

Article

Mobile-Aware Deep Learning Algorithms for Malaria Parasites and White Blood Cells Localization in Thick Blood Smears

Rose Nakasi ^{1,*}, Ernest Mwebaze ² and Aminah Zawedde ³

¹ Department of Computer Science, College of Computing and Information Sciences, Makerere University, Kampala P.O. Box 7062, Uganda

² Sunbird AI, Kampala P.O. Box 11296, Uganda; emwebaze@sunbird.ai

³ Department of Information Technology, College of Computing and Information Sciences, Makerere University, Kampala P.O. Box 7062, Uganda; sazawedde@gmail.com

* Correspondence: g.nakasirose@gmail.com

Abstract: Effective determination of malaria parasitemia is paramount in aiding clinicians to accurately estimate the severity of malaria and guide the response for quality treatment. Microscopy by thick smear blood films is the conventional method for malaria parasitemia determination. Despite its edge over other existing methods of malaria parasitemia determination, it has been critiqued for being laborious, time consuming and equally requires expert knowledge for an efficient manual quantification of the parasitemia. This poses a big challenge to most low developing countries as they are not only highly endemic but equally low resourced in terms of technical personnel in medical laboratories. This study presents an end-to-end deep learning approach to automate the localization and count of *P.falciparum* parasites and White Blood Cells (WBCs) for effective parasitemia determination. The method involved building computer vision models on a dataset of annotated thick blood smear images. These computer vision models were built based on pre-trained deep learning models including Faster Regional Convolutional Neural Network (Faster R-CNN) and Single Shot Multibox Detector (SSD) models that help process the obtained digital images. To improve model performance due to a limited dataset, data augmentation was applied. Results from the evaluation of our approach showed that it reliably detected and returned a count of parasites and WBCs with good precision and recall. A strong correlation was observed between our model-generated counts and the manual counts done by microscopy experts (posting a spear man correlation of $\rho = 0.998$ for parasites and $\rho = 0.987$ for WBCs). Additionally, our proposed SSD model was quantized and deployed on a mobile smartphone-based inference app to detect malaria parasites and WBCs in situ. Our proposed method can be applied to support malaria diagnostics in settings with few trained Microscopy Experts yet constrained with large volume of patients to diagnose.

Keywords: deep learning; thick blood smear; parasite count; white blood cell count; parasitemia determination; Faster R-CNN; SSD MobileNet; smartphone deployment



Citation: Nakasi, R.; Mwebaze, E.; Zawedde, A. Mobile-Aware Deep Learning Algorithms for Malaria Parasites and White Blood Cells Localization in Thick Blood Smears. *Algorithms* **2021**, *14*, 17. <https://doi.org/10.3390/a14010017>

Received: 1 December 2020

Accepted: 5 January 2021

Published: 11 January 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Malaria is still a global health concern with nearly half of the world's population at risk. Two hundred and twenty-eight million cases were reported in 2018, and the estimated number of malaria deaths stood at 405,000 [1]. These statistics are majorly a reflection of cases in sub-Saharan Africa, but other regions, including South-East Asia, Eastern Mediterranean, Western Pacific, and United States of America, are also at risk.

When diagnosing malaria, the quantitative content of malaria parasites in the blood also referred to as malaria parasitemia is key as it plays a pivotal role in decision support for clinicians with respect to the severity of the malaria, choice of treatment, and probable cause of the disease [2]. However, in highly endemic but low resourced regions, like sub-Saharan Africa, there is lack of sufficient skilled human capacity to efficiently determine malaria parasitemia density of patients. The situation is not made any better by

the existing conventional methods for parasitemia determination, like microscopy. This de facto gold standard method for malaria detection and parasitemia determination in most endemic countries involves the manual count of parasites (trophozoites) and White Blood Cells (WBCs) (see in Figure 1) by expert Microscopists [3] (as shown in Figure 1). Counting malaria parasites and WBCs by microscopy method is not only laborious and time consuming but also susceptible to subjectivity by Microscopists [4].

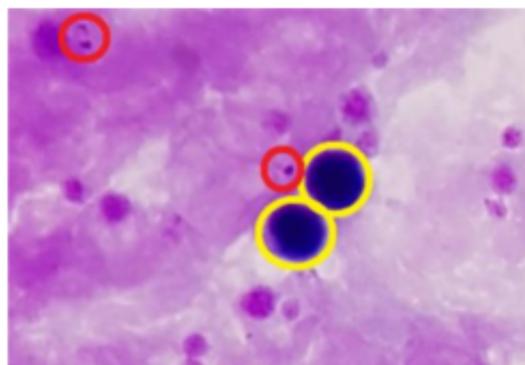


Figure 1. Thick blood smear. Red circles are trophozoites and yellow circles are White Blood Cells (WBCs) [5].

In view of the above microscopy challenges, this research sought to develop an efficient computer vision automated system for identification and quantification of malaria parasitemia and couple the solution with a real time mobile smart phone localization app for malaria diagnosis. Malaria diagnosis automation using a mobile smart phone app for parasitemia determination improves reliability and exactness in interpreting malaria disease severity. It also increases throughput especially in highly endemic but low resourced developing countries.

Current advancements in technologies have made it possible to change the way microscopy of malaria is performed in developing countries. New technologies, like deep learning, can be leveraged with a combination of smart phones to improve disease diagnosis. Given that mobile smartphones are widely owned across the developing world, there is a technological opportunity to address malaria microscopy. Mobile smart phones are now equipped with digital cameras and increased computational power which affords them the potential to capture and process microscopy images. This process is reinforced using pre-trained deep learning models based on transfer learning. Since medical image datasets are usually smaller in size often termed inadequate for learning [6,7], its important to use pre-trained deep learning as a way of utilizing small datasets to produce results that would otherwise be produced by a big dataset. Specifically, transfer learning focuses on sharing knowledge gained while solving one task and applying it to a different but related problem [8]. The learned features can serve as initialization for the another model. In transfer learning, representations learned on a large dataset can be transferred to a model with a totally different yet small data domain [9,10]. Transferring such knowledge from already trained models using big datasets makes computational training of small datasets faster and efficient. Since in this study we use a smaller dataset, the would be challenges are hence cured by using the deep learning models where our small dataset is trained through transfer learning by pre-trained models on already existing big datasets, such as Common Objects in Context (COCO) [11].

In this study, therefore, we propose the use of novel pre-trained deep learning models for a multi-class detection task and Mobile smart phone app for automated localization and quantification of malaria density in thick blood smears. Due to limited datasets, we adopted pre-trained models of Faster Regional Convolutional Neural Network (Faster R-CNN) and Single Shot Multibox Detector (SSD) MobileNet. The Faster Regional Convolutional Neural Network (Faster R-CNN) is known for high accuracy and the Single Shot Multibox

Detector (SSD) MobileNet is easily deployable on low core devices, like mobile phones, which we actually employ in our model. A combination of the two methods, therefore, is able to grant us high efficiency and easy deployment on mobile smartphone.

The major contributions of this study are threefold:

We demonstrate that the use of pre-trained deep learning models, particularly Faster R-CNN and SSD MobileNet, in a multi-class detection task and localization of both malaria parasites and WBCs in thick blood smears is efficient. Under this contribution, we extend to validate and compare model performance on a held out dataset with expert level detection performance.

Secondly, we extend the image analysis task to a count of parasites and WBCs and provide an end to end approach for determination of malaria parasitemia based on thick blood smears.

