Machine Learning Engineer Nanodegree

Capstone Proposal

Musa Mikail

July 2019

Domain Background

Malaria is a tropical infectious disease caused by the Plasmodium parasite. The World Health Organization (WHO[1]) reported that in 2017 alone, about 53.7 Million cases of malaria were reported in Nigeria with about 79,800 deaths reported as result of the disease. Diagnosis is of the disease if often made by observing and counting the number of infected blood cells under the microscope. However, the accuracy of diagnosis depends largely on the judgement of a human observer to classify and count infected and health cells. This become more challenging in northern Nigeria due to scarcity of competent laboratory personnel. Sivaramakrishnan Rajaraman., et al[2] has developed a Convolution Neural Network (CNN) that facilitates malaria parasite detection from microscopic images.

In this project, a binary classifier will be trained and deployed through a user friendly windows application that could improve efficiency and quality of plasmodia

Problem Statement

The goal of this project is to develop a windows PC application (to be called *i*-mal) that will simplify and improve the effectiveness of malaria diagnosis especially in rural parts of northern Nigeria.

The steps to follow towards achieving this goal are as follows:

1. Retrieve, explore and pre-process a relevant training dataset
2. Segment and count individual red blood cells using edge detection
3. Train a binary classifier to count number of infected red blood cells
4. Freeze and Deploy the classifier using a pyQt PC application
5. Analyse images of test slides and make inference
6. Compare performance of the deployed application with manual diagnostic techniques

[1] World Health Organization. World Malaria Report 2014 (World Health Organization, 2014)

[2] Sivaramakrishnan Rajaraman, Stefan Jaeger, Sameer K. Antani. "Performance evaluation of deep neural ensembles toward malaria parasite detection in thin-blood smear images."

Datasets and Inputs

The model will be trained using a dataset published by the United States National Library of Medicine[3] (. It includes images of segmented cells taken from thin blood smear slides of both infected and uninfected patients. The dataset comprises of 27,558 PNG images of the segmented cell partitioned into two (2) separate folders labelled “parasitized” and “uninfected”, each representing the two possible labels of either been infected or not. The dataset is balanced with each class containing 13,780 images.

For the purpose of this work, 75% of the labelled training data will be used for model training while 25% will be reserved for testing/validation.

Solution Statement

Design and implementation of i-mal will follow best practices in software engineering. Implementation of the Binary classifier will begin with exploration and pre-processing of the downloaded dataset. The data exploration operation will seek to explore additional insight about the available data like its dimension, distribution and other statistical features. The data will then move to the pre-processing stage where it is augmented through scaling, rotation and shifting operations to make it representative of the wide variations of cell images obtainable during inference. Additional, 2D convolution and Max pooling filters will also be added for dimensionality reduction.

The Cell classification model will be a Binary Image classifier based on a Convolution Neural Network (CNN). Three different CNN architectures namely the VGGNet (based on the work of Simonyan et al[4]) . The binary model will be benchmarked against a simple multi-layer perceptron prior to deployment.

The total cell count will be evaluated using the Canny edge detector method of the Open CV framework[5]. Images of a microscopic slides will first be converted into a grey image before the edge detection operation is applied.

The deployed application (i-mal) will take in as input, the image of a slide from a thin smear test and output the percentage parasitemia (%P), a parameter that gives the percentage of red blood cells infected by the plasmodium parasite.

---

[3] Jaeger, Stefan. “Malaria Datasets.” U.S National Library of Medicine, U.S Department of Health and Human Services, July. 18, 2019, https://ceb.nlm.nih.gov/repositories/malaria-datasets/

[4] Simonyan∗ & Andrew Zisserman “Very Deep Convolution Networks for Large Scale Image Recognition.”, ICLR. 2015

[5] “Canny Edge Detector.” https://docs.opencv.org/2.4/doc/tutorials/imgproc/imgtrans/canny_detector/canny_detector.html

To achieve this, the following (2) key parameters that will be evaluated by the model.

1. Total number of cell counts observed from the thin smear test (T)
2. Number of Red blood cells infected by the malaria parasite (Ti)

Therefore,

$$\%P = \frac{T_i}{T} \times 100$$

Benchmark Model

In this project, the accuracy of a vanilla ANN without convolution or maxpoolling and will be used to benchmark the performance of our deployed model after tuning. The selection of accuracy as the evaluation metrics is premised on the fact that the dataset ids balanced.

