UNIVERSITY OF CAPE TOWN

DEPARTMENT OF COMPUTER SCIENCE

CS/IT Honours
Final Paper 2019

Title: MalCADE: Computer Vision and Supervised Learning Techniques for Computer Aided Detection of Malaria

Author: Michael White

Project Abbreviation: MALSEG

Supervisor(s): Patrick Marais

Category	*Min*	*Max*	Chosen
Requirement Analysis and Design	*0*	*20*	
Theoretical Analysis	*0*	*25*	
Experiment Design and Execution	*0*	*20*	20
System Development and Implementation	*0*	*20*	5
Results, Findings and Conclusion	*10*	*20*	20
Aim Formulation and Background Work	*10*	*15*	15
Quality of Paper Writing and Presentation	*10*		10
Quality of Deliverables	*10*		10
Overall General Project Evaluation (*this section allowed only with motivation letter from supervisor*)	*0*	*10*	
Total marks		**80**	80

MalCADe: Computer Vision and Supervised Learning Techniques for Computer Aided Detection of Malaria

Michael James White
Department of Computer Science
University of Cape Town
mike.james.white@icloud.com

ABSTRACT

Malaria is a devastating disease that leads to hundreds of thousands of deaths each year. Currently, most malaria diagnoses are performed manually by experts using microscopes, which is time consuming. This may result in it taking longer to diagnose patients, especially those in poor and rural areas. This problem provides motivation for the development of reliable computer aided detection (CADe) tools. The application of deep learning techniques, specifically convolutional neural networks (CNNs), has seen success in the existing literature. However, CNNs are more computationally expensive and require more training data than non-deep supervised techniques, such as random forests (RFs) and support vector machines (SVMs). There has been limited application of these non-deep techniques in the existing literature. This paper evaluates the performance of SVM and RF models when applied to the problem of malaria diagnosis, with the aim of understanding the viability of their use in real world CADe systems. Testing on a set of 20000 images, accuracy scores of 96.29% and 93.06% are achieved with tuned RF and SVM models, respectively. Testing on a small dataset of images gathered from a different source achieves similar performance, suggesting the models may generalise to different imaging conditions. The developed models are shown to achieve higher recall than existing supervised approaches. Moreover, the RF model achieved accuracy, recall and precision within 4% of the highest performing CNN approach.

1 INTRODUCTION

Malaria is a parasitic disease that can have devastating effects, not only for individuals who contract it, but also for their families and communities, which may suffer economic harm as a result [30]. Malaria disproportionately affects African countries, with 92% of global infections and 93% of global deaths falling in the World Health Organisation (WHO) African region. Malaria also disproportionately affects poorer and rural areas, where access to diagnosis and treatment is limited. According to the WHO, prompt diagnosis and treatment are the most important factors in preventing mild cases from becoming more severe [37].

This presents a clear problem, as those who are most in need of medical assistance are less likely to receive it in time. Currently, the gold standard for malaria diagnosis is manual microscopy performed by an expert. While this is a reliable method of diagnosis, it is also costly and time consuming [31]. Part of the reason poorer and rural areas are worst affected by malaria may be attributable to a lack of access to these experts, whether due to cost or location. This motivates the need for a low-cost and reliable computer aided detection (CADe) system that minimises the time burden on expert medical professionals.

Computer vision techniques may be applicable in the development of CADe systems for malaria diagnosis from blood smear images. The field of computer vision (CV) has seen important advances in recent years due in large part to the rise in popularity of deep learning and convolutional neural networks (CNNs), and their application to computer vision tasks. While they have proven to be highly effective in a broad range of CV tasks, they require large sets of training data. For example, Krizhevsky et al. used CNNs to achieve significant improvements in the task of image recognition on the ImageNet dataset, which is made up of over 15 million labelled images [15].

While CNNs have proven effective when directly applied to the problem of malaria diagnosis, with one example acheiving over 99% accuracy, the limited amount of publicly accessible data may be a limiting factor in widespread usage [28]. Moreover, the extensive computational resources required to run CNN models may limit their application in poorer and rural areas. These limitations motivate more focused efforts to apply non-deep techniques to the problem.

On the other hand, while non-deep machine learning classification approaches may be more limited in their performance, they also tend to be less reliant on large datasets. For example, support vector machines (SVMs) and random forests (RFs) have shown success in various computer vision tasks [9, 40].

This paper aims to evaluate combinations of supervised learning models and pre-processing techniques, comparing them to the most successful CNN and non-deep approaches documented in the existing literature. Fortunately, while many of the existing attempts at automated malaria diagnosis were trained and tested on private datasets, the current cutting edge CNN approach by Rajaraman et al. made use of a dataset that was subsequently made publicly available, enabling a direct comparison [27]. On the other hand, the top performing supervised learning approach by Diaz et al. was not evaluated on a public dataset, so a completely level comparison was not possible [8].

A secondary aim of this work is to deduce how well the proposed systems generalise to images collected under different conditions. While the cutting edge approaches mentioned above were shown to be effective on a large amount of test data, all of this data originated from the same source. This raises questions about the viability of these approaches when applied to data collected under different conditions. To limit similar concerns about the systems proposed here, an additional performance evaluation is run with test data gathered from a different dataset.

Section 2 presents background information on the techniques used in this work. Section 3 examines the related work around automated malaria diagnosis. Section 4 describes the evaluation framework developed to run experiments. Section 5 discusses the methodology followed when conducting experiments, as well as the experimental design. Section 6 discusses the results of experimentation, and lays out the limitations of the experimental process. Section 7 draws conclusions from the results, and Section 8 notes future work to be completed. Finally, Section 9 provides information on the ethics of the research.

2 BACKGROUND

This section presents background information on the evaluation criteria, machine learning models and pre-processing techniques applied in this paper.

2.1 Evaluation Criteria

It is important to ensure that any computational model is evaluated by the correct criteria, but this is even more critical when dealing with medical diagnosis. The output of a medical diagnosis model may inform treatment decisions, and so it is necessary to have significant evidence of the reliability of a model before it can be deployed. This section describes the criteria upon which the models proposed in this paper will be judged.

A medical diagnosis is a procedure that classifies a patient as positive or negative, depending on whether a disease is present or not. To evaluate the diagnostic procedure, we run it on data for which we already know the correct classifications. Results that correctly label a patient as positive are called true positives (TP), while those that falsely label a patient as positive are called false positives (FP). Similarly, results that correctly label a patient as negative are called true negatives (TN), and those that falsely label a patient as negative are called false negatives (FN) [2, 41].

These categories are useful for determining the reliability of diagnostic procedures. In particular, there are four widely used evaluation criteria based on the proportion of test results that fall into each category, namely recall (also known as sensitivity), precision, specificity and accuracy. Recall is a measure of the probability that a positive case will be detected as such, while precision is a measure of the probability that a positive prediction will be true. Specificity is a measure of the probability that a negative will not be falsely predicted as a positive. Specificity and precision are rarely used together, as they both intuitively indicate the performance of a model in not making false positive predictions. In this paper, precision is used due to its prominence in the machine learning field as a whole, and because it is directly implemented in scikit-learn, the machine learning library used for development of models in this paper. Finally, accuracy is a measure of the probability that any result will be correct [16, 41]. The formulae used to calculate these criteria are shown in Figure 1.

$$Recall/Sensitivity = \frac{TP}{(TP+FN)} \tag{1}$$

$$Precision = \frac{TP}{(TP+FP)} \tag{2}$$

$$Specificity = \frac{TN}{(TN+FP)} \tag{3}$$

$$Accuracy = \frac{(TN+TP)}{(TN+TP+FN+FP)} \tag{4}$$

Figure 1: Formulae for calculation of evaluation criteria

When screening for a disease, recall is highly important, as it could prove disastrous to incorrectly clear a patient who is actually infected. Often it is better to achieve a higher recall at the cost of some precision or specificity when screening for a treatable disease. If the patient is tested as positive, it may be possible to then perform a more thorough test before beginning treatment. For example, Lalkhen and McCluskey discuss testing for cervical cancer in women. Initially, a high recall but relatively lower precision and specificity smear test is done, which effectively screens for cases where it is highly unlikely that the disease is present. Then, for patients who test positive, a more costly investigative surgery procedure may be performed. While the smear test is not perfect, it serves the purpose of greatly cutting down the need for investigative surgery and therefore allows for more cost effective and time efficient handling of the disease [16].