Thirdly, we develop a prototype low core device deployment of image analysis on a mobile smartphone for localization of malaria parasites and WBCs. We believe that our proposed approach can relieve the few available skilled Microscopists the burden of manually counting parasites under the conventional method of malaria microscopy a method that has been faulted for being tedious, time-consuming, subjective and error-prone.

The rest of the paper is organized as follows. In Section 2, we present the related work. In Section 3, details of our proposed method are presented. Experimental results, as well as the discussion, are elaborated in Section 4. In Section 5, conclusions of the study and future work are presented.

2. Related Work

Previous studies have attempted to address the problem of parasitemia estimation for example Sio et al. [12] presented a software, MalariaCount, that automatically generates parasitemias from images of thin Giemsa-stained blood smears. In a subsequent effort, Savkare and Narote [13], presented a technique for determining malaria parasitemia based on a count of Red Blood Cells (RBCs) through a classification task of parasitized and non parasitized cells. They both applied the Otsu thresholding technique on gray image and green channel of the blood image for cell segmentation. Here, watershed-transform was used for separation of touching cells and color. Statistical features were then extracted from segmented cells and SVM binary classifier was used for classification of normal and infected cells. Poostchi et al. [14] proposed an end-to-end automated detection system for identifying and quantifying malaria parasites (*P. falciparum*) in thin blood smears of both human and mouse. The authors use a combination of color and texture features to characterize segmented RBCs and a linear Support Vector Machine (SVM) model to classify infected and uninfected cells. These studies have mainly focused on thin blood smear images [15]. However, according to WHO, thick blood films are the gold standard for the determination of malaria parasitemia [16] and provide better detection of parasites than thin blood smears [14]. Thick blood smears contain several layers of red cells, whereas thin films contain a single layer of spread red blood cells. Therefore, for a fixed number of microscope fields, thick films allow the microscopist to examine a larger number of red cells for the presence of parasites, and consequently low parasitemia can be readily identified in thick films [17].

Frean [18] used a semi-automated method for parasitemia determination in thick smears by using an open source java based image processing program ImageJ (version 1.41) for image analysis. In essence, the program segments or classifies particles to be counted on the basis of their relative density (darkness) compared with the background via a thresholding process. For this study, conventional statistical evaluations were done using Statistica 8.0 [18]. The image analysis software used was based on morphological feature extraction with no detection specificity reported. The authors enumerated parasites per image manually by counting parasites on the captured images.

Currently, deep learning models come with an added advantage in that they do not require expertise for the hand engineered features [19]. They have, therefore, aided devel-

opment of systems for medical image analysis [20]. Yang et al. [5] developed a smartphone application based on intensity-based Iterative Global Minimum Screening (IGMS). The researchers use this method for automatic pre-selection of malaria parasites in thick blood smears and a customized Convolutional Neural Network (CNN) classification model of the parasite. In a separate experiment, the authors localized WBCs using segmentation. However, Customized CNN models have been associated with back and forth fine-tuning of the model layers and huge volumes of annotated data for improved accuracy which may not be available for medical image analysis [6,21]. Moreover, the authors implemented a single-class detection task for mobile deployment.

As opposed to semi-automated and customized CNN models for classification of tasks on malaria parasitemia, this study presents an end-to-end automated pre-trained deep learning approach for detection and counting of malaria parasites and WBCs for improved parasitemia determination in thick blood smears. Unlike the previous study that focused on a single class object detection task for malaria parasites in thick blood smear [22], the pre-trained deep learning approach utilized in this study is a multi-class detection task for parasites and WBCs. It is significantly faster, efficient and requires no additional operator input or pre-processing of the images for automated detection and counts. This is well suited for improving malaria parasitemia quantification in resource-constrained settings with few skilled lab technologists to interpret microscopy test results, as well as facilitating training of the models in a low data regime.

3. Materials and Methods

In order to accomplish our proposed task for automated malaria parasitemia task, we designed an end-to-end pipeline for achieving automated localization of trophozoites and WBCs in thick blood smear images as shown in Figure 2. We precisely present the data preparation and labeling procedure as elaborated in Section 3.1, we present the deep learning approach and discuss our choice for the approach in achieving our object detection task in Section 3.2, and then we elaborate the training environment of the selected pre-trained deep learning meta-architecture in Section 3.3. In Section 3.4, we provide model evaluation using standard object detection evaluation matrices of mAP (mean Average Precision), precision, recall, and F1 score. The final objects of interest out of our object detection experiments are localizations of trophozoites and WBCs as elucidated in Section 4. SSD model deployment is extended for mobile phone-based inference, as presented in Section 4.1.

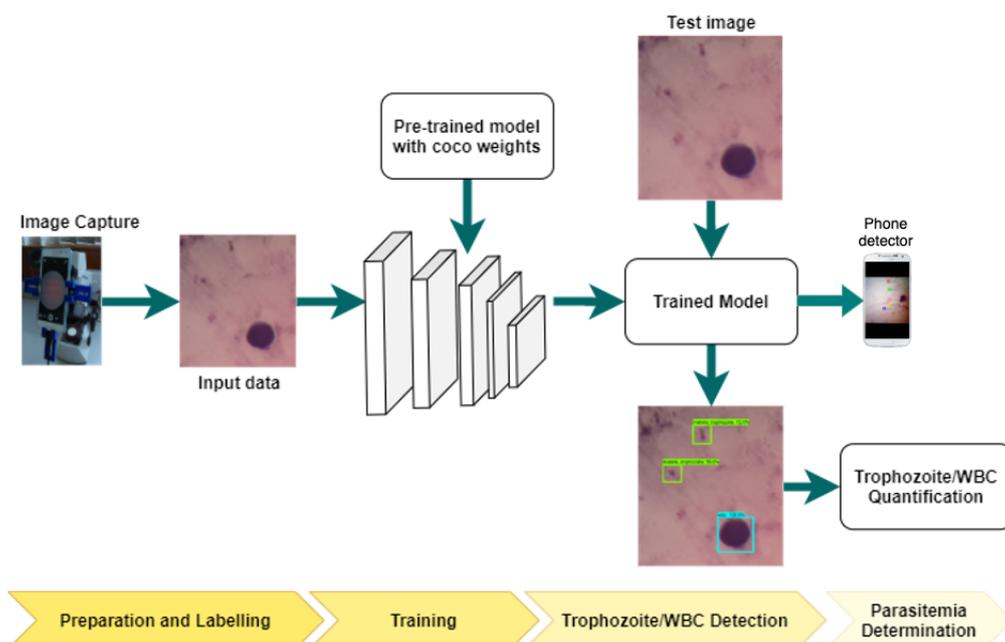


Figure 2. Pipeline for our automated mobile-aware localization of malaria trophozoites and White Blood Cells (WBCs).

