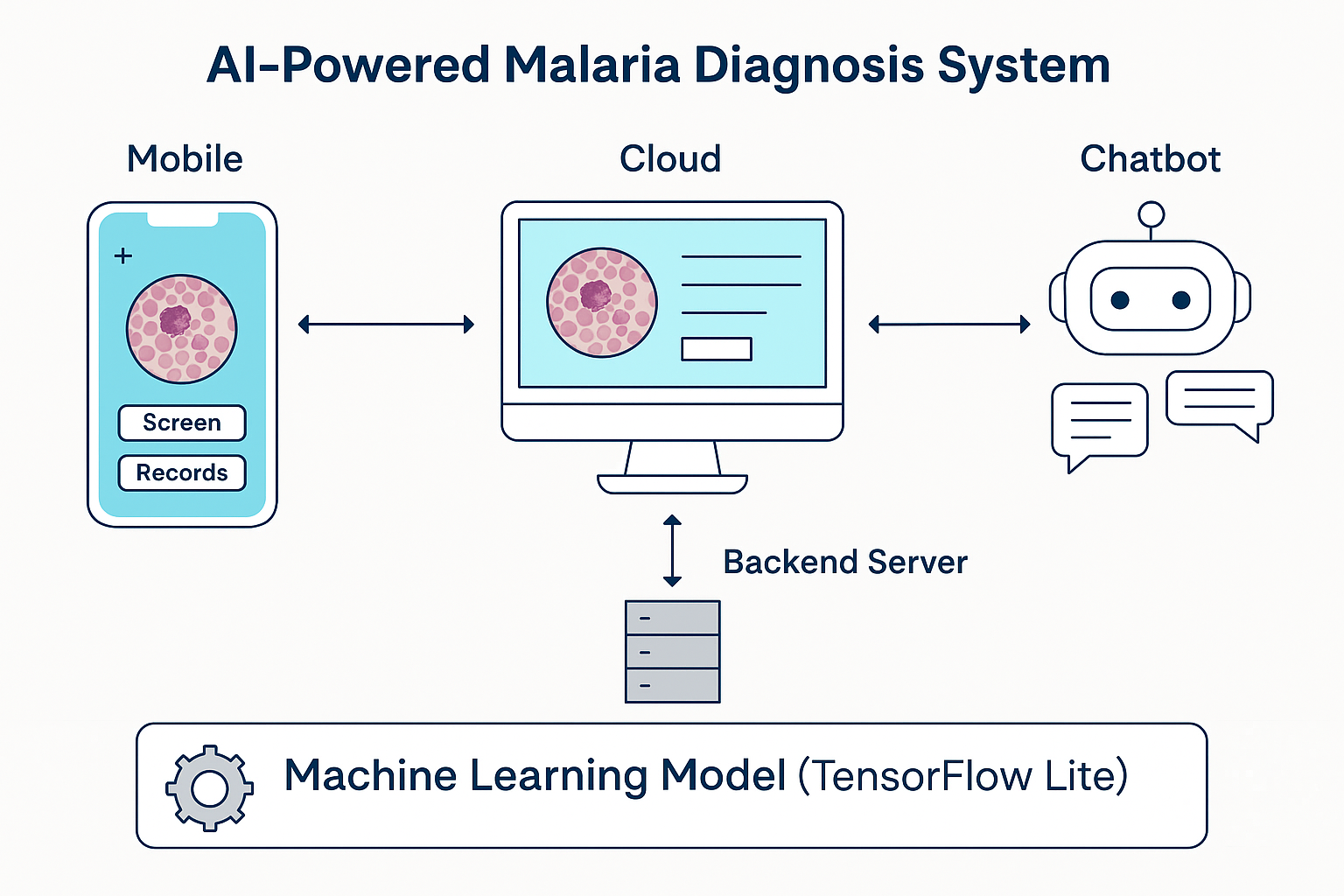
* **Image Normalization**
* **Data Augmentation** (rotation, flipping, zoom)
* **TFLite Model Quantization** for on-device inference
* Training done on Google Colab Pro with early stopping and validation checks

**Implementation**

**Architecture Overview**

**Three Interfaces**:

1. **Mobile App** (Flutter): Camera-based diagnosis, patient registration, results storage
2. **Web Dashboard**: For clinicians to view history, analyze reports
3. **Chatbot**: Simple chatbot integration to ask questions or view results



**Postgres Database Schema (on-device)**

* Patient: ID, Gender, Age, Initials
* Slide: Date, Operator, Site
* Image/ImageThick: Infected counts, GT values

**Example Workflow**

1. User captures a slide image via the app.
2. The model processes and classifies the image as infected or not.
3. Results are stored in Postgres and optionally uploaded to the cloud (future scope).
4. Users can query results via web or chatbot.

**Results & Findings**

|  |  |
| --- | --- |
| **Metric** | **Value** |
| Accuracy (val) | 94.3% |
| Model Size | 3.4 MB (TFLite) |
| Inference Time | ~140ms/image |
| Platforms Tested | Android, Web, Telegram Bot |

Visualization:  
Confusion matrix, ROC curve, and segmentation overlay (to be added in the final web version).

A blue squares with white text

AI-generated content may be incorrect.

**Key Reflections**

* **Challenges**: Lighting inconsistencies during image capture were a major challenge.
* **Limitations**: Current model doesn’t support multi-species classification (e.g., P. vivax vs. P. falciparum).
* **Future Scope**:
  + Real-time cloud sync for centralized data
  + Model retraining on user feedback (active learning)
  + Support for other tropical diseases using similar pipelines

**Conclusion**

We built **Malaria Screener** as a fully integrated AI-powered solution targeting malaria diagnosis through mobile, web, and chatbot interfaces. Our aim was to **democratize access to malaria testing** through accessible, low-cost technology — and we believe we’ve taken a solid first step.

**Recommendations**

* Ministries of Health and NGOs could scale such tools across community health workers.
* Developers should consider edge AI optimization for healthcare apps.
* Future research should explore integrating blood count analysis and parasite species detection.

**References**

1. Chittagong Medical College Hospital. (2020). *De-identified patient records*. IRB#129
2. Google Health. (2022). *AI for Malaria Diagnosis*. <https://ai.google/social-good/health/>
3. NIH Malaria Dataset. *National Institutes of Health*. <https://lhncbc.nlm.nih.gov/publication/pub9932>
4. Palanica et al. (2019). *AI Chatbots in Healthcare: A Systematic Review*. JMIR.
5. Rosado et al. (2017). *Mobile-Based Analysis of Malaria-Infected Blood Smears*. Sensors.
6. TensorFlow Team. (2023). *On-Device ML with TF Lite*. <https://www.tensorflow.org/lite>
7. NIH Malaria Dataset: <https://lhncbc.nlm.nih.gov/LHC-publications/pubs/MalariaDatasets.html>
8. WHO. (2021). *Digital Health Interventions*. <https://www.who.int/health-topics/digital-health>
9. WHO. (2021). *Malaria Treatment Guidelines*, 3rd Edition.
10. Yang et al. (2019). *Deep Learning for Smartphone-Based Malaria Parasite Detection*. IEEE JBHI.
11. Yu et al. (2020). *Malaria Screener: A Smartphone Application for Automated Malaria Screening*. BMC Infectious Diseases.