Similarly, an automated malaria test with high recall could serve as an initial screening process to lower the burden on expert medical practitioners who would then only have to manually test those with a positive screening result. The benefit of this use case is supported by an interview done with a pathologist from PathCare, who stated that negative test results actually produce the greatest time burden. This is because the whole blood smear must be examined manually, which may take a human expert up to 40 minutes. An automated screening test with high recall has the potential to significantly reduce the time taken, and would require little human input. Moreover, in cases where one or more positives are detected, the system could flag these particular blood cells for investigation by the pathologist in an effort to reduce the time taken to confirm positive results.

2.2 Support Vector Machines (SVMs)

SVMs, first introduced by Cortes and Vapnik, are a machine learning approach typically associated with classification problems [7]. Intuitively, SVM classifiers fit a hyperplane between classes in a particular feature space. An optimal hyperplane is found by iteratively reducing error, with higher weight being placed on the support vectors, which are the data points closest to the division between classes. They are computationally efficient in a relatively small feature space, but when working with raw images, this feature space expands rapidly. As such, SVMs rely on feature extraction algorithms for computer vision applications.

SVMs have seen extensive use for CV applications in the existing literature. For example, Zhang and Wu used SVMs to classify images of fruit into 18 categories with an 88.2% accuracy, using a training set of 1322 images [40]. They have also shown success when applied to other medical diagnosis problems, such as brain tumor segmentation [17], which motivates their inclusion in this paper.

SVMs have numerous hyperparameters that can be adjusted, which may greatly impact their performance. The SVM models in this paper are tuned by changing three hyperparameters:

(1) Kernel function
(2) Degree
(3) Gamma value

The chosen kernel function typically has the greatest impact on the performance of a model, as it fundamentally changes the underlying calculations done to predict correct classifications. Two kernel functions are evaluated in this paper, namely Radial Basis Function (RBF) and Polynomial. The degree hyperparameter is only relevant to polynomial SVMs, defining the degree of the underlying polynomial function.

The gamma value determines how closely the classifier attempts to fit the training data. As the gamma value increases, it becomes more likely that the model will start to overfit the training data. Conversely, with too low a gamma value, the model may underfit the data.

2.3 Random Forests (RFs)

The idea of random forests was introduced by Ho, who described the benefits of generating many complementary decision trees operating in random subspaces of the entire feature space [12].

Traditional ensemble tree methods struggle to generalise as the complexity of the feature space grows. On the other hand, RFs were shown to achieve generalisation in more complicated feature spaces than would be possible with traditional decision tree methods, while still retaining high speeds of execution. For example, Ho demonstrated increased accuracy on the classification of images of hand-written digits. Breiman improved on Ho's model by integrating the concept of bagging, which reduces variance by creating trees each with only a subset of the overall training set [3, 4].

The combination of high speeds of execution and their ability to generalise even in complex feature spaces have made RFs popular for a broad range of CV tasks. For example, Fanelli et al. used an RF model to classify facial poses with a 90.4% accuracy, using a training set of 50000 automatically generated renders of a 3D modelled face [9]. RFs have also been applied to medical imaging problems, with Lee et al. having achieved over 98% sensitivity and 97% specificity when applying RFs to the problem of lung nodule detection [18]. The success of RFs when applied to other CV problems, such as those discussed above, motivates their inclusion in this paper.

RFs can be constructed using several hyperparameters, which may affect their output substantially. The models proposed in this paper will be tuned by adjusting two hyperparameters:

(1) Number of trees in the forest
(2) Maximum depth of trees

Typically, having more trees in the RF tends to improve performance, but also increases time taken for training and execution. As such, it is important to find a good balance, at the point where increases in the forest size start to give diminishing returns.

The maximum depth of trees in the forest dictates how far trees are allowed to expand. A high maximum depth may cause the model to overfit to the training data, while a low maximum depth may cause the model to underfit.

2.4 Image Filters

Filters are algorithms that can be applied to images to adjust their characteristics in various ways. Applying filters to images before they are fed into machine learning models may serve to accentuate differences in visual characteristics between positive and negative cases. This may, in turn, result in improved model accuracy. The image filters used in this paper are outlined below, and Figure 2 demonstrates the effect that each has on a sample image of an infected red blood cell.

2.4.1 HSV Conversion. The HSV colour space is made up of hue, saturation and value (the brightness of a pixel). This colour space is designed to better mimic the way humans perceive light compared to the more common RGB colour space, which simply represents pixels as a combination of red, green and blue colour intensities. Existing papers have shown greater success using HSV colour space compared to RGB, both broadly in the field of CV and specifically in relation to malaria diagnosis [1, 6].

2.4.2 Channel Isolation. Image channels are the different values in a colour space that describe each pixel. Typical three channel colour images provide a lot of information about each pixel, but this extra detail can be distracting to a machine learning model if there is little differentiation between positive and negative examples in a particular channel. Some papers in the field of malaria diagnosis have demonstrated success when isolating the saturation channel of HSV blood cell images, possibly due to the effects of the stain added before imaging [1, 23].

2.4.3 Contrast. The difference in intensity between the darkest and brightest parts of an image is known as contrast. In the case of image classification problems, image contrast may affect the ability of a model to differentiate between positive and negative cases. If there is a difference in intensity between the typical background of images and objects that must be detected, then increasing the image contrast may enhance the ability of a model to recognise this difference. Existing papers have used increased contrast as an initial enhancement for more effective detection of malaria parasites, justifying its use in the current work [1].

2.4.4 Threshold. Threshold functions operate similarly to contrast functions, and have the same justifications for use. However, while contrast functions change the intensity of pixels smoothly, threshold functions change the value of all pixels above a threshold to a specified value, and all pixels below that threshold to a different value. This may create a more dramatic separation between the object and background, possibly increasing the ability of models to detect objects. In the past, thresholding has been applied to the problem of hand gesture recognition to differentiate the foreground of an image from the background [26]. In this paper, we apply thresholding after contrast has been increased with the intention of better isolating the parasite from the blood cell.

2.5 Feature Extraction

It is infeasible to feed raw image data into SVM and RF models as this tends to result in poor performance and long runtimes, both when training and running predictions. Image features are summaries of images that can be extracted through traditional computer vision algorithms to greatly reduce the size of the input that is fed into models. Once extracted, features can be concatenated and flattened into a final feature vector before they are used as input to models.

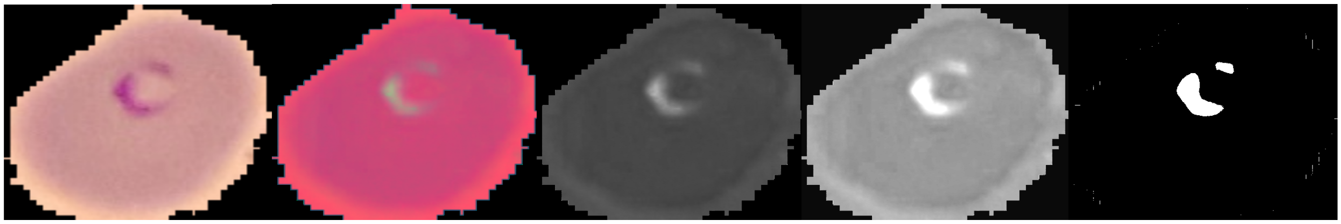

Figure 2: Image filtering techniques. From left to right, we first see the unfiltered image, followed by the HSV converted version. Next, the isolated saturation channel is displayed. Following this, we see the same isolated channel but with added contrast. Finally, we see the application of the threshold function after contrast boosting.