3.1. Data Acquisition and Preparation

To accomplish the localization task in this study, 903 images of thick blood smear with field stain were collected from Mulago National referral hospital in Uganda. Before collection of the image samples, proper permissions were obtained from the people providing the samples, as well as the hospital institution for this data collection. All the data collected was treated in line with the ethical considerations and guidelines governing medical data. Images were captured using a smartphone camera attached to an Olympus microscope with objective magnification ($1000\times$). The attachment mechanism of the smartphone to the eyepiece of the microscope was supported by a 3-D printable adapter [23]. For this experiment, images were taken using a Samsung J6 Android smart-phone camera with a resolution of 5 megapixels to producing images with a dimension of 3264×2448 pixels. For each blood slide, the microscope was adjusted to obtain different view point images. Each image was obtained with the corresponding setup parameters including the microscope slide number, stage micrometer grid readings (x and y), zoom level of the smartphone used for image collection, objective size of the microscope, and the staining reagent used. Images of thick blood smears captured with the phone and uploaded to the server, were downloaded and manual data annotations were performed by experienced Lab technicians using the annotation tool LabelImg [24]. The Annotations were in form of bounding boxes drawn around the parasites and WBCs in an image of a blood slide. These annotations were then saved in the Pascal VOC (Visual Object Classes) format [25] and they formed the base data input to our selected models. This Dataset is available at (<https://drive.google.com/drive/folders/1p45Dt-BJy8hhoI-rYnhcaL6IMl5FsFL-?usp=sharing>).

Preparation of our input data for our selected data analysis models involved decoding of the images and their corresponding XML file annotations from the base dataset into TF (Tensorflow) Recordfiles. The *TFRecord* record is the data input format for a training job optimized for utilizing the TensorFlow Object detection API (Application Programming Interface). To train and evaluate of our models, the dataset was randomly split into a 9:1 train and test set.

The data pre-processing procedure included conversion of malaria trophozoite and WBC labels to TF-Record format [26] that is compatible with the TensorFlow framework.

3.2. Deep Learning Approach to Malaria Trophozoite and WBC Localization

In medical image analysis, Deep learning has particularly become a choice for methodology in different health fields [20]. Unlike conventional machine learning, deep learning does not require expertise for hand crafted features. Deep learning comprises of multiple processing layers to automatically learn good data representations with multiple levels of abstraction. Therefore, deep learning is promising in a wide variety of applications including malaria parasite detection and detection of WBCs for automated parasitemia determination. This can thus compensate for the limited skilled Microscopists in low but highly endemic settings.

In this study, we employed CNN architecture called Faster R-CNN [27] model and SSD model [28] for the task of localization, detection and counting of the malaria trophozoites and WBCs in thick blood smears. Faster R-CNN has displayed good performance in different object detection tasks [29–31], including a similar study for detection of malaria parasites [22], while SSD MobileNet provides opportunities for mobile smartphone deployment, a feature that was important for the choice of this model.

Faster R-CNN, is an improvement to the Region based CNN model (R-CNN) whose detection procedure occurs over two stages. The first stage uses Region Proposal Networks (RPNs) to extract features using selective search purposefully to generate proposals (boxes) which constitute feature maps that are later used by the neural network to check the occurrence of objects. The idea is to reduce the search for relevant features related to the objects of interest in the dataset [27]. In the second stage, bounding box proposals obtained in the previous stage are further cropped and processed. This combination helps Faster R-CNN to have a leading performance on accuracy though leads to a two-stage

architecture network which reduces the speed of processing of this method [32]. In this study, we thus implement fully convolutional Residual Network(ResNet 101) [33] as the backbone network for Faster R-CNN. Unlike training from scratch, we exploited transfer learning by using the Faster R-CNN ResNet 101 model pre-trained on the COCO (Common Objects in Context) dataset [11] to achieve faster training coupled with high accuracy.

Single Shot Multibox Detector(SSD) [28] was also implemented in our experimentation. SSD relies on a feed-forward convolutional network to produce a fixed-size collection of bounding boxes and scores for object presence in those boxes. Finally, through a non-maximum suppression step, detections are produced. The Single Shot Multibox Detector (SSD) MobileNet architecture model is generated from depth-wise convolution. A single filter is applied to each neural input channel to begin feature extraction, a 1×1 point-wise convolution combines the outputs of the depth-wise convolution. Resultant from a depth-wise convolution are two layers generated, a separate layer for filtration and a layer for combination. According to Howard [34], the combination of the two layers minimizes the model size to create room for reducing computational power demands. Furthermore, model efficiency is enhanced through depth-wise convolution by preventing Graphical Processing Unit (GPU) over consumption on less demanding devices such as smartphone platforms. However, the low GPU consumption also creates a lack of usage equilibrium, which hinders the training model causing slow progress and intervals. In this study, we implemented SSD MobileNet V2 model because of its efficiency for low core deployment on mobile smartphones.

3.3. Training Approach

All the proposed models in our experiments were trained and tested on Ubuntu system with a 5th Gen Intel Core processor i7 and 16 GB RAM, a GPU-enabled Nvidia GTX 1060 Graphical Processing Unit (GPU) with 6 GB RAM, Python 3.5, with Tensorflow back-end. Training the models was done using the open-source library Tensorflow [26] on a TFrecord generated from a training dataset of 803 images. To train and test model, we employed the implementation of Faster R-CNN ResNet 101 and SSD MobileNet V2 provided by the Google's open-source Tensorflow object detection API [35]. Following a transfer learning Scheme, pre-trained models were used to initialize the weights of the previously learned model to adapt to a new task [36]. Transfer learning has the benefit of decreasing the training time for a neural network model and can result in lower generalization error. Faster-R-CNN and SSD models used were pre-trained on the Microsoft COCO [11] dataset that has a lot more labeled data than our limited size of dataset. In our experiments, to fine-tune and share the connection weights between a previously learned model on COCO dataset to our dataset, all layers except the fully connected layer were re-trained without freezing and the weights of the pre-trained models were used as initial weights.

To increase the sample size and variation, data augmentation was applied to optimize and obtain better performance. Augmentation of training data was achieved through random vertical and horizontal flipping of images through the TensorFlow object detection API.

Additionally, the TensorFlow Object detection API enables specification of different training options and parameters by editing a configuration file. Table 1 shows our hyper-parameter settings. We set a small learning rate (lr) to work well with the other selected hyper-parameters. A base lr was kept very low at 0.0003 and was later reduced to 0.00004 for Faster R-CNN and SSD, respectively.

Table 1. Training parameters used for each of the models.

Parameter	Faster R-CNN ResNet101	SSD Mobilenet
Learning rate	0.0003	0.0004
Batch size	1	12
Input image size (pixels)	750 × 750	300 × 300
Momentum optimizer	0.9	0.9
<i>IoU</i> threshold	0.5	0.5

For training of machine learning models, having a large batch size like one we used for SSD MobileNet (light model) is usually desirable, although, for experiments using Faster R-CNN, a batch size of 1 was used. This was inevitably due to model's complex architecture and the limited memory resources that were available for us to execute the training job. Subsequently, We chose momentum optimizer over other optimization algorithms because it helps move more quickly towards the minimum, builds speed, and quickens convergence [37]. To avoid fluctuations associated with smaller optimizer beta values (less than 0.5), we used 0.9 beta value for the momentum optimizer to achieve a more smoother curve.

To fine-tune the models to detect the spatial features of the objects of interest (parasites and WBCs), all models were trained for 200,000 time steps though each model had its own level of attained optimal performance (lowest loss and highest mAP) at respective steps (see Table 2). These are steps at which the model converges with early stopping. It was realized that SSD MobileNet trains for fewer steps in comparison to Faster R-CNN model. SSD is simpler than other networks as it performs all computations in a single network. SSD combines the predictions from numerous feature maps having different resolutions to handle objects of various sizes. It does not involve regional proposal generating or feature re-sampling, like previous networks did. This makes it easy to train and integrate into systems where detection is required [38]. The training processes were monitored on TensorFlow's visualization toolkit, TensorBoard to understand all variations that occurred during training.

Table 2. mAP@0.5, precision, and recall performance for Faster Regional Convolutional Neural Network (R-CNN) model for detection of malaria parasites and WBC.

