Technical Documentation for MboaLab App for typhoid screening:

Rachana Sooraj

MboaLab aims to improve diagnostics of Typhoid through Open Science which is an AI based technique. The idea is to utilize the high-resolution camera and powerful computing power of modern smartphones to screen blood smear images automatically for parasites such as Salmonella Typhi and infected red blood cells.

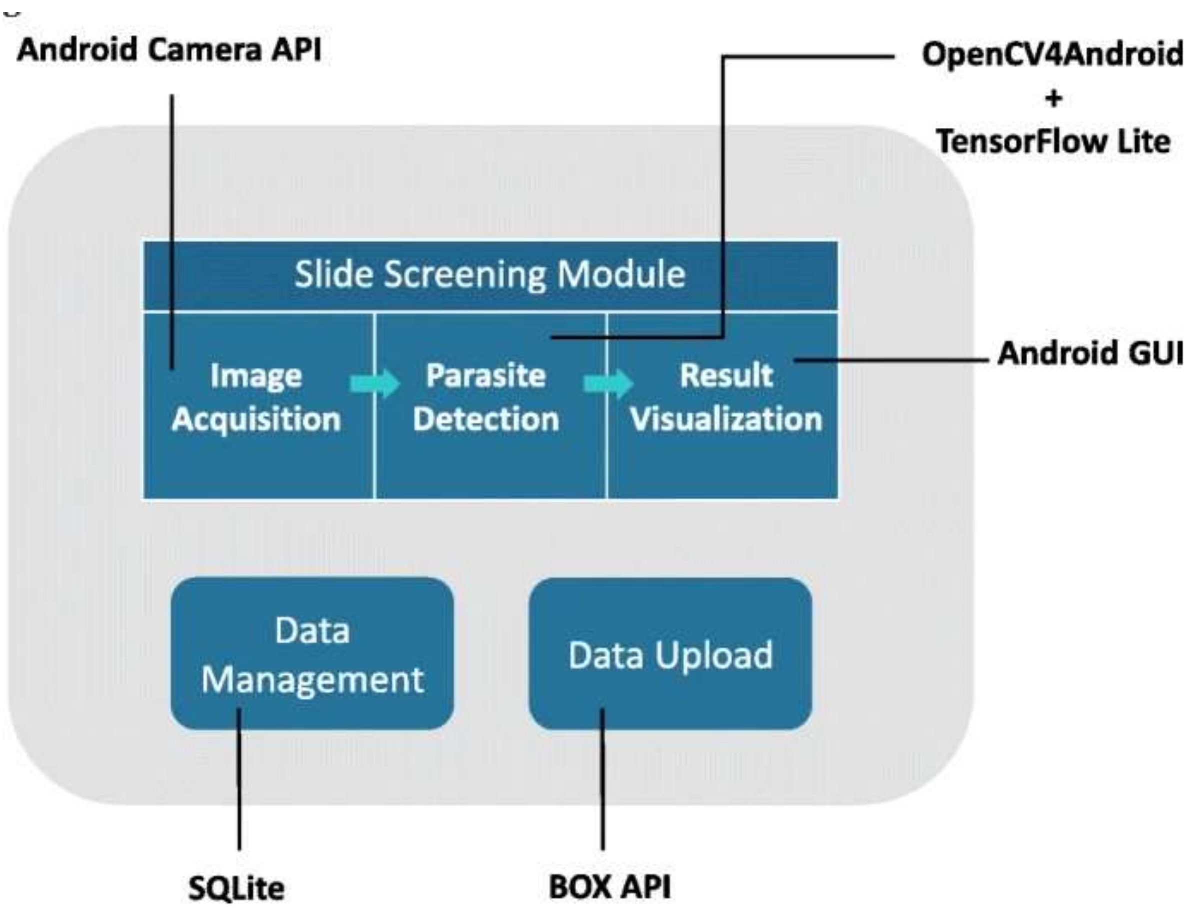
This app will require image acquisition, smear image analysis, and result visualization in its slide screening process. It will also be equipped with a database to provide easy access to the acquired data.

We will need the following software to implement this:

1. Android Studio
2. Tensorflow Lite
3. Tensorflow and Keras.
4. Sqlite

Working of the typhoid screener app:

The below diagram shows the different API that is required to run the typhoid detection model. A diagram of the application's architecture is shown in Fig.It has been referred from the Malaria Screener.