The feature extraction techniques used in this paper have all seen success when applied to various other image classification tasks. The details of each are discussed below.

2.5.1 Hu Moments. Hu proposed a set of seven moment invariants, all of which are invariant to scale, rotation and translation [13]. These features, known as the Hu moments, have been applied broadly to problems in the field of computer vision. For example, Otiniano-Rodriguez et al. achieved over 90% accuracy using the Hu moments as input to an SVM classifier for sign language recognition [25].

2.5.2 Haralick Texture Attributes. Haralick et al. presented a set of 14 textural features that can be extracted from images to improve image classification accuracy [11]. These features have since been used in a broad range of image classification tasks. For example, Mery et al. used haralick texture attributes as part of their selected feature space for automatic quality grading of corn tortillas [24]. Applications of haralick texture attributes extend to medical imaging applications, with Roula et al. having documented their use as part of their feature space for classifying prostatic neoplasia in microscopic images of samples taken by needle biopsy [29].

2.5.3 Histograms. Histograms sort the pixels of an image into a specified number of bins depending on their intensity in each colour channel. Existing works have used histograms for various image classification tasks. For example, Chapelle et al. extracted histograms from a sample of stock images in seven categories and used these as the sole inputs to an SVM classifier, achieving a best error rate of 11% [5]. Szummer and Picard used histograms as part of a feature set for indoor-outdoor image classification, resulting in 90.3% accuracy [32].

3 RELATED WORK

There has been significant research interest around the topic of automated malaria diagnosis, and the existing literature has demonstrated varying levels of success using a wide variety of techniques. This section critically analyses the existing literature to identify the most promising methods that have been proposed thus far, specifically those that have been evaluated rigorously on a large, disjoint test set.

The problem of automated malaria diagnosis can be thought of as two or three separate computer vision tasks. First, the raw blood smear image must be segmented into many images of individual red blood cells. Second, each cell must be classified as either infected or healthy. Third, as an extension on the second task, specific identification of the malaria species and stage of development could be performed, though this is a more complex problem that would likely require significantly more data to generalise successfully. Each of these tasks are discussed in separate subsections.

3.1 Segmentation

Segmentation refers to the process of automatically recognising relevant objects in an image and producing cropped sub-images of each object identified by the algorithm. The main performance indicators of a segmentation algorithm are how closely it is able to crop each object (minimising background distraction), and how accurately it is able to recognise objects.

The primary challenges of cell segmentation are to deal with overlapping blood cells and to differentiate between blood cells and distractors, such as dust or stain marks on the blood slide.

Numerous approaches have shown success when applied to the problem. One such approach is the use of a circular hough transform. This measures the minimum and maximum radii of a suspected red blood cell and averages this to infer a best fit when cropping. Various papers have achieved accuracies of over 90% in counting the number of red blood cells with this approach [20, 21]. For example, in their malaria diagnosis system, Rajaraman et al. used a level-set based algorithm to perform blood cell segmentation, for which they did not provide runtime metrics [27]. While the authors showed it to be quite effective, with a positive predictive value of 94.4%, level-set segmentation is typically quite costly [35]. Given that deployment of these systems is likely to have the greatest impact in rural clinics with limited computational resources, it is not clear what effect this may have on the system's real world viability.

Due to both time constraints as well as the extensive collection of existing work, it was decided not to develop a segmentation algorithm for this paper. Rather pre-cropped red blood cell images are used as inputs to the infection classification models that form the focus of the paper.

3.2 Infection Classification

Infection classification is the process of taking in an image of a red blood cell and classifying the cell as infected or uninfected. The performance indicators of infection classification algorithms are sensitivity, precision, specificity and accuracy (described in section 2.1).

Liang et al. proposed a solution to malaria infection classification based on CNNs, showing promising results. They achieved accuracy of 97.37%, sensitivity of 96.99% and specificity of 97.75% after training the CNN with 24300 labelled images and testing with a disjoint set of 2700 images. The authors do acknowledge some limitations of the approach, specifically the need for large sets of training data and significant computing power for a CNN model to be viable. Unfortunately, all blood smear images used in this study were acquired from the same hospital archive. As a result, the model may be biased towards the particular conditions under which those images were collected [19]. In order to ensure robustness and real world applicability of the model, it would need to be able to accurately classify images collected in a wide range of conditions. The use of a small testing set causes further doubt over the final results.

Rajaraman et al. used CNNs to achieve accuracy of 98.6% and later improved this to 99.5%, using a public dataset hosted by the U.S. National Library of Medicine [27, 28]. The improved model demonstrated precision of 99.8%. However, these metrics are given at the patient level rather than the cell level, which is more commonly used in the existing literature. Unfortunately, similarly to Liang et al., the dataset consisted of images taken under the same conditions, with the same staining method and from the same archive. Moreover, the images of individual blood cells were segmented using a level-set based algorithm, which may be computationally expensive. This complexity would compound with the already resource intensive CNN model, possibly limiting the viability of the system for deployment in poorer or rural areas if the model is not able to maintain high performance when less computationally expensive segmentation techniques are applied.

Transfer learning is a technique that involves using a CNN for feature extraction before performing classification with an external algorithm. Mehanian et al. propose a solution for malaria diagnosis using transfer learning with a feature extraction CNN and non-deep logistic regression classifier. They demonstrated good performance, having achieved sensitivity of 91.6% and specificity of 94.1%. Though the results of their study are not quite as positive as Liang et al. or Rajaraman et al., they also tested the model on substantially more data that originates from 12 countries [22]. This suggests that the model developed by Mehanian et al. may be more robust and have more real world applicability. While the logistic regression classifier is a non-deep supervised learning technique, the reliance on the underlying CNN feature extractor suggests that the real world application of this approach would still be limited by its computational complexity.

While deep learning approaches have been more prevelent in the existing literature, there are a few papers using non-deep supervised learning techniques. For example, Tek et al. used a k-nearest-neighbours (KNN) classifier to perform binary detection of malaria parasites and reported an accuracy of 93.3% [33]. However, this high accuracy is achieved due to the combination of an imbalanced testing set, with many more negatively classified images, and a high specificity of 97.6%. The sensitivity of the approach was measured as only 72.4%, suggesting that the approach may not be viable for real world applications. On the other hand, Diaz et al. achieved 94% sensitivity and 99.7% specificity using an SVM based approach [8]. However, these results were achieved when running the model on the full dataset used in the study, including those images used for training, casting some doubts over the validity of the reported results. Nonetheless, these results have been used as a benchmark in other studies evaluating non-deep approaches [36].

3.3 Species and Stage Classification

Species and stage classification is an extension of the previous task of infection classification. This process involves taking in an image of an infected red blood cell and classifying the parasite's species and life cycle stage. This is a significantly more complex problem than simply labelling a cell as infected or not.