Algorithm	Steps	mAP@0.5	Precision	Recall	F1 score
Faster R-CNN (trophozoites_only)	138,700	0.5506	0.6720	0.802	0.7312
Faster R-CNN (WBC_only)	42,800	0.892	0.855	0.959	0.9039
Faster R-CNN (trophozoites + WBC)	98,700	0.6609	parasites: 0.686 WBC: 0.805	parasites: 0.9303 WBC: 0.884	parasites: 0.7897 WBC: 0.8426
SSD MobileNet (trophozoites + WBC)	80,000	0.6292	parasites: 0.760 WBC: 0.89	parasites: 0.501 WBC: 0.806	parasites: 0.6038 WBC: 0.7528

Once training was accomplished, a set of checkpoint files comprising of learned features were generated from the training dataset. The checkpoint files were frozen into a protobuff file (a Tensorflow format for standard representation of machine learning models).

Using the same setup, different experiments were conducted to investigate model performance with respect to a single-class object detection task (that is, only malaria trophozoites detection and only WBCs detection) and multi-class object detection (both malaria trophozoites and WBCs).

3.4. Model Evaluation

The study evaluated the trained models on a validation dataset of 100 thick blood smear images. To clearly understand how well a model performed in accomplishing the goal of counting trophozoites and WBCs on any validation set image, the TensorFlow object detection API was used. It handled the evaluation of a trained model using the standard COCO evaluation metrics [11]. We adopted the standard evaluation metrics for an object detection task. Specifically, the common metrics of mAP (mean Average Precision) [39], precision, recall, and F1 Score were used.

In order to use mAP [39] as a standard evaluation metric for the object detection task, the performance was first evaluated in terms of Intersection-over-Union (*IoU*) as implemented in the Pascal VOC Challenge [40]. The precision of a single detection can be measured using *IoU* as defined in Equation (1).

$$IoU(A, B) = \left| \frac{A \cup B}{A \cap B} \right|, \quad (1)$$

where A represents the ground-truth box collected in the annotation, and B represents the predicted result of the transfer learning model. The *IoU* measures the percentage or ratio of overlap of the predicted bounding box to the ground truth boxes [39]. mAP is calculated by getting the average *IoU* for all detections made and in our experiments, we considered an *IoU* threshold of 0.5. The lower bound of the *IoU* threshold (0.5) reflects the detection accuracy based on bounding box coverage characteristics of each detected object of interest.

For a single class in a single image, *Precision* is a measure of the ratio of the true positives detected by the model to the total number of detections made as defined in Equation (2):

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

where TP (True Positives) represents the number of true object detections, and FP (False Positives) represents wrongly detected objects. On the other hand, *Recall* measures how many true detections were identified by the model as defined in Equation (3):

$$Recall = \frac{TP}{TP + FN}, \quad (3)$$

where TP (True Positives) represent Number of true object detections, and FN (False Negatives) represent true undetected objects.

In this study, *F1 Score* was also used to provide a better understanding of the overall model performance for different experiments conducted. *Precision* and *Recall* can be combined to generate the *F1 Score* as defined in Equation (4):

$$F1Score = \frac{2 \times (Precision \times Recall)}{Precision + Recall}. \quad (4)$$

4. Results

Results from our training experiments show that initially, mAP increases fast and reaches an optimum for the number of steps different for each model as indicated in Table 2. Figures 3 and 4 present the graphical representation of the resultant mAP and loss curves of Faster R-CNN and SSD MobileNet.



Figure 3. mAP graph and the corresponding training loss graph for Faster R-CNN. The model learns the data while achieving a high accuracy at about 98,700 iterations.

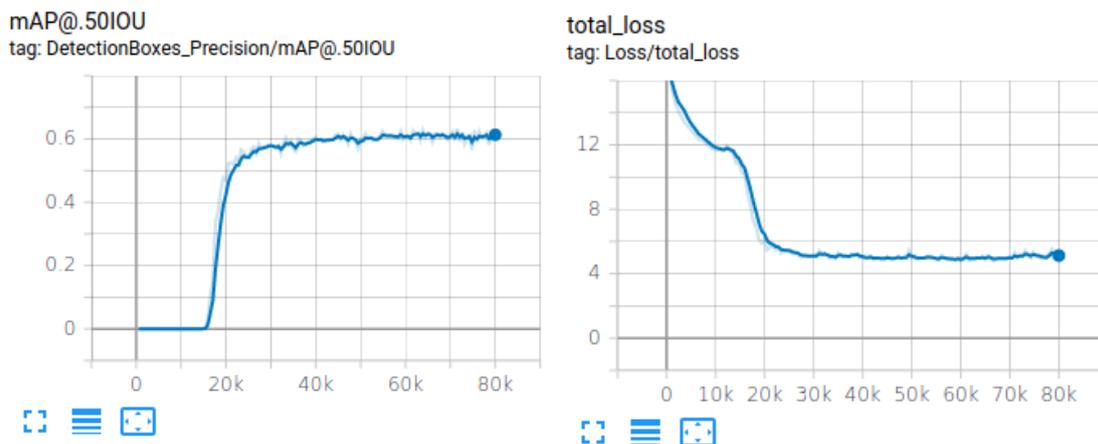


Figure 4. mAP graph and the corresponding training loss graph for Single Shot Multibox Detector (SSD) MobileNet. The model learns the data, while achieving a high accurac at about 80,000 iterations.

Figures 3 and 4 show that Faster R-CNN model achieves better accuracy (in terms of mAP and F1 score) than SSD model for localizing both parasites and WBCs as expected [41]. To visualize the results, model performance inference detections on a sample test image are shown in Figure 5. Again, We observed from the depiction that Faster R-CNN (b) performs better than SDD MobileNet model (SSD (c)).

A general representation of our experimental results is summarized in Table 2.

Table 2 shows that a multi-class detection task of both malaria trophozoites (parasites) and WBCs produces better results than a single-class task (trophozoites only), but lower than a single-class of WBCs (mAP = 0.892). This is attributed to the fact that WBCs are bigger objects and as such are easier examples to learn as compared to smaller objects (trophozoites only). The better performance of the multi-class task is attributed to having bigger objects complementing small objects (trophozoites). The multi-class mAP value of 0.6609 (@0.5IOU) was considered to be good enough performance because of the presence of small objects(trophozoites).

We notice that the recall value generated by SSD MobileNet is lower than that of Faster R-CNN to a great extent. This could be attributed to the fact that using more complex (two-stage) object detectors, such as Faster R-CNN, for difficult small images(parasites), provides better optimization than less complex (single-stage) detectors (SSD) [42]. We show that Faster R-CNN ResNet 101 model produces better detection accuracy for our proposed multi-class detection task for trophozoites (F1 score = 0.7897) and WBCs (F1 score = 0.8426)

than its variant SSD MobileNet v2 model for detection of trophozoites (F1 score = 0.6038) and WBCs (F1 score = 0.7528).

