3.1 Overview

The goal of this work is to develop a system for diagnosing malaria using microscopic images of stained blood samples. This being an image recognition and classification task, a systematic sequence of events was followed to achieve the objective. Generally, the procedure followed in solving such a problem is as follows.

First an image is acquired and pre-processed, it is then segmented into different regions and appropriate features extracted.

Next, a suitable classifier is used to categorize the features into their different classes.

Finally, a decision is made about the information conveyed by the image based on the classes of features found by the classifier.

A similar criterion is used in this study to develop an algorithm for detection and quantification of Plasmodium parasites. Thin blood smear images were acquired from Centre for Disease Control (CDC) website [34] and captured from the Reference Laboratory of Malaria, in Sudan Ministry of Health. Images from CDC are posted to the website for either confirmation of diagnosis or archiving from laboratories all over the world. The Plasmodium life stages and species are specified for each image obtained from the website. These images are of different visual quality i.e. the images vary in their intensity contrast, hue, and magnification. This is a consequence of different techniques used in sample preparation, image capturing and processing.

Test results of the malaria diagnosis system developed in this chapter are presented and discussed in chapter 4. The performance of this system is compared to the results given by CDC. A discussion of the main findings, strengths and limitations of this work is also given.

3.2 The Process Model

Here, a process model for malaria diagnosis was developed. This model was supposed to take thin blood smear images as its input and give correct malaria diagnosis. Figure 3-1 shows the black box model of the system.

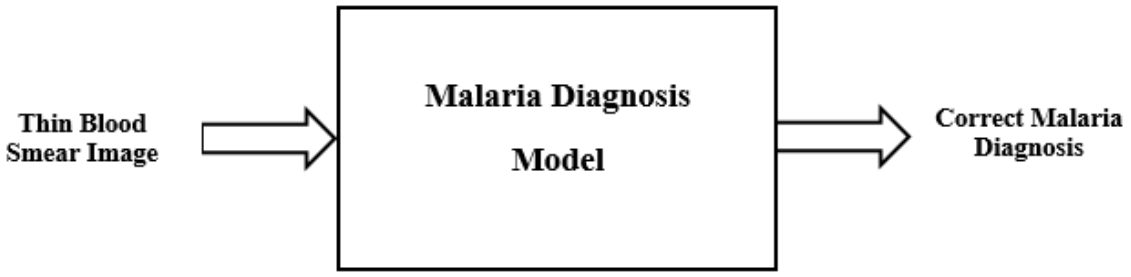

Figure 3-1: Black box model of malaria diagnosis system

The above model is implemented using six main processes, namely; image acquisition, image preprocessing, image segmentation, feature extraction, comparison and classification as shown in Figure 3-2.

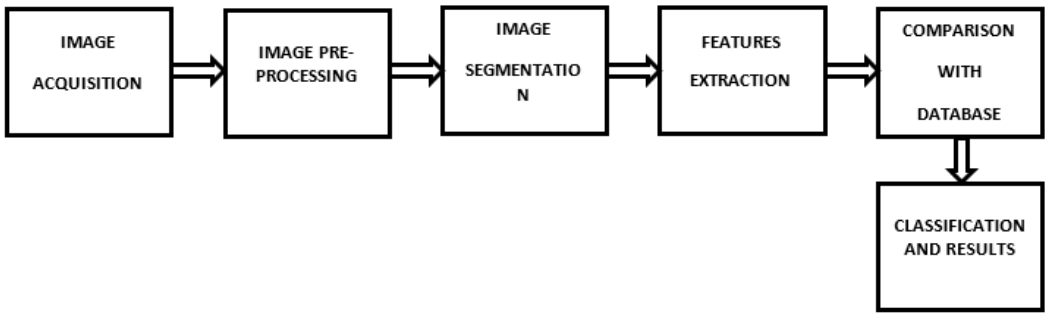

Figure 3-2: Block diagram of the malaria diagnosis system

Image acquisition denotes capturing of thin blood smear images using a digital camera mounted in the optical path of the microscope. After images are captured, they are loaded to a computer where they are processed. Processing involves the following stages: Image resizing, noise reduction, segmentation of RBCs and feature extraction. Finally, trained classifiers use the extracted features to determine whether the image is infected or not.

Every step in the algorithm involves rigid consideration in order to achieve an efficient and reliable malaria diagnosis system. Moreover, decisions on the best features to choose in designing the malaria diagnosis system are then made based on the test results obtained.

3.2.1 Image acquisition

Thin blood smear images were acquired from Centre for Disease Control (CDC) website [34] and captured from the Reference Laboratory of Malaria, in Sudan Ministry of Health.