There are four species of the Plasmodium parasite that generally cause malaria in humans, namely *P. vivax*, *P. falciparum*, *P. malariae* and *P. ovale*. Less often, humans may be infected by a fifth species, *P. knowlesi*, but this normally affects animals. Each species appears differently under a microscope [38]. Adding further complexity, each parasite goes through a life cycle defined by three stages. At each stage of a parasite's life cycle, it appears differently, and the distinction between different stages may not be clear as it transitions from one to another. Plasmodium parasites begin life as a trophozoite and later develop into a schizont and eventually a gametocyte [38]. This means that, overall, there are 15 species-stage combinations that would need to be classifiable by a successful algorithm.

Some existing attempts at automated classification of malaria species and stage can be found in the literature. Tek et al. made use of the k-nearest-neighbours algorithm for malaria species and stage classification, proposing two configurations [33]. The first configuration is a 16-class classifier that assumes the input image is of an infected cell and classifies it as one of the 15 possible combinations of species and stage or, if the classifier is unable to conclusively categorise the input, as an inconclusive image.

The second configuration is a 20-class classifier that combines simple infection classification with the more complex problem of species and stage classification. It classifies an input image as one of the 15 species-stage combinations, as one of four non-parasite artefacts, or as inconclusive. The 16-class classifier achieved accuracy of 91.2% in diagnosing the parasite species and 90.1% in diagnosing its life stage. The 20-class classifier achieved accuracy of 94.4% in overall infection detection, 90.6% in diagnosing parasite species and 89.9% in diagnosing parasite life stage. The study is limited by the relatively small amount of data used, with 669 blood cell images labelled as infected and 3431 labelled as uninfected. This limited amount of data led to the authors using the leave-one-out approach, which seeks to maintain independence between training and testing data without making them completely disjoint as mandated by the hold-out approach [10]. While better than simply reusing the training set for testing, the leave-one-out approach may still produce results that are more optimistic than warranted.

It was decided to limit the scope of this paper to simple infection classification rather than stage and species classification due to both the short period of time available to conduct the research as well as the lack of publicly available datasets with stage and species labels.

4 EVALUATION FRAMEWORK

A framework was developed to automatically run evaluations on a set of models, with the goal of simplifying the evaluation process. Three specific requirements were identified:

(1) The framework needed to handle loading of various datasets, while ensuring good evaluation standards by automating the process of splitting data into training and testing sets.
(2) It also needed to allow for the decoupled development of models and the framework itself, which was done by providing a model interface to be implemented by concrete model classes.
(3) Finally, the framework needed to facilitate the use of short testing scripts to specify the design of experiments.

Each of these framework components are described in detail below.

4.1 Dataset Loader

To ensure the validity of results, an iterative random hold-out approach was taken, whereby random subsets of the dataset are selected at each iteration to form the training and test sets. The test set is not used for training models so that when testing occurs, that data is unseen by the model. The metrics achieved for each iteration are aggregated to form the final result.

K-fold cross validation is another popular approach that has been used widely in the existing literature, especially when the size of datasets is very limited. However, when working with large datasets, both approaches have been shown to be accurate in judging model classification performance [14, 39]. The dataset loader implemented the randomised loading of data, and automatic splitting into training and test sets, while the evaluation runner, described below, handled the iterative process.

The Python glob and os packages are used to get lists of files in a specified dataset folder. Python's random package is used to select random samples of files in the specified directory, which are then loaded into memory. Once loaded, files are resized to specified dimensions to achieve consistency across all images in the dataset. Loading and resizing of image data is handled using the OpenCV library. Once loaded and resized, images are split into training and test sets of specified sizes and returned to the evaluation runner.

4.2 Model Interface

The model interface is an abstract class with two methods that must be implemented in concrete sub-classes, for training and running the model. The benefit of this is that the details of each model's implementation are completely hidden from the evaluation runner, decoupling the development of each. This allows much greater flexibility in designing models, while still making it simple to integrate them with the evaluation runner.

For this paper, concrete SVM and RF model classes are developed. These act as wrappers for model implementations provided by the scikit-learn library, to allow their use with the evaluation runner. These model classes also handle the application of image filtering and feature extraction algorithms to input data.

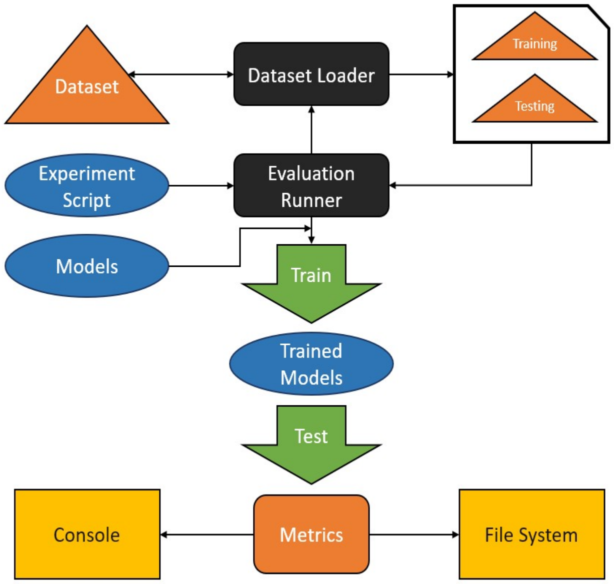

Figure 3: Evaluation framework system diagram.

4.3 Evaluation Runner

The evaluation runner handles the training and testing of models, and recording of metrics. The primary benefit of this framework component is that experiments can be designed without needing to write code to explicitly handle these routine tasks. It allows for a number of iterations to be set, with each iteration retrieving new data from the dataset loader before running the same training and testing process. Metrics are calculated using the scikit-learn metrics package and, for each iteration, printed to the console and written to file. The resulting trained models for each iteration are dumped to file using the joblib framework, which allows them to be loaded at a later point for further testing. Finally, the metrics from all iterations are aggregated, printed to console and written to file.

4.4 Software Architecture

The framework and models were developed using Python (version 3.6.7) and various libraries that have been noted above. Specific efforts were made to follow the PEP-8 code style conventions, which have been widely adopted by Python developers [34]. The code has been thoroughly documented and commented to ensure that it is simple to understand by other developers. Assertion testing was conducted to ensure the validity of the produced code. Git source control was used and code was regularly pushed to a remote GitHub repository.

5 METHODOLOGY

This section lays out the process by which the research aims were addressed. Specifically, two research questions are posed:

(1) Can non-deep supervised learning models paired with image filtering and feature extraction algorithms achieve accuracy scores within 5% of the top performing deep learning model when trained and tested on the same dataset?

(2) Does the performance of models degrade when applied to testing sets gathered from different sources to the training set?

The details of the datasets and computational equipment used are presented, followed by in-depth explanations of the model tuning and experimentation processes.

5.1 Datasets

Two datasets are used to run the experiments detailed in this study. The first is a publicly accessible dataset, originating from a study by Rajaraman et al. and made available by the U.S. National Library of Medicine (NLM) [27]. This dataset is made up of pre-cropped blood cell images, with 13779 classified as infected and the same number classified as uninfected, for a total of 27558 images.

The second is a private dataset provided to the author by PathCare Laboratory Services. This dataset was provided as blood slide images, not pre-cropped but with infected cells having been identified by a pathologist. As mentioned previously, it was decided not to implement an automatic cropping algorithm for this paper due to time constraints. As such, this dataset was manually cropped into a set of 120 cell images, with 60 labelled as infected and 60 labelled as uninfected.

Images in each dataset were not of a standardised size, and so it was decided to resize all images to 50 pixel by 50 pixel squares. These measurements were decided to avoid artifacts generated from excessive upscaling or stretching in either axis, with most of the original images being about the same size or larger and roughly square in shape.

5.2 Computational Equipment

Development, training and testing were all conducted on the same machine, a laptop running Ubuntu 18.04.1 LTS. The machine had 16GB of RAM and a four core Intel i7 CPU, with clock speeds of 2.7GHz. The machine did not have a dedicated graphics card. The operating system and all program code were stored on an solid-state drive.