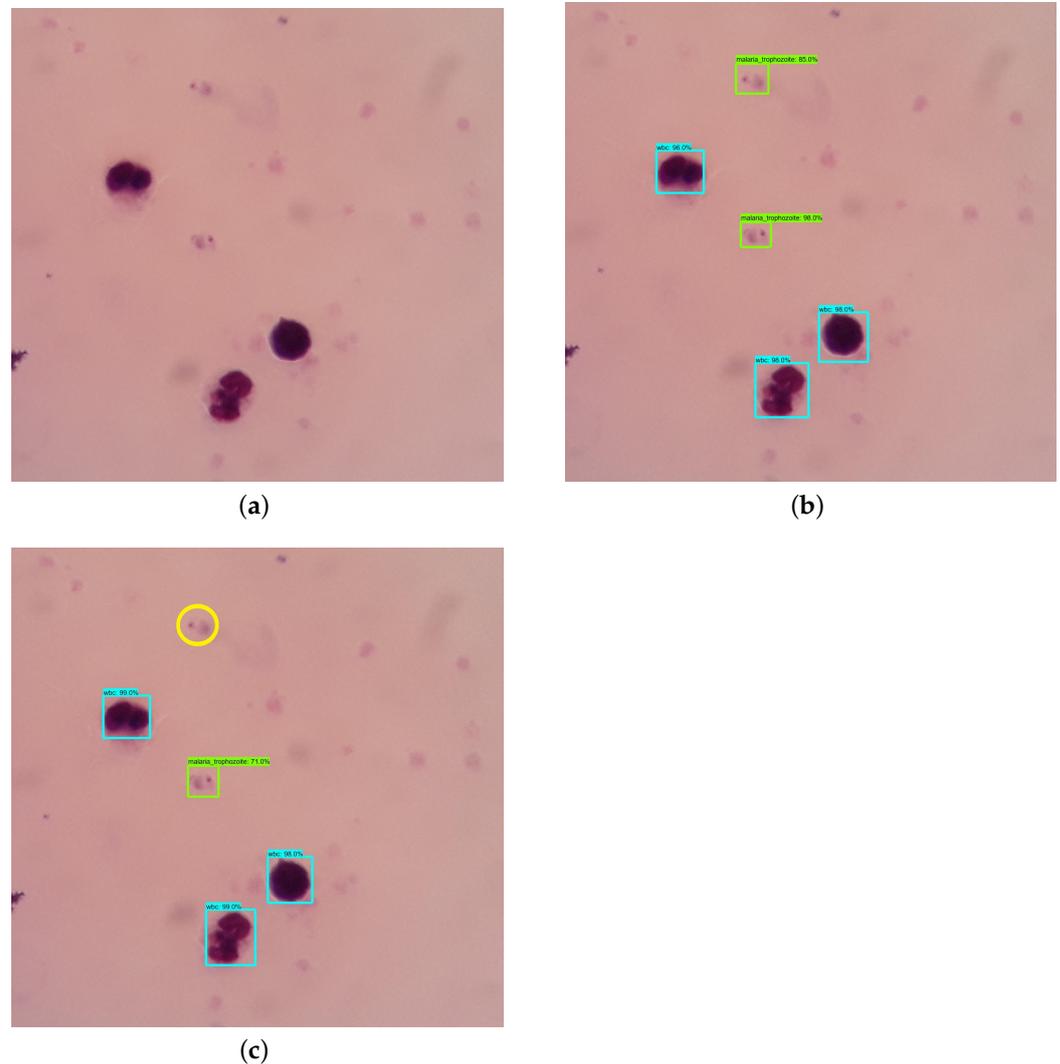


Figure 5. Results for detection with Faster R-CNN (b) and SSD (c) for a sample image (a). The blue rectangles indicate WBCs, and the green rectangles indicate malaria trophozoites. The yellow circle indicates a False Negative.

Furthermore, in Figure 6, Faster R-CNN model-generated parasite counts are higher than both SSD and manual-expert counts, which shows that Faster R-CNN models perform better at identification of parasites and WBCs in comparison to SSD and experts. The performance disparity with respect to the manual expert count could potentially signify the effect of false-positives through low quality annotations, thus a lower precision registered for parasites. We next analyze the magnitude of the disparity between Faster R-CNN and manual expert count. Results generated show that Faster R-CNN model-generated counts correlate well with manual-expert counts on a held-out dataset of 8 thick blood smear films with more than 100 images each (high Spearman's correlation coefficients for trophozoites ($\rho = 0.998$) and WBCs ($\rho = 0.987$)), as presented in Figure 7. In general, the precision, recall, and F1 score of Faster R-CNN multi-class detection task were comparably higher than SSD.

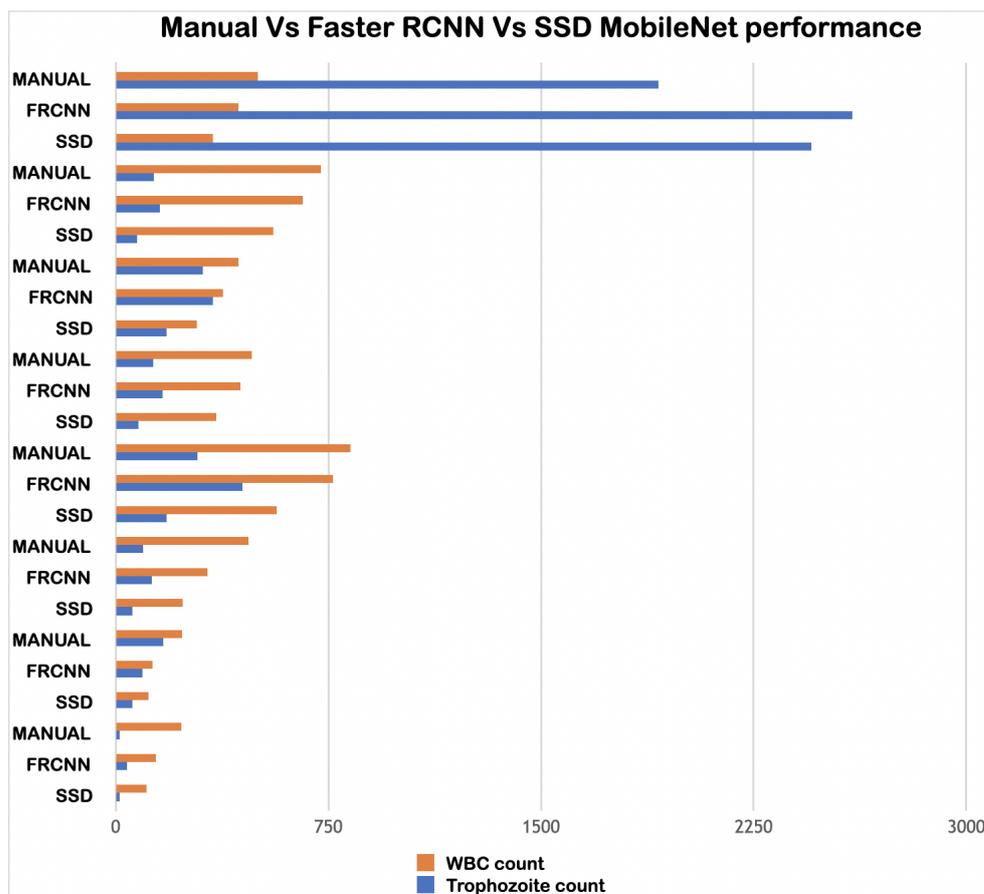


Figure 6. Manual experts vs Faster R-CNN model vs SSD model counts per blood film.