3.2.2 Image pre-processing

The goal of this step is to make the acquired images more suitable for subsequent processes - mainly image segmentation and feature extraction. Basically, there are three main objectives for image pre-processing [35]. One is to resize the image for the purposes of either magnifying the image through digital zooming, or reducing the image size in order to speed up processing. The second objective of image pre-processing is to reduce or eliminate noise from the acquired image. Third objective is to enhance the image contrast for visual evaluation.

In this work, digital zooming and contrast enhancement are not necessary since the task of image classification and recognition is to be performed by a computer and not a human operator. However, size reduction is necessary in order to have all images with uniform size proceeding to the next stage. All images are rescaled to have the same size using the built in MATLAB™ function `imresize`. Since all CDC images have pixel resolution of (300 × 300 pixels) they do not need to be rescaled. The size of CDC images is adopted to be the standard image size of the pre-processed images.

Noise reduction is also considered to reduce some undesirable effects in the images, which often are acquired during the process of sample preparation and image acquisition such a non-uniform illumination, salt and pepper noise and image blurring.

Both captured images and CDC images are converted from RGB to gray scale to reduce the processing time. RGB to gray conversion is done by using the built in MATLAB™ function `rgb2gray` which converts RGB images to grayscale by eliminating the hue and saturation information while retaining the luminance [36], [37].

Filtering operation using a square median filter is performed to images. This operation served to remove spurious noise present in the images. Some of the possible sources of such noise include unbalanced illumination of the sample in the microscope, poor sample preparation, sample degradation or a combination of these factors [38].

using the built in MATLAB™ function `medfilt2` The length of the median filter used is 7 by 7, a value obtained from [39].

These steps have been implemented in MATLAB™ in Appendix A.

3.2.3 Image segmentation

Image segmentation is the fundamental step to analyze images and extract data from them. Image segmentation is a mid-level processing technique used to analyze images and can be defined as a processing technique used to classify or cluster an image into several disjoint parts by grouping the pixels to form a region of homogeneity based on the pixel characteristics like gray level, color, texture, intensity and other features [40], [41]. The purpose of

the segmentation process is to get more information about the regions of interest in an image, which helps in annotation of the object scene. The main goal of segmentation is to clearly differentiate between the object and the background in an image.

There are two objectives for image segmentation. One is to isolate the RBCs from the background and the second is to extract all the RBCs and process them individually in order to facilitate the process of feature extraction. The `m_segmentation` MATLAB™ user defined function developed to perform segmentation process in Appendix A.

3.2.3.1 Isolate RBCs from the background and edge detection

Isolating RBCs from the background involves the following steps: first, the gray threshold is calculated and the gray image is converted to binary image. The resulted binary image is enhanced and refined by removing small objects and filling the holes within the RBCs. Next, the refined binary image is segmented to a number of objects with each object represents an RBC or a number of overlapped RBCs.

In this work, edge detection is used for specify the boundary of each object in addition to announcing the presents of overlapped RBCs. Here a technique known as SUSAN (Small unvalued segment assimilating nucleus) is used for edge detection. The MATLAB™ code for SUSAN is in Appendix A.

3.2.3.2 RBCs extraction

In order to facilitate the process of feature extraction and reduce the computation cost, each segmented object (i.e. RBCs) is extracted and treated individually. This is achieved by applying a suitable mask for the certain object. The mask need three inputs: the labeled image, the grayscale image and the number of objects in the image. To extract a certain object from the

image, all the pixels in the labeled image are zeroed out except the pixels within that object boundary, which replaced with the Corresponding values from the grayscale image. The MATLAB^{TM} code for RBCs Extraction is in Appendix A.

In summary, the segmentation process separates the objects of interest from the background and from each other and defines the zone of measurement, i.e. the region where to measure the characteristics of the object. The objects of interest are in this case red blood cells, which are either infected or not by the plasmodium parasite. The zone of measurement is the area of the whole single cell.

3.2.4 Feature extraction

In order to distinguish between infected and non-infected red blood cells, we need to extract features from the image array and compute new variables that concentrate information to separate classes. The set of features should discriminate between infected and non-infected RBCs as well as possible. An additional requirement is robustness, so that the results can be reproduced for new independently collected material.

Raw images cannot be used directly as features due to high variations in morphology which are coupled with arbitrary rotations and scales and because the raw images contain large amount of data, but relatively little information. This is the aim of feature extraction to transform the input data into a reduced set of features that extract the relevant information from the input data.

Following the concept introduced in [42], the feature extraction process can be expressed in terms of the definition of the zone of measurement, and then measure the information required from that zone. This is the process generally followed in this work. Since we are working with the original

images, the pre-processing step was added to correct some deficiencies in the input images. Namely, illumination correction and noise filtering are performed depending on the particular set of features. The zone of measurement in this case is the whole area of the cell, which is defined by the mask obtained as a result of the preceding segmentation. The final measurement on the transformed image delivers the feature value, which is a scalar.