5.3 System Selection and Tuning

We can think of each combination of model, hyperparameters, feature extraction algorithms and image filtering algorithms as an experimental system. Due to the high number of potential experimental systems, it was infeasible to perform an exhaustive search for the best performing combination, though this would theoretically result in an optimal system. Instead, a three-stage system selection process was designed, with each stage focused one component of the system. The selected options in each stage are used in the following stages until final tuned systems are selected at the end of the third stage. These stages are described in subsections below.

Evaluations at each stage are made based on the mean accuracy, recall and precision metrics, rounded to four decimal places, obtained from five iterations. For each iteration, systems are trained on 5000 images and tested on disjoint sets of 5000 images.

5.3.1 Feature extraction algorithms. The feature extraction algorithms described in section 2.5 are considered. Greyscale and colour histograms are evaluated with 2, 4, 8, 16, 32 and 64 bins to determine the optimal size for each. In cases where explicit channel isolation was not conducted, the standard greyscale image conversion function provided by OpenCV was applied preceding greyscale histogram extraction. Because this was only an initial parameter selection process, accuracy was used as a singular metric for evaluation. More granular examination of the balance between recall and precision is done in future evaluations. After the selection of bin sizes, the performance of these histograms is compared with that of Haralick texture attributes and Hu moments. The most successful feature extraction approaches are then used in the image filtering selection phase.

5.3.2 Image filtering algorithms. The image filters described in section 2.4 are considered in various combinations. Specifically, models with the previously selected feature extraction algorithms are used with five different image filter combinations:

(1) No filters
(2) HSV conversion
(3) Saturation channel isolation
(4) Saturation channel isolation with contrast
(5) Saturation channel isolation with contrast and thresholding

The most successful image filtering and feature extraction combinations for each model are then used in the final hyperparameter tuning phase.

5.3.3 Hyperparameter tuning. The hyperparameters for each model, discussed in sections 2.2 and 2.3, are tuned by adjusting one at a time, finding the optimum value and locking that value for the rest of the tuning process. While in an ideal situation tuning would search through all the possible combinations of hyperparameters, it is not feasible in this case given the high number of potential combinations.

The SVM model is first tuned by selection of its kernel function. The standard RBF kernel used by default in the scikit-learn library is compared to 1st, 2nd, 3rd and 5th degree polynomial kernels. The SVM model is then tuned by selection of its gamma value. Gamma values that are tested include the automatically calculated gamma value used by default in scikit-learn as well as 1, 0.1, 0.01 and 0.001. The optimal resulting model is selected to proceed to the two final experiments.

The RF model is first tuned by the selection of its forest size. Forests with 1, 10, 50, 100 and 250 trees are evaluated. Following the selection of the number of trees, the maximum depth of trees is evaluated. Maximum depths of 10, 100, 250 and 1000 are compared, along with the default scikit-learn setting of no maximum depth, to select the final RF model to proceed to the final two experiments.

5.4 Experiment Design

Following the system selection process, two experiments are conducted on the final selected systems, allowing for evaluation of various experimental hypotheses.

5.4.1 Experiment 1: Performance of systems on NLM data. Systems are evaluated on their ability to operate on data that is part of the same dataset used for training. Results are obtained from ten iterations, each time trained on 5000 images and tested on a disjoint set of 20000 images. This experiment is designed to evaluate the ability of the systems to predict infection in images collected in a similar way to the training data. It was hypothesised that the best systems proposed in this paper would not outperform the most successful CNN based approaches in the literature, but that they would achieve accuracy within 5%, and improve on existing supervised approaches.

5.4.2 Experiment 2: Performance of systems on PathCare test data. The systems are put through a final evaluation to test their ability to operate on datasets other than the one used for training. Results are obtained by loading models from the first experiment, pre-trained on the NLM dataset, and tested on the full PathCare dataset of 60 infected and 60 uninfected images.

This experiment is designed to evaluate the generalisability of systems trained on a dataset gathered from a singular source, answering the second research question. It was hypothesised that the prediction performance on the PathCare dataset would be worse than that seen on the NLM dataset used for training, as it is unlikely that the systems will generalise successfully without a broad range of collected data.

6 RESULTS AND DISCUSSION

In this section, the results of system selection and tuning and the two experiments are presented. The implications of these results are discussed, and possible explanations are laid out. Finally, limitations of the experimentation process are noted.

6.1 System selection and tuning

This subsection details the results of system selection and tuning. Detailed tables of results are available in the supplementary information, from S1 to S8.

6.1.1 Selection of feature extraction approaches. Models showed significant accuracy improvements as the number of bins increased up to 64 for greyscale histogram inputs. Colour histograms resulted in peak accuracy at 16 bins for both models. Unfortunately, 64 bin colour histograms caused a memory error on the computer used for testing and 32 bin colour histograms resulted in prohibitively long training and execution times when fed into SVM models. Based on these initial results, 64 bin greyscale histograms and 16 bin colour histograms were selected.

Following the bin size selection, colour and greyscale histograms were compared to Hu moments and Haralick texture attributes. Results when using Hu moments were significantly worse, for all metrics, than any of the other feature extraction approaches, for both models. Colour histograms performed the best on all metrics, for both models. Haralick texture attributes resulted in better accuracy than greyscale histograms for the SVM model, but the opposite was true for the RF model. It was decided to eliminate Hu moments at this point, but evaluate each of the other techniques further in the next stage.

6.1.2 Selection of image filtering approaches. Interestingly, while colour histograms achieved the best performance when no image filtering was applied, once these filters were applied, other feature extraction methods produced better results.

In almost every case, filtering significantly improved performance, though colour histograms saw minimal performance improvements when used with the RF model, and actually decreased in performance when used with the SVM model. In the case of RFs, it was expected that the performance increase would be minimal, as unfiltered colour histograms already demonstrated high accuracy of over 95%.

Haralick feature extraction performance improved significantly for both models when image filters were applied. In particular, the SVM model saw its highest performance when used with these combinations. SVMs saw a 17.15% increase in accuracy with Haralick feature extraction on images with HSV conversion. This image filtering approach only attained a recall of 90.55%, however. On the other hand, while the application of thresholding on the contrast boosted saturation channel saw 0.55% lower accuracy, it saw an increase of 5.19% in recall, with a relative drop in precision. These methods were the best performing overall for the SVM model, but it was decided to select the thresholding approach for the next stage despite its lower accuracy and precision, due to the high importance of recall for automated medical screening tests.

While the RF model also saw significant performance improvements with Haralick combined with image filtering, in the best case seeing an accuracy improvement of over 20%, even better performance was attained with histogram extraction. HSV converted colour histograms saw very slight improvements in all metrics compared to the unconverted version, but each of these metrics were lower compared to the best greyscale histogram combination. When the greyscale histogram was used with isolated saturation channel filtering, it achieved an accuracy of 95.98%. This was the best overall accuracy of any filtering method, though the contrast boosted version achieved a very minor improvement in recall of 0.02%. The cost of this minimal improvement would be 2.22% accuracy, so it was decided to select the isolated saturation channel filtering without contrast boosting for the next stage.

6.1.3 Hyperparameter tuning. The SVM model was first tuned by selection of its kernel function. Overall it was observed that the performance difference between different kernel functions was very minimal. The 3rd degree polynomial kernel demonstrated the highest accuracy of 92.9%, and was therefore selected to go forward to gamma value tuning.

Again, very minor differences in performance were observed between different gamma values. The automatically calculated gamma value achieved the highest recall value of 96.36% and the second highest accuracy of 93.07%. The highest accuracy score, achieved with a gamma value of 0.1, was only an improvement of 0.02%, but the recall score achieved was 0.51% lower. It was chosen to use the automatically calculated gamma value going forward into the two experiments, with the intention of optimising for higher recall.