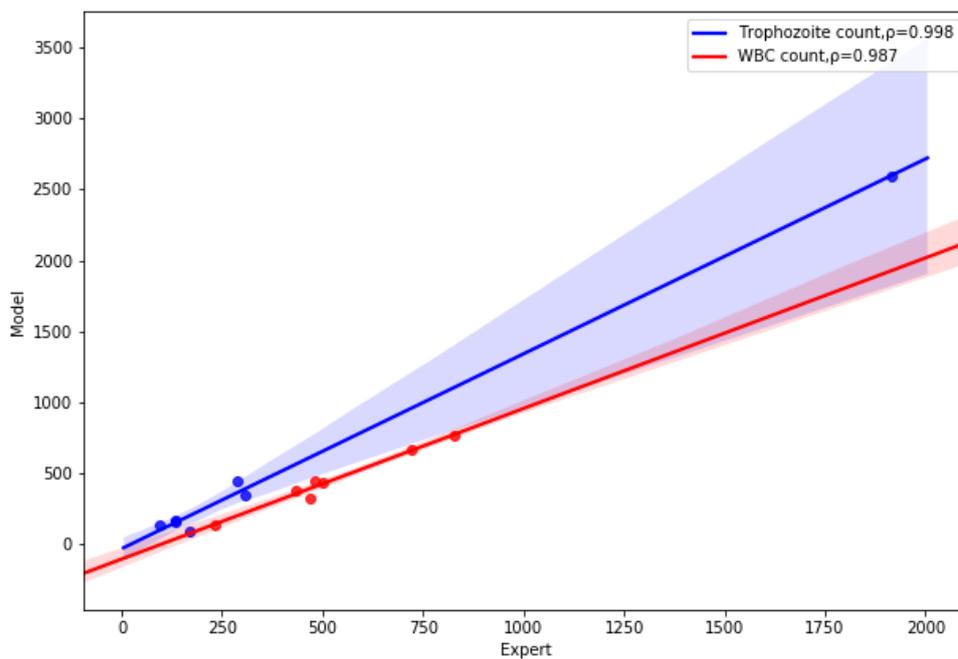


Figure 7. Correlation between model counts and manual expert counts per film.

Following from Table 3, film-wise parasitemia density based on *Parasites/μL* method per film [43,44] was evaluated through the Equation (5):

$$\text{Parasites}/\mu\text{L} = \frac{\text{No. of counted parasites} \times \text{No. of counted WBC}}{\text{No. of assumed WBC (8000)}} \quad (5)$$

Malaria parasitemia clinical interpretations [45] were deduced as shown in Table 3. Slides 1, 2, 3, 4, 5, 6, and 7 represent parasitemia level above which immune patients will exhibit malaria symptoms whereas Slide 8 represents a patient with maximum parasitemia.

Table 3. Malaria parasitemia on a unique set of each validation thick blood smear slide with corresponding clinical interpretation.

Slide	Faster R-CNN Parasites/μL	SSD Parasites/μL	Manual Parasites/μL
Slide ₁	2326	1028	485
Slide ₂	5740	4174	5768
Slide ₃	3160	1958	1638
Slide ₄	4682	2535	2783
Slide ₅	3011	1808	2195
Slide ₆	7238	5007	5685
Slide ₇	1906	1081	1483
Slide ₈	47,982	57,404	30,502

4.1. Deployment of SSD MobileNet V2 on a Smartphone

It should be noted that although the mAP, precision, recall, and F1 score values are lower than that of Faster R-CNN model, a quantized SSD MobileNet model can easily be deployed on low cost device operating systems, such as Android. Quantization of the model reduces the size of the model through model compression to allow models run faster on low cost devices. This, however, comes with a trade-off in performance.

The TensorFlow Lite converter (see Figure 8), a Python API was used to convert our pre-trained TensorFlow model (.pb file format) into the TensorFlow Lite format (.tflite). TensorFlow Lite has capability for achieving low latency, optimizing the kernels for mobile deployment and pre-fused activation to allow smaller and faster models [46].

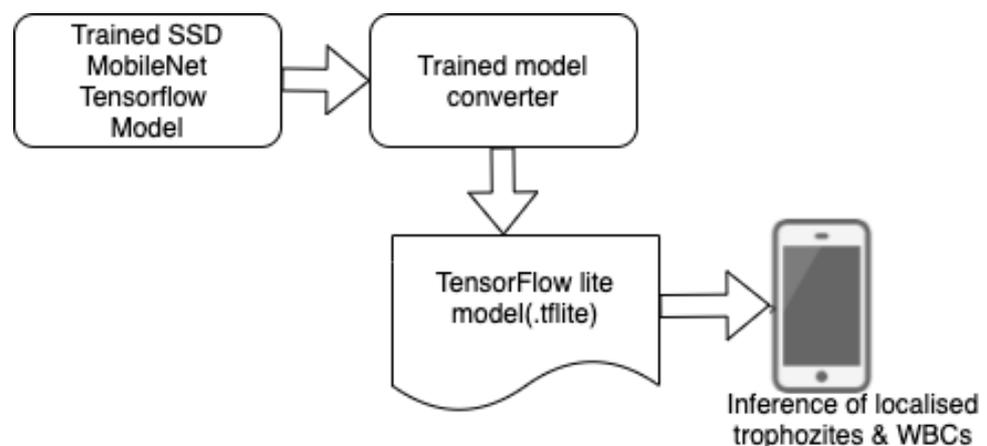


Figure 8. Pipeline for model conversion and integration of pre-trained model for mobile deployment.

The application was developed in Flutter using the tflite package and a pre-trained quantized SSD-MobileNet model for localizing malaria parasites and WBCs in images and real-time camera streams. Images can be taken from within the app and are then fed to the detection model.

The image-picker was used for picking images from the gallery, tflite for running our model, and Flutter's camera package for controlling the camera. Once the app is launched, camera descriptors are passed on to all classes that require it. To detect objects in images, we first load the model using the Tflite.loadModel method available in the tflite package. We then retrieve images from the phone gallery or take them from the app camera. On loading the image, we feed it into the model, which returns the detected class, confidence, and points of interest that will help us draw bounding boxes around the objects (in our case trophozoites and WBCs). The basic idea of drawing bounding boxes around the detected object is to use the model's POIs (Point Of Interests) and the image dimensions (see APK (Android Package Kit) at (https://drive.google.com/file/d/1GmWMyQ_IznxCZQbmW7g4QUa2OpuhLl4/view?usp=sharing)). All mobile smartphone model deployment experiments were performed on a Samsung j6 Android smartphone camera operating at a resolution of 3264×2448 pixels (see inference in Figure 9).