Evaluation Metrics

The Accuracy of *i*-mal will be the combined accuracy of the Canny edge detection estimator (for counting the total number of cells), and the accuracy of the cell binary classifier. Because the available malaria dataset is balanced, accuracy of the binary classifier against the validation data will be used as a metric for evaluating its performance.

Given,

A_c = Accuracy of the Canny Edge detection estimator
A_b = Accuracy of the binary cell classifier

Then,

$$A_c = \frac{Cell\ count\ from\ Canny\ edge\ detector - Cells\ from\ manual\ count}{Cells\ from\ manual\ count} \times 100$$

$$A_b = \frac{Cells\ correctly\ classified}{Size\ of\ Dataset} \times 100$$

$$A = A_c \times A_b$$

where,

A = Overall Accuracy of the System

Project Design

The system implementation will follow the methodology shown in figure 1.

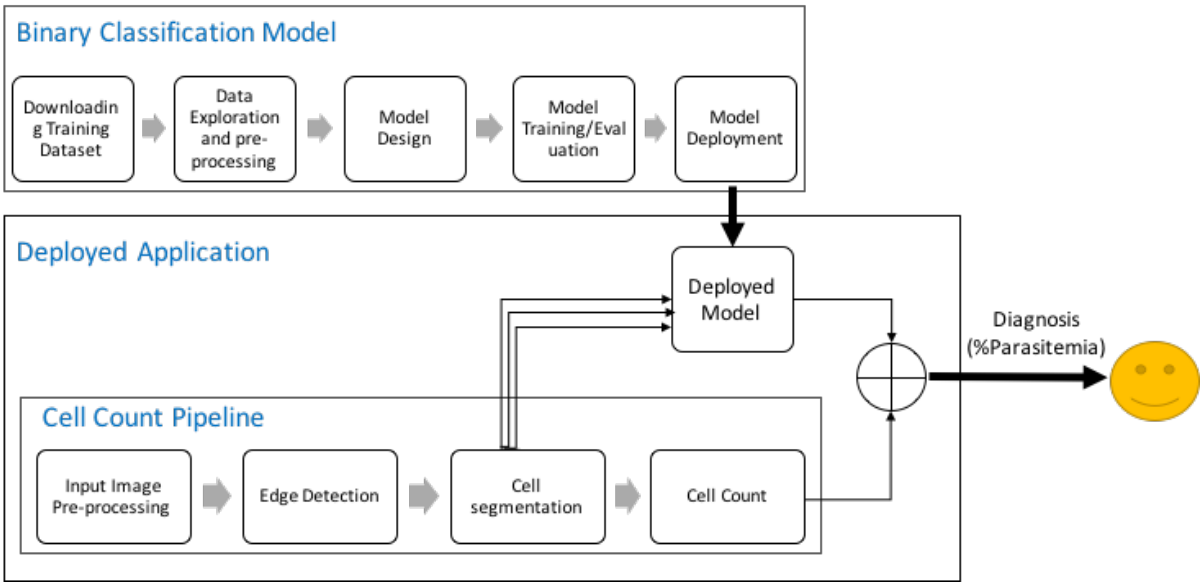

Figure 1 Methodology for Implementing the *i*-mal Application

Binary Classifier Design

- The malaria dataset from the United States National Library of Medicine will be used.
- Training images will be analysed to identify features and resized for uniformity before passing them into the CNN.
- The model will be implemented as a Convolution Neural Network (CNN) in Tensorflow and keras. Model Hyperparameters will be selected to achieve accuracy > 95%.
- Training will be done on Amazon SageMaker and model artefacts will be frozen and downloaded for Deployment.

Deployed Application

- The key components of the Deployed Application will be the trained model and the red blood cells counter.
- The red blood cells counter will be implemented using Open CV with the the in-built Canny edge detection library. Accuracy of the red blood cells countered will be evaluated separately from the Binary Classifier.
- A Graphical User Interface (GUI) will be provided for user to upload, visualize and make inference from available images, as well as inputting relevant patient data.