The RF model was first tuned by selection of its forest size. All metrics peaked at a forest size of 100, so it made sense to select this going forward. Next, the maximum depth value was tuned. Increasing this value had very little effect after 100, and even the increase from 10 to 100 only had an accuracy improvement of 0.33%. It was therefore decided to select a value of 100, as unnecessarily increasing this value may result in overfitting of the data and increased computation times.

Model	Accuracy	Recall	Precision	Train Time	Test Time
Tuned SVM	0.9306	0.9613	0.9058	63.9346s	254.9055s
Tuned RF	0.9629	0.9606	0.9649	1.2108s	1.0796s

Table 1: Results of Experiment 1. The best result for each column is highlighted.

Model	Accuracy	Recall	Precision	Train Time	Test Time
Tuned SVM	0.9417	1.000	0.8955	n/a	2.053s
Tuned RF	0.9667	1.000	0.9375	n/a	0.016s

Table 2: Results of Experiment 2. The best result for each column is highlighted.

6.2 Experiment 1: Performance of Tuned Models on NLM Data

The results observed when testing on the NLM dataset were highly positive. The SVM system achieved an accuracy of 93.06%, recall of 96.12% and precision of 90.58%. The best performing existing approach by Diaz et al. does not report precision but reports specificity of 99.7% and recall of 94% [8]. As described in section 2.1, precision and specificity are comparable metrics. While the SVM system's precision is significantly lower than the specificity reported by Diaz et al., it also demonstrates a 2.12% improvement in recall. The RF system achieved an accuracy of 96.29%, recall of 96.06% and precision of 96.49%. This amounts to a 2.06% increase in recall with a 3.68% lower precision than the specificity achieved by Diaz et al.

As argued in section 2.1, for an automated screening test increases in recall may be more beneficial than higher specificity or precision. In this sense, the hypothesis that the systems in the current work would improve on existing supervised approaches is confirmed, though it is acknowledged that the improvement is not observed across all metrics.

The current work also evaluates models on significantly more testing data than Diaz et al.: 20000 blood cells compared to 12557. Moreover, the parasitemia of the testing data used by Diaz et al. is reported as 5.6%, amounting to approximately 703 infected blood cells. On the other hand, the testing data used in this experiment is balanced, with 10000 infected cells, possibly resulting in a more accurate reflection of the real-world recall that can be expected from the system.

The models did not achieve better results than the CNN approaches detailed by Rajaraman et al. However, the RF model's metrics all fell within 5%, as hypothesised [27, 28]. The SVM achieved a recall score within 5% of that reported for the approaches by Rajaraman et al., but precision and overall accuracy did not fall within this bracket. Mehanian et al. achieved recall of 91.6% and specificity of 94.1% with their transfer learning model [22]. Both the SVM and RF systems improve significantly on this recall, and the RF also shows higher precision than the specificity reported by Mehanian et al. The testing set used in this paper was significantly larger than those used by both Mehanian et al. and Rajaraman et al., though that used by the former was far more diverse, with images taken under different conditions and originating from 12 countries.

Papers in the existing literature have not reported training or testing time metrics, so it is not possible to provide a comparison here. However, comparing the RF and SVM systems proposed in this paper, it is clear that the RF system is significantly less computationally expensive. The testing time of the SVM system is a factor of 236 greater than that of the RF system, while the training time is a factor of 53 greater. These runtime metrics suggest that the RF system may be more suitable for deployment in situations where computational power is limited.

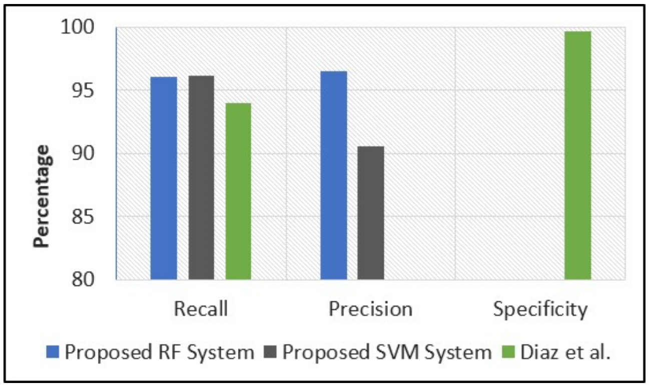

Figure 4: Comparison of proposed systems with the current best performing non-deep approach.

6.3 Experiment 2: Performance of Tuned Models on PathCare Data

Both the SVM and RF systems saw slight improvements in accuracy when run on the PathCare dataset. The precision of both decreased: by 1.03% for the SVM system and 2.74% for the RF system. However, each achieved 100% recall. It must be noted that these results are likely overly optimistic due to the exceptionally small size of the PathCare dataset. For this reason, further work is warranted using a larger dataset. With this limitation acknowledged, the hypothesis that performance would decrease on the PathCare dataset is falsified, if recall is judged to be more important than precision for automated screening systems, as argued in section 2.1. However, it is acknowledged that the increased performance was not universal across all metrics.

The runtime metrics of this experiment confirm the observation made in the first experiment, that the RF system is more computationally efficient than the SVM system. While train times are not evaluated, as the systems were pretrained, the test time of the SVM system is a factor of 128 greater than that of the RF system.

6.4 Limitations

The small size of the PathCare dataset limits the credibility of the results of the second experiment. While the results achieved were very positive as an initial evaluation, further experimentation on a significantly larger set of data is necessary to confirm the viability of the system.

Due to time constraints, an automatic cropping algorithm was not developed for this paper and instead manual cropping of the

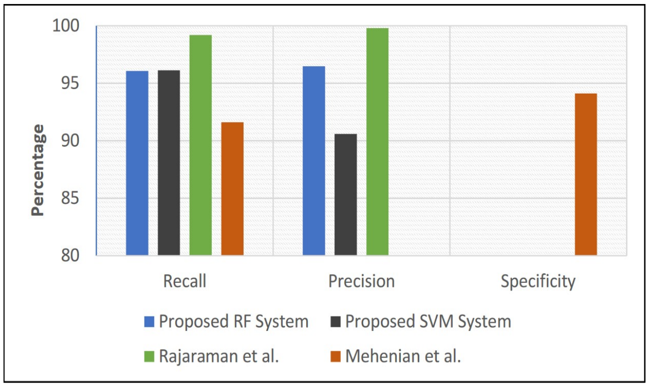

Figure 5: Comparison of proposed systems with existing deep learning approaches.

PathCare dataset was performed. While there are successful examples of automatic segmentation of red blood cells in the existing literature, it is not certain what effect these algorithms may have had on the performance of any of the models presented in this paper. Certainly, this warrants future work to build and evaluate a full diagnosis pipeline that performs both segmentation and infection classification.

The data that was made available for this research was only classified in two classes: as parasitised or non-parasitised. However, in reality there are multiple species of the malaria parasite, each with various life stages. Because the data was only classified in this binary manner, it is unclear whether the models generalise to provide similar performance for all species and life stage combinations.

7 CONCLUSIONS

The conducted experiments demonstrated the usefulness of supervised learning techniques as an alternative to the popular deep learning approaches, which have been the primary focus of the existing literature. Various conclusions can be drawn from the results of these experiments.