Slide Screening Module:

This module consists of three parts:

1. Image Acquisition
2. Parasite Detection
3. Result Visualisation

1.**Image Acquisition**:
In image acquisition, a user will use camera mounted on top of a microscope to take photo of the microculture.

- ***Implementation:***
  To acquire images, a customised camera function can be implemented using **Android Camera API**. This includes a Camera object that controls the intrinsic parameters of the camera hardware, and a CameraPreview object that displays the preview image to the user. During a screening session, the user presses a button to capture the image when a suitable field of view becomes visible.

2. Parasite Detection:

The captured image is then passed to the parasite detection module as input. The app can examine both thin and thick smears with potential infections. For a single thin smear image, the goal is to detect the number of infected red blood cells (RBCs) and the total number of RBCs in the image. For a thick smear image, the goal is to detect the number of parasites and white blood cells (WBCs). The parasite detection module has a *ThinSmearProcessor* class and *ThickSmearProcessor* class to handle each of the two scenarios.

- ***Implementation:***
  The acquired images will undergo data enhancement techniques such as **CLAHE(Contrast limited adaptive histogram equalization)** to improve the quality of images and assist the deep learning model in detecting parasites easily. These enhanced images then undergo modelling with the help of **Convolutional Neural Network (CNN)** .For modelling we can also use transfer learning methods such as **Densenet**. Both classes use these models to make classifications: infected vs uninfected RBC in the case of a thin smear, or parasite vs background in the case of a thick smear. The images need to be classified as Typhoid negative, Typhoid Positive-Stage1, Typhoid Positive-Stage2, Typhoid Positive-Stage3, Typhoid Positive-Stage4. This can be done by calculating the total number of parasites/infected cells.The CNN models are pre-trained on a PC with **TensorFlow** and Keras, which outputs the weights of the trained models in HDF5 (.h5) format. Next, the models are converted to Protocol Buffers (.pb) format and deployed to the app using **TensorFlow Lite**.

3. **Result Visualisation:**
   This class generates a down-sampled version of the captured image with labels drawn on the infected RBCs (parasites for thick smear images). In addition, it uses a table to show the numerical results.
   - ***Implementation:***

   The result visualization module uses *ResultDisplayerActivity,* a UI class that was implemented to present the detection results to the user.

Validation using Google Forms survey:

After the result visualisation, the symptom survey form pops up. This can again detect and classify the disease severity. The form acts as an decision support system and helps in validating the results of the typhoid screener from microculture images.

Data management module

The data management module stores the images and the corresponding metadata acquired during slide screening sessions. Finally, the data upload module transfers the local data to an online repository for record-keeping and further training of the system. The data can then be stored in a SQLite database.

The data management module also offers a UI to let the user browse the images and metadata stored in the SQLite database.

Data upload module

Images and metadata in the database can be exported and uploaded to an online repository. The uploaded data can be used to examine the app performance, and to improve the classifier of the parasite detection module with additional training.To start an upload event from the database UI, select 'Go To Database' option in the landing page.With this option, the app will scan for all data that has not been uploaded yet, which will then be uploaded.

- Implementation:

  The back-end of this module is implemented with a mixture of both Android and Box API. Box API is used to implement functions to execute upload tasks to a Box repository.

Steps required in building the typhoid detection app:

Step 1:

Prepare the dataset. This will include getting huge amount of images showing microcultures of the parasite.The images need to be classified as Typhoid negative, Typhoid Positive-Stage1, Typhoid Positive-Stage2, Typhoid Positive-Stage3, Typhoid Positive-Stage4. Each of these should contain sufficient number of images to get highly accurate model. Since one of the issue faced by Mboalab is the lack of images, we could try different methods like image augmentation techniques on existing images such as Rotation_range, Zoom_range etc to present it as different images to the model.

Step 2:

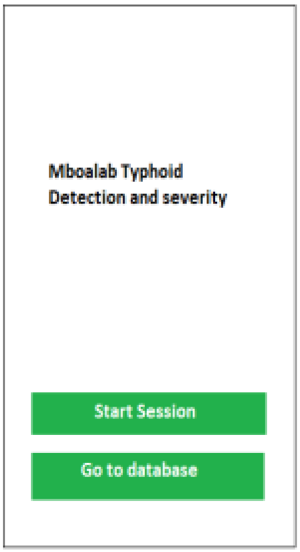
Train the model on the images. This could be done on Tensorflow, Keras etc.