In this work, the task is to distinguish whether or not a red blood cell is infected by malaria and, therefore, the selected features must provide information with which it is possible to carry out such classification. When extracting features for the subsequent classification, it is advantageous to apply expert, a priori knowledge to a classification problem [43], [44].

3.2.4.1 Intensity features

Intensity features are based only on the absolute value of the intensity measurements in the image. For most of the feature extraction methods, only one intensity value per pixel is assumed, i.e. the methods assume a gray-scale image. For some methods, the original RGB image is converted to gray-scale by eliminating the hue and saturation information while retaining the luminance. For certain features, the performance have been evaluated for several channels to choose the channel with best discriminating power. Intensity values of the original image represent the transmitted light. However, for calculating some features, the pixels values from the extinction image can be more directly useful [42].

A set of features is proposed and implemented in this stage of the project. The selection of the features for the further evaluation is based on the visual differences between infected and non-infected red blood cells, the measures

of infected red blood cells that are commonly used by other cytological studies. The chosen features can be categorized as intensity features.

The two intensity features used in this work are Variance and Skewness, which are calculated as follow [21], [45]–[47]:

Let h(v) denotes the frequency of pixel intensity value v ($v = 1, \dots, N$) in the object's histogram (h) and p(v) is the probability function, which is computed from the histogram by dividing it by the object's area(A).

$$\mathbf{A} = \sum_{\mathbf{v}} \mathbf{h}(\mathbf{v}) \tag{3.1}$$

$$\mathbf{p}(\mathbf{v}) = \frac{\mathbf{h}(\mathbf{v})}{\mathbf{A}} \tag{3.2}$$

$$\boldsymbol{\mu} = \sum_{\mathbf{v=1}}^{\mathbf{N}} \mathbf{v}\, \mathbf{p}(\mathbf{v}) \tag{3.3}$$

- **Variance**

$$\boldsymbol{\sigma}^{\mathbf{2}} = \sum_{\mathbf{v=1}}^{\mathbf{N}} (\mathbf{v} - \boldsymbol{\mu})^{\mathbf{2}}\, \mathbf{p}(\mathbf{v}) \tag{3.4}$$

- **Skewness**

$$\boldsymbol{\mu}_{\mathbf{3}} = \frac{\mathbf{1}}{\boldsymbol{\sigma}^{\mathbf{3}}} \sum_{\mathbf{v=1}}^{\mathbf{N}} (\mathbf{v} - \boldsymbol{\mu})^{\mathbf{3}}\, \mathbf{p}(\mathbf{v}) \tag{3.5}$$

Equation (3.3) is used to calculate the mean, which gives an estimate of the average intensity level in the region of the cell and the variance is a measure

of the dispersion of region intensity. Histogram skewness is a measure of histogram symmetry and it shows the percentage of the region's pixels that favor intensities on either side of the mean.

3.2.4.2 Threshold features

Here a suitable threshold is defined as clutch that can distinguish between the infected and non-infected RBCs. The experiments shows that most of the infected RBCs have some pixels with values (i.e. gray level) less than 130, these pixels are mostly within the parasite region. In this work the threshold is 130, this value is calculated by experiments.

The features described above are chosen among many set of features, because they can lead to higher discriminative capabilities and then can improve the classification performance. Data collected from features is used to construct a Database. The computation of the intensity and Threshold features is implemented in function `m_feature.m` Appendix [A].

3.2.5 Detection of plasmodium parasites

Detection of Plasmodium parasites was done by using two methods, the first method by using the developed algorithm based on parasite features and characteristics. The second method by using a trained multilayer neural network. The network was trained with the features extracted form RBCs. The network searched through the images and identified regions infected by Plasmodium parasites.

3.2.6 Database creation

The database is created by using a total of 77 images from these images, a total of 1120 erythrocytes sub-images were cropped. 120 sub-images comprised of infected erythrocytes while 1000 images were non-infected. The features extracted from all available erythrocytes.

3.2.7 Artificial neural network (ANN) classification

NPRTOOL command was used to generate a MATLAB script which solves a Pattern Recognition problem with a Neural Network and it uses back propagation algorithm. In the script, firstly the input and target data was defined. The input data is a matrix with dimension of 736 x 3 which represents the selected features, and the target data is a matrix of 2 x 736 and contains zeros and ones only with respect to normal and abnormal features in the input matrix. The hidden layers was set to 20 then the network training was done using MATLAB™ function `train` which it's input is the created hidden layer, input data, and the target, then network was tested and its performance was found. The command `view` was used to view the network windows and finally the network was saved as `Test.mat`. The previous illustrated network that was created was running alt of time as training process of the network until a suitable accuracy was acquired.

TABLE OF CONTAINTS

Table of Contents