Firstly, the proposed RF and SVM systems outperform existing supervised approaches in terms of recall, which is seen as the most important metric for judging the performance of initial screening tests. Secondly, the proposed systems are able to achieve recall within 5% of that reported by existing CNN approaches. This indicates that supervised techniques may be an effective alternative in situations where the high computational power required by CNN systems is not possible, such as in rural clinics. Thirdly, initial results when run on a small dataset gathered from a different source to the training data seem to indicate that the promising performance demonstrated by the proposed systems may generalise to various imaging conditions. However, further work is necessary to confirm this. Fourthly, the RF system appears more suitable for real-world use, as it achieves better accuracy and is far more computationally efficient than the SVM system. Finally, increased performance is demonstrated when systems apply appropriate image filtering as pre-processing before feature extraction is conducted, which encourages the use of image filtering in future work on this problem.

On the whole, it is shown that non-deep supervised learning techniques have great promise for reducing the burden on medical professionals in performing malaria diagnosis. This may, in turn, result in much faster times to diagnosis, and allow quicker intervention, which is described by the WHO as the most important factor in preventing severe cases and deaths from occuring [37]. The higher computational efficiency of non-deep systems, such as those presented in this paper, may allow for more widespread adoption, especially in areas where extensive computational resources are not available. Thus, the value added by these systems may have a significant impact on rural and poor communities, particularly those in the WHO African region.

8 FUTURE WORK

The systems proposed in this paper show promise as a computationally cheap alternative to CNN based systems, with a lower training data requirement. As such, future work to develop a fully integrated CADe system is warranted. Such a system should include the full computational pipeline of automated blood cell cropping, pre-processing, feature extraction and classification. Evaluation of this system would focus on the accuracy in predicting parasitemia for blood smear images, which is likely to be a more acceptable metric for predicting real-world performance of the system than the accuracy of predicting infection on a per cell basis.

Rajaraman et al. improved the performance of their initial CNN approach by adopting an ensemble strategy, whereby numerous CNNs each 'vote' with their classification of a cell and the classification with the most votes is returned [28]. Similarly, future work may produce better results using an ensemble of the most successful non-deep models presented in this paper.

9 ETHICS

This paper involved the use of both a publicly-available dataset from the U.S. National Library of Medicine, as well as a private dataset acquired from PathCare Laboratory Services, a pathology practice in South Africa. Both datasets are de-identified, containing no personal or demographic information on the patients corresponding to each blood cell image. As such, the blood samples in each dataset can not be linked back to the patient from whom they were taken, or any particular demographic group.

Ethics clearance was sought from the University of Cape Town's Science Faculty Reasearch Ethics Committee, which was granted. Clearance for the usage of PathCare data was granted by the PathCare Research Committee, following the submission of a project proposal, including proof of the ethics clearance sought from the aforementioned research ethics committee.

10 ACKNOWLEDGEMENTS

The author thanks A/Prof. Patrick Marais for his guidance and supervision while undertaking this research. Thanks are also extended to PathCare Laboratory services for their contribution of blood sample data, and particularly Dr. Ilse Louw and Marike Ubbink, who provided additional guidance and knowledge on the topic of malaria diagnosis.

REFERENCES

[1] Abdul-Nasir, A. S., Mashor, M. Y., and Mohamed, Z. Colour image segmentation approach for detection of malaria parasites using various colour models and k-means clustering. *WSEAS transactions on biology and biomedicine 10*, 1 (2013), 41–55.

[2] Altman, D. G., and Bland, J. M. Diagnostic tests. 1: Sensitivity and specificity. *BMJ: British Medical Journal 308*, 6943 (1994), 1552.

[3] Breiman, L. Bagging predictors. *Machine learning 24*, 2 (1996), 123–140.

[4] Breiman, L. Random forests. *Machine learning 45*, 1 (2001), 5–32.

[5] Chapelle, O., Haffner, P., and Vapnik, V. N. Support vector machines for histogram-based image classification. *IEEE transactions on Neural Networks 10*, 5 (1999), 1055–1064.

[6] Chaves-González, J. M., Vega-Rodríguez, M. A., Gómez-Pulido, J. A., and Sánchez-Pérez, J. M. Detecting skin in face recognition systems: A colour spaces study. *Digital Signal Processing 20*, 3 (2010), 806–823.

[7] Cortes, C., and Vapnik, V. Support-vector networks. *Machine learning 20*, 3 (1995), 273–297.

[8] Díaz, G., González, F. A., and Romero, E. A semi-automatic method for quantification and classification of erythrocytes infected with malaria parasites in microscopic images. *Journal of Biomedical Informatics 42*, 2 (2009), 296–307.

[9] Fanelli, G., Gall, J., and Van Gool, L. Real time head pose estimation with random regression forests. In *CVPR 2011* (2011), IEEE, pp. 617–624.

[10] Fukunaga, K., and Hayes, R. R. Estimation of classifier performance. *IEEE Transactions on Pattern Analysis and Machine Intelligence 11*, 10 (1989), 1087–1101.

[11] Haralick, R. M., Shanmugam, K., et al. Textural features for image classification. *IEEE Transactions on systems, man, and cybernetics*, 6 (1973), 610–621.

[12] Ho, T. K. Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition* (1995), vol. 1, IEEE, pp. 278–282.

[13] Hu, M.-K. Visual pattern recognition by moment invariants. *IRE transactions on information theory 8*, 2 (1962), 179–187.

[14] Kim, J.-H. Estimating classification error rate: Repeated cross-validation, repeated hold-out and bootstrap. *Computational statistics & data analysis 53*, 11 (2009), 3735–3745.

[15] Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (2012), pp. 1097–1105.

[16] Lalkhen, A. G., and McCluskey, A. Clinical tests: sensitivity and specificity. *Continuing Education in Anaesthesia Critical Care & Pain 8*, 6 (2008), 221–223.

[17] Lee, C.-H., Schmidt, M., Murtha, A., Bistritz, A., Sander, J., and Greiner, R. Segmenting brain tumors with conditional random fields and support vector machines. In *International Workshop on Computer Vision for Biomedical Image Applications* (2005), Springer, pp. 469–478.

[18] Lee, S. L. A., Kouzani, A. Z., and Hu, E. J. Random forest based lung nodule classification aided by clustering. *Computerized medical imaging and graphics 34*, 7 (2010), 535–542.

[19] Liang, Z., Powell, A., Ersoy, I., Poostchi, M., Silamut, K., Palaniappan, K., Guo, P., Hossain, M. A., Sameer, A., Maude, R. J., et al. Cnn-based image analysis for malaria diagnosis. In *2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)* (2016), IEEE, pp. 493–496.

[20] Mahmood, N. H., and Mansor, M. A. Red blood cells estimation using hough transform technique. *Signal & Image Processing 3*, 2 (2012), 53.

[21] Mazalan, S. M., Mahmood, N. H., and Razak, M. A. A. Automated red blood cells counting in peripheral blood smear image using circular hough transform. In *2013 1st International Conference on Artificial Intelligence, Modelling and Simulation* (2013), IEEE, pp. 320–324.

[22] Mehanian, C., Jaiswal, M., Delahunt, C., Thompson, C., Horning, M., Hu, L., Ostbye, T., McGuire, S., Mehanian, M., Champlin, C., et al. Computer-automated malaria diagnosis and quantitation using convolutional neural networks. In *Proceedings of the IEEE International Conference on Computer Vision* (2017), pp. 116–125.

[23] Mehrjou, A., Abbasian, T., and Izadi, M. Automatic malaria diagnosis system. In *2013 First RSI/ISM International Conference on Robotics and Mechatronics (ICRoM)* (2013), IEEE, pp. 205–211.

[24] Mery, D., Chanona-Pérez, J. J., Soto, A., Aguilera, J. M., Cipriano, A., Veléz-Rivera, N., Arzate-Vázquez, I., and Gutiérrez-López, G. F. Quality classification of corn tortillas using computer vision. *Journal of food engineering 101*, 4 (2010), 357–364.