Step 3:

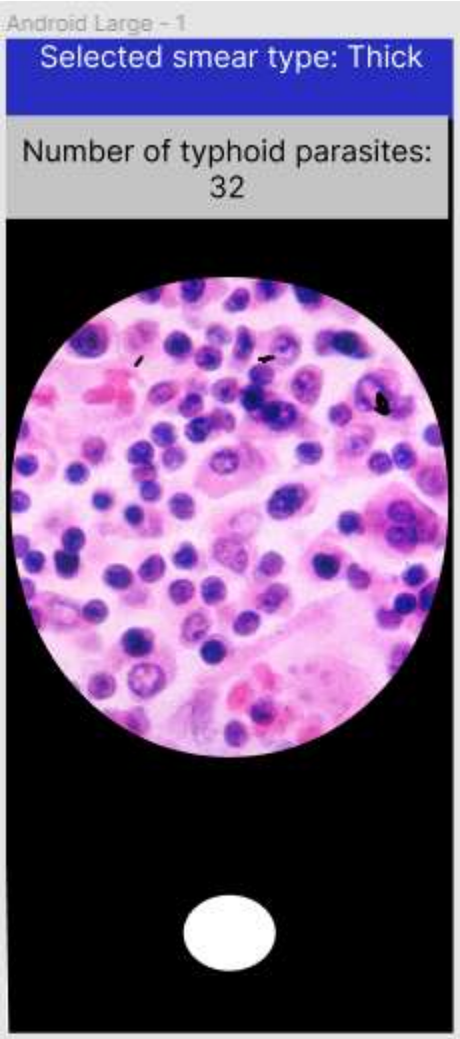
Deploy the model in android using TensorflowLite.

A prototype of how the App looks like: This has been created using Figma.

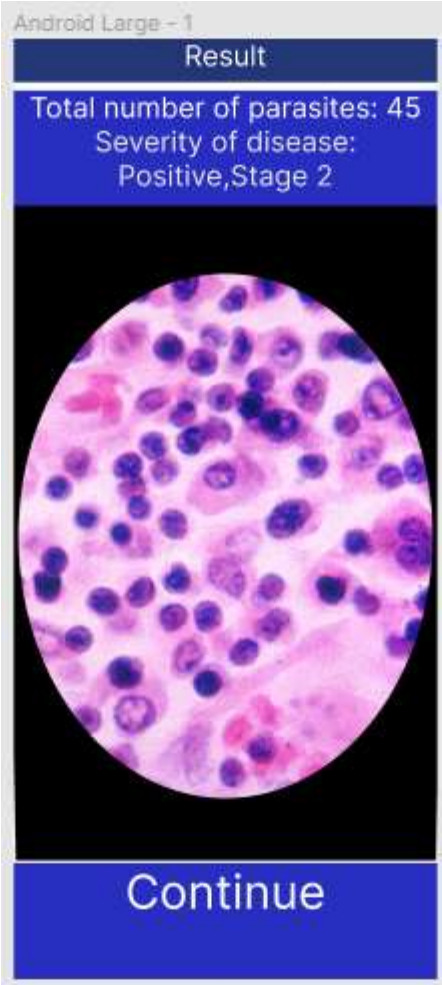
Page 1: The landing page where the user can select if they wish to upload data from database or open the camera to capture images of the microculture.



Page 2: This is the page that opens up when user selects ‘Start session’. It opens up the camera for the user to take images. Based on the image captured(Thick or thin smear), the Machine Learning model, identifies the parasite or infected RBC in the culture.



Page 3: The page displays the result of the scan. It shows the number of parasites and based on the number of parasites, classifies the disease as stage 1,stage 2 or stage 3.



Page 4: The next page shows the validation of result that can be done using the developed data collection tool such as google form. On entering the details of the form, the form can help in classifying the severity of the disease based on the number of symptoms.

This in addition to the ML classification will help in establishing the detection as well as severity of the disease.

Complete symptoms survey

Section 1 of 4

Typhoid Detection and Severity

This form is designed to collect clinical data about typhoid. This form will he
first or second or third stage of disease based on symptoms as well as numb
disease.

Name

Short answer text

Age

Short answer text

Weight

Short answer text

Next