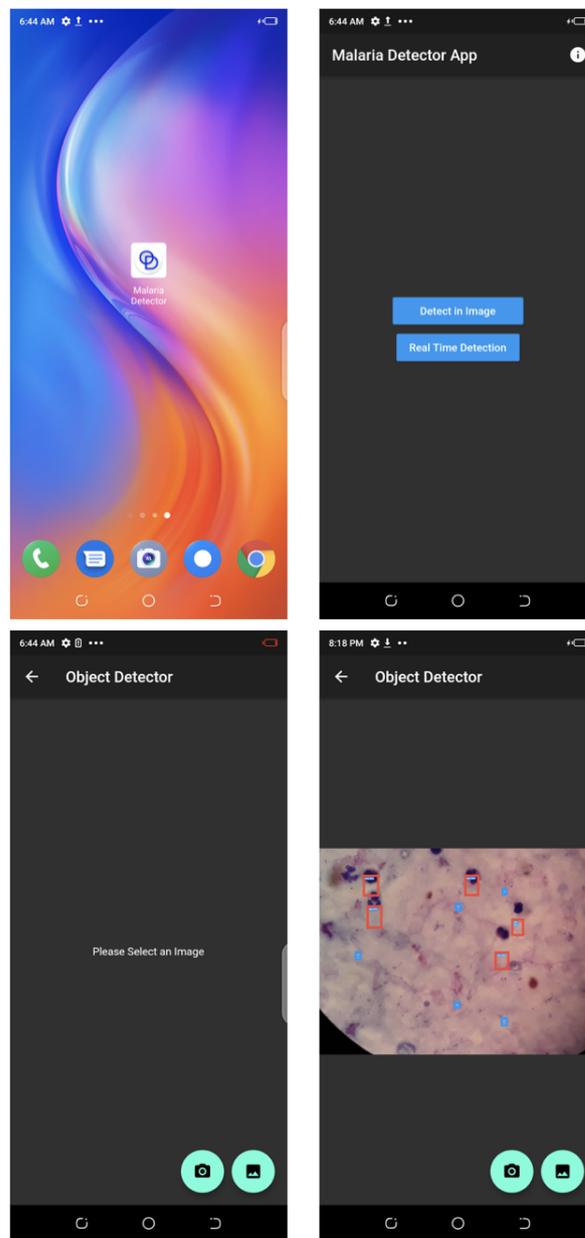


Figure 9. Malaria detection application on Android smartphone.

5. Conclusions and Future Work

In this study, we have provided an end-to-end machine learning approach for localization of trophozoites and WBC in thick blood smears for the determination of malaria parasite density and integrate the model into a mobile smartphone detection App. We present a low-cost and reliable mobile phone malaria app diagnostic solution that is relevant to highly endemic low resourced settings. We leverage transfer learning and propose a Faster R-CNN ResNet 101 model and SSD MobileNet V2 as the on-device inference approach. All models were pre-trained on COCO dataset for the task of object detection. We show that our proposed models can accomplish parasite and WBC localization and count effectively in a fast, accurate and consistent way for malaria parasitemia determination.

Furthermore, we evaluated and compared our proposed model based on a multi-class task for detection of both malaria trophozoites and WBCs with performance from single-class tasks for detection of trophozoites (only) and WBCs (only) using the same model. The mAP performance of a multi-class task (0.6609) is better than that of a single class of trophozoite detection (0.5506) and thus a higher precision and recall but lower than that of a single-class of WBCs (0.892). It is expected that models perform better with bigger images, like WBCs, compared to smaller objects (trophozoites only). In fact, SSD MobileNet performed better than Faster R-CNN on bigger objects (WBCs). The better performance of the multi-class task is attributed to having bigger objects (WBCs) complementing small objects (trophozoites). Generally, the Faster R-CNN model for a multi-class task achieved a mAP value of 0.6609 (@0.5IOU) with F1 score of (parasites: 0.7897 and WBC: 0.8426) providing better model performance in comparison to SSD MobileNet model with a mAP value of 0.6292 (@0.5IOU) and (parasites: 0.6038 WBC: 0.7528) as expected [32,41]. SSD provided capacity for mobile smartphone deployment. Conclusively, therefore, different models can be chosen according to the actual needs with respect to accuracy versus low-core device model deployment.

In future work, we intend to work towards reducing false positives and false negatives generated by the model. This could be through acquiring a bigger dataset with quality data annotation. In its current form, the mobile smartphone app still outputs false alarms. Model improvement is still needed in respect to reduction of false alarms and accuracy improvements.

Author Contributions: R.N. and E.M. designed the study; R.N. implemented the system; A.Z. and E.M. participated in the study design and supervision; R.N. wrote the manuscript; all authors analyzed the results and helped in proofreading the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: The first author of this study was funded in part by the Swedish International Development Cooperation Agency (SIDA) and Makerere University under SIDA Contribution No.: 51180060. The grant is part of the European and Developing Countries Clinical Trials Partnership (EDCTP2) programme supported by the European Union.

Acknowledgments: The authors thank Ministry of Health, Uganda for providing authorization to use the data. They also acknowledge the laboratory Technicians in Mulago Referral Hospital, Uganda especially Alfred Andama and Vincent Wadda for granting us medical and technical support in data acquisition, annotation and interpretation of results.

Conflicts of Interest: The authors declare that they have no competing interests.

References

1. WHO. *World Malaria Report*; Technical Report; World Health Organisation: Geneva, Switzerland, 2018.
2. Omeara, W.P.; McKenzie, F.E.; Magill, A.; Forney, J.; Permpanich, B.; Lucas, C.; Gasser, R.; Wongsrichanalai, C. Sources of variability in determining malaria parasite density by microscopy. *Am. J. Trop. Med. Hyg.* **2005**, *73*, 593–598. [[CrossRef](#)]
3. WHO. *Guidelines for Treatment of Malaria*, 3rd ed.; World Health Organisation: Geneva, Switzerland, 2015.
4. Kilian, A.; Metzger, W.; Mutschelknauss, E.; Kabagambe, G.; Langi, P.; Korte, R.; von Sonnenburg, F. Reliability of malaria microscopy in epidemiological studies: results of quality control. *Trop. Med. Int. Health* **2000**, *5*, 3–8. [[CrossRef](#)] [[PubMed](#)]