[25] Otiniano-Rodriguez, K., Cámara-Chávez, G., and Menotti, D. Hu and zernike moments for sign language recognition. In *Proceedings of international conference on image processing, computer vision, and pattern recognition* (2012), pp. 1–5.

[26] Phu, J. J., and Tay, Y. H. Computer vision based hand gesture recognition using artificial neural network. *Faculty Of Information And Communication Technology, University Tunku Abdul Rahman (Utar), Malaysia* (2006).

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

[28] Rajaraman, S., Jaeger, S., and Antani, S. K. Performance evaluation of deep neural ensembles toward malaria parasite detection in thin-blood smear images. *PeerJ 7* (2019), e6977.

[29] Roula, M., Diamond, J., Bouridane, A., Miller, P., and Amira, A. A multispectral computer vision system for automatic grading of prostatic neoplasia. In *Proceedings IEEE International Symposium on Biomedical Imaging* (2002), IEEE, pp. 193–196.

[30] Russell, S. The economic burden of illness for households in developing countries: a review of studies focusing on malaria, tuberculosis, and human immunodeficiency virus/acquired immunodeficiency syndrome. *The American journal of tropical medicine and hygiene 71*, 2_suppl (2004), 147–155.

[31] Shillcutt, S., Morel, C., Goodman, C., Coleman, P., Bell, D., Whitty, C. J., and Mills, A. Cost-effectiveness of malaria diagnostic methods in sub-saharan africa in an era of combination therapy. *Bulletin of the World Health Organization 86* (2008), 101–110.

[32] Szummer, M., and Picard, R. W. Indoor-outdoor image classification. In *Proceedings 1998 IEEE International Workshop on Content-Based Access of Image and Video Database* (1998), IEEE, pp. 42–51.

[33] Tek, F. B., Dempster, A. G., and Kale, I. Parasite detection and identification for automated thin blood film malaria diagnosis. *Computer vision and image understanding 114*, 1 (2010), 21–32.

[34] Van Rossum, G., Warsaw, B., and Coghlan, N. Pep 8: style guide for python code. *Python. org 1565* (2001).

[35] Vese, L. A., and Chan, T. F. A multiphase level set framework for image segmentation using the mumford and shah model. *International journal of computer vision 50*, 3 (2002), 271–293.

[36] Vink, J., Laubscher, M., Vlutters, R., Silamut, K., Maude, R., Hasan, M., and De Haan, G. An automatic vision-based malaria diagnosis system. *Journal of microscopy 250*, 3 (2013), 166–178.

[37] WHO. *World malaria report 2018*. World Health Organization, 2018.

[38] WHO, and CDC. *Basic malaria microscopy*. World Health Organization, 2010.

[39] Yadav, S., and Shukla, S. Analysis of k-fold cross-validation over hold-out validation on colossal datasets for quality classification. In *2016 IEEE 6th International Conference on Advanced Computing (IACC)* (2016), IEEE, pp. 78–83.

[40] Zhang, Y., and Wu, L. Classification of fruits using computer vision and a multiclass support vector machine. *sensors 12*, 9 (2012), 12489–12505.

[41] Zhu, W., Zeng, N., Wang, N., et al. Sensitivity, specificity, accuracy, associated confidence interval and roc analysis with practical sas implementations. *NESUG proceedings: health care and life sciences, Baltimore, Maryland 19* (2010), 67.

SUPPLEMENTARY INFORMATION

This section contains additional tables of results which could not be included in the main paper due to space limitations.

	Grey RF	Grey SVM	Colour RF	Colour SVM
2 bin	0.5568	0.5923	0.8959	0.6970
4 bin	0.6159	0.6216	0.9289	0.7360
8 bin	0.6945	0.5901	0.9521	0.7686
16 bin	0.7760	0.6308	0.9528	0.8263
32 bin	0.8114	0.6564	0.9351	n/a
64 bin	0.8242	0.6686	n/a	n/a

Table S1: Results of histogram bin size selection process. Values are the mean accuracy score achieved, rounded to four decimal places. Maximum values for each column are highlighted.

Model	Features	Accuracy	Recall	Precision
RF	Haralick	0.7060	0.7462	0.7064
	Hu	0.5587	0.7054	0.5473
	Grey Hist	0.8228	0.8512	0.8064
	Colour Hist	0.9515	0.9574	0.9462
SVM	Haralick	0.6964	0.6258	0.8071
	Hu	0.5939	0.5111	0.6632
	Grey Hist	0.6715	0.6314	0.6885
	Colour Hist	0.8182	0.7559	0.8646

Table S2: Results from the feature extraction selection process. The best results for each model are highlighted.

Features	Filtering	Accuracy	Recall	Precision
Haralick	None	0.7242	0.8033	0.7005
	H	0.9159	0.9186	0.9160
	H + IS	0.9095	0.9080	0.9133
	H + IS + C	0.7845	0.7227	0.8271
	H + IS + T	0.9291	0.9376	0.9219
Grey Hist	None	0.8222	0.8521	0.8043
	H	0.8741	0.8910	0.8619
	H + IS	0.9598	0.9644	0.9556
	H + IS + C	0.9376	0.9646	0.9153
	H + IS + T	0.9223	0.9328	0.9136
Colour Hist	None	0.9498	0.9564	0.9439
	H	0.9525	0.9606	0.9453

Table S3: Results from the RF image filtering selection process. The best results for each feature extraction approach are highlighted.

Features	Filtering	Accuracy	Recall	Precision
Haralick	None	0.7615	0.7900	0.7741
	H	0.9350	0.9055	0.9632
	H + IS	0.9090	0.9214	0.9032
	H + IS + C	0.8466	0.8586	0.8414
	H + IS + T	0.9295	0.9574	0.9068
Grey Hist	None	0.6645	0.6422	0.6733
	H	0.7058	0.7368	0.6939
	H + IS	0.7810	0.6592	0.8730
	H + IS + C	0.7758	0.7021	0.8244
	H + IS + T	0.6486	0.3481	0.8727
Colour Hist	None	0.8160	0.7374	0.8753
	H	0.7744	0.7126	0.8136

Table S4: Results from the SVM image filtering selection process. The best results for each feature extraction approach are highlighted.

Forest Size	Accuracy	Recall	Precision
1	0.9274	0.9228	0.9314
10	0.9594	0.9595	0.9594
50	0.9610	0.9587	0.9631
100	0.9618	0.9596	0.9647
250	0.9615	0.9594	0.9644

Table S5: Results of the RF forest size selection process. The best result for each column is highlighted.

Max Depth	Accuracy	Recall	Precision
None	0.9658	0.9633	0.9682
10	0.9624	0.9569	0.9675
100	0.9657	0.9629	0.9683
250	0.9654	0.9630	0.9677
1000	0.9654	0.9638	0.9669

Table S6: Results of the RF maximum depth selection process. The best result for each column is highlighted.

Kernel Function	Accuracy	Recall	Precision
RBF	0.9287	0.9558	0.9067
1st degree polynomial	0.9281	0.9519	0.9087
2nd degree polynomial	0.9288	0.9560	0.9067
3rd degree polynomial	0.9290	0.9566	0.9066
5th degree polynomial	0.9288	0.9568	0.9062

Table S7: Results of the SVM kernel selection process. The best result for each column is highlighted.

Gamma	Accuracy	Recall	Precision
Auto	0.9307	0.9636	0.9041
1	0.9305	0.9631	0.9041
0.1	0.9309	0.9585	0.9084
0.01	0.9301	0.9526	0.9116
0.001	0.9301	0.9526	0.9116

Table S8: Results of the SVM gamma hyperparameter selection process. The best result for each column is highlighted.