5. Yang, F.; Poostchi, M.; Yu, H.; Zhou, Z.; Silamut, K.; Yu, J.; Maude, R.J.; Jaeger, S.; Antani, S. Deep Learning for Smartphone-Based Malaria Parasite Detection in Thick Blood Smears. *IEEE J. Biomed. Health Inform.* **2020**, *24*, 1427–1438. [[CrossRef](#)] [[PubMed](#)]
6. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016.
7. Kelly, C.; Karthikesalingam, A.; Suleyman, M.; Corrado, G.; King, D. Key challenges for delivering clinical impact with artificial intelligence. *BMC Med.* **2019**, *17*, 195. [[CrossRef](#)]
8. Lu, Y. *Transfer Learning for Image Classification*; NNT: LYSEC045; Niversité de Lyon: Lyon, France, 2017.
9. Razavian, A.; Azizpour, H.; Sullivan, J.; Carlsson, S. CNN features off-the-shelf: An astounding baseline for recognition. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops, Columbus, OH, USA, 24–27 June 2014; pp. 806–813.
10. Cheplygina, V.; de Bruijne, M.; Pluim, J. Not-so-supervised: A survey of semi-supervised, multi-instance, and transfer learning in medical image analysis. *arXiv* **2018**, arXiv:1804.06353.
11. Lin, T.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; pp. 740–755.
12. Sio, S.; Sun, W.; Kumar, S.; Bin, W.; Tan, S.; Ong, S.; Kikuchi, H.; Oshima, Y.; Tan, K. MalariaCount: An image analysis-based program for the accurate determination of parasitemia. *J. Microbiol. Methods* **2007**, *68*, 11–18. [[CrossRef](#)]
13. Savkare, S.; Narote, S. Automatic Detection of Malaria Parasites for Estimating. *Int. J. Comput. Sci. Secur.* **2011**, *5*, 310.
14. Poostchi, M.; Ersoy, I.; McMenamin, K.; Gordon, E.; Palaniappan, N.; Pierce, S.; Maude, R.; Bansal, A.; Srinivasan, P.; Miller, L.; et al. Malaria parasite detection and cell counting for human and mouse using thin blood smear microscopy. *J. Med. Imaging* **2018**, *5*, 044506. [[CrossRef](#)]
15. Rosado, L.; Correia da Costa, L.; Elias, J.; Cardoso, D. A Review of Automatic Malaria Parasites Detection and Segmentation in Microscopic Images. *Anti Infect. Agents* **2016**, *14*, 11–22. [[CrossRef](#)]
16. WHO. Informal Consultation on Quality Control of Malaria Microscopy. 2006. Available online: https://apps.who.int/iris/bitstream/handle/10665/70075/WHO_HTM_MAL_2006.1117_eng.pdf?sequence=1 (accessed on 23 November 2020).
17. Bejon, P.; Andrews, L.; Hunt-Cooke, A.; Sanderson, F.; Gilbert, S.; Hill, A. Thick blood film examination for Plasmodium falciparum malaria has reduced sensitivity and underestimates parasite density. *Malar. J.* **2006**, *5*. [[CrossRef](#)]
18. Freen, J. Reliable enumeration of malaria parasites in thick blood films using digital image analysis. *Malar. J.* **2009**, *8*, 218. [[CrossRef](#)]
19. Bengio, Y.; Vincent, P.; Courville, A. Representation learning: A review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 1798–1828. [[CrossRef](#)]
20. Litjens, G.; Thijs, K.; Babak, E.B.; Arnaud, A.A.S.; Francesco, C.; Mohsen, G.; Jeroen, A.L.; Bram, V.G.; Clara, I.S. A survey on deep learning in medical image analysis. *Med. Image Anal.* **2017**, *42*, 60–88. [[CrossRef](#)] [[PubMed](#)]
21. Najafabadi, M.; Villanustre, F.; Khoshgoftaar, T.; Seliya, N.; Wald, R.; Muharemagic, E. Deep learning applications and challenges in big data analytics. *J. Big Data* **2015**, *2*. [[CrossRef](#)]
22. Nakasi, R.; Mwebaze, E.; Zawedde, A.; Tusubira, F.; Akera, B.; Maiga, G. A new approach for microscopic diagnosis of malaria parasites in thick blood smears using pre-trained deep learning models. *SN Appl. Sci.* **2020**, *2*, 1255. [[CrossRef](#)]
23. Quinn, J.; Nakasi, R.; Mugagga, P.; Byanyima, P.; Lubega, W.; Andama, A. Deep Convolutional Neural Networks for Microscopy-Based Point of Care Diagnosis. In Proceedings of the Machine Learning for Healthcare Conference, Los Angeles, CA, USA, 19–20 August 2016; Volume 50.
24. Tzutalin, D. LabelImg. 2015. Available online: <https://github.com/tzutalin/labelImg> (accessed on 6 May 2020).
25. Everingham, M.; Eslami, S.M.A.; Gool, L.V.; Williams, C.K.I.; Winn, J.; Zisserman, A. The pascal visual object classes challenge: A retrospective. *Int. J. Comput. Vis.* **2015**, *111*, 98–136. [[CrossRef](#)]
26. (TensorFlow): Large-Scale Machine Learning on Heterogeneous Systems. Available online: <https://arxiv.org/abs/1603.04467> (accessed on 14 April 2020).
27. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *39*, 1137–1149. [[CrossRef](#)] [[PubMed](#)]
28. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.; Berg, A. SSD: Single Shot MultiBox Detector. In Proceedings of the European Conference on Computer Vision—ECCV, Amsterdam, The Netherlands, 8–16 October 2016; pp. 21–37.
29. Sa, B.; Owens, W.; Wiegand, R.; Studin, M.; Capoferri, D.; Barooha, K.; Greaux, A.; Rattray, R.; Hutton, A.; Cintineo, J. Intervertebral disc detection in x-ray images using faster r-cnn. In Proceedings of the 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Seogwipo, Korea, 11–15 July 2017; pp. 564–567.
30. Sun, X.; Wu, P.; Hoi, S. Face detection using deep learning: An improved faster rcnn approach. *Neurocomputing* **2018**, *299*, 42–50. [[CrossRef](#)]
31. Yang, L.; Sang, N.; Gao, C. Vehicle parts detection based on faster-rcnn with location constraints of vehicle parts feature point. In *Pattern Recognition and Computer Vision*; International Society for Optics and Photonics: Bellingham, WA, USA, 2017; Volume 10609, p. 106091.
32. Nguyen, S.; Do, T.; Ngo, T.; Le, D.D. An Evaluation of Deep Learning Methods for Small Object Detection. *J. Electr. Comput. Eng.* **2020**, *2020*, 3189691. [[CrossRef](#)]
33. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27 June–30 June 2016; pp. 770–778.

34. Howard, A.G. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv* **2018**, arXiv:1704.04861.
35. Huang, J.; Rathod, V.; Sun, C.; Zhu, M.; Korattikara, A.; Fathi, A.; Fischer, I.; Wojna, Z.; Song, Y.; Guadarrama, S.; et al. Speed/accuracy trade-offs for modern convolutional object detectors. In Proceedings of the CVPR, Honolulu, HI, USA, 21–26 July 2017.
36. Shin, H.; Roth, H.; Gao, M.; Lu, L.; Xu, Z.; Nogues, I.; Yao, J.; Mollura, D.; Summers, R.M. Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning. *IEEE Trans. Med. Imaging* **2016**, *35*, 1285–1298. [[CrossRef](#)] [[PubMed](#)]
37. Dogo, E.; Afolabi, O.; Twala, B.; Nwulu, N. A Comparative Analysis of Gradient Descent-Based Optimization Algorithms on Convolutional Neural Networks. In Proceedings of the 2018 International Conference on Computational Techniques, Electronics and Mechanical Systems (CTEMS), Belgaum, India, 21–22 December 2018. [[CrossRef](#)]
38. Yadav, N.; Binay, U. Comparative Study of Object Detection Algorithms. *Int. Res. J. Eng. Technol.* **2017**, *4*, 586–591.
39. Cartucho, J. mAP. 2019. Available online: <https://github.com/Cartucho/mAP> (accessed on 7 July 2020).
40. P.N. of Excellence. Pascal VOC. 2019. Available online: <http://host.robots.ox.ac.uk/pascal/VOC/> (accessed on 11 May 2020).
41. Sanchez, S.A.; Romero, H.J.; Morales, A.D. A review: Comparison of performance metrics of pre-trained models for object detection using the TensorFlow framework. In *IOP Conference Series: Materials Science and Engineering*; IOP Publishing: Bristol, UK, 2020; Volume 844, p. 012024. [[CrossRef](#)]
42. Soviany, P.; Ionescu, R.T. Optimizing the Trade-off between Single-Stage and Two-Stage Deep Object Detectors using Image Difficulty Prediction. *arXiv* **2018**, arXiv:1803.08707v3.
43. WHO. Malaria Parasite Counting. 2009. Available online: <https://apps.who.int/iris/bitstream/handle/10665/274382/MM-SOP-09-eng.pdf?sequence=14&isAllowed=y> (accessed on 8 May 2020).
44. Kloub, A. Methods of Estimating and Counting Malaria Parasites Density. *EC Microbiol.* **2019**, *15*, 800–803.
45. Garcia, L. Parasitemia Determined from Conventional Light Microscopy: Clinical Correlation and Interpretation. 2015. Available online: http://www.med-chem.com/pages/lab_procedures/pdf/determination_of_parasitemia.pdf (accessed on 17 August 2020).
46. Farhoodfar, A. Machine Learning for Mobile Developers: Tensorflow Lite Framework. 2019. Available online: https://www.researchgate.net/publication/333659766_Machine_Learning_for_Mobile_Developers_Tensorflow_Lite_Framework (accessed on 5 January 2021).