IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 12 August 2022, accepted 14 September 2022, date of publication 21 September 2022,
date of current version 30 September 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3208270

APPLIED RESEARCH

Deep Learning for Real-Time Malaria Parasite Detection and Counting Using YOLO-mp

ANAND KOIRALA[1], MEENA JHA[2], SRINIVAS BODAPATI[3], ANIMESH MISHRA[4], (Member, IEEE), GIRIJA CHETTY[5], (Senior Member, IEEE), PRAVEEN KISHORE SAHU[6], SANJIB MOHANTY[6], TIMIR KANTA PADHAN[6], JYOTI MATTOO[7], AND AJAT HUKKOO[3]

[1]Centre for Intelligent Systems, School of Health, Medical and Applied Sciences, Central Queensland University, Rockhampton, QLD 4701, Australia
[2]Centre for Intelligent Systems, School of Engineering and Technology, Central Queensland University, Sydney, NSW 2000, Australia
[3]Intel Corporation, Santa Clara, CA 95052, USA
[4]NVIDIA Corporation, Santa Clara, CA 95051, USA
[5]Faculty of Science and Technology, University of Canberra, Canberra, ACT 2617, Australia
[6]Community Welfare Society Hospital, Rourkela, Odisha 769042, India
[7]Intel Technology India Pvt. Ltd., Bengaluru, Karnataka 560103, India

Corresponding author: Meena Jha (m.jha@cqu.edu.au)

This work was supported in part by the Australia India Council Grant through Australian Government Department of Foreign Affairs and Trade under Grant AIC-116-2021, and in part by the Center for Intelligent Systems, Central Queensland University, Australia.

ABSTRACT Malaria in the rural and remote regions of tropical countries remain a major public health challenge. Early diagnosis and prompt effective treatment are the basis for the management of malaria and for reducing malaria mortality and morbidity worldwide and the key to malaria elimination. While Rapid Diagnostic Test (RDT) remains the current mainstay testing malaria infections, it is usually used in conjunction with clinical findings and lab tests of blood films through Microscopy- the gold standard of malaria diagnosis. Recent reports suggest that the accuracy of RDTs could be compromised due to parasite antigen gene deletion(s), and the lack of expertise and high turnover time makes microscopy impractical to be used in rural and remote areas which impede the diagnosis and treatment of the disease. Delay in receiving treatment for uncomplicated malaria is reported to increase the risk of developing severe malaria and mortality. Thus, the need to develop advanced, faster, and smarter tools for malaria diagnosis is paramount, specially to reinforce the gold standard method, i.e., malaria microscopy which is a full-proof tool given the limitations be addressed. Deep learning-based methods have proven to provide human expert level performance on object detection/classification on image data. Such methods can be utilized for automation of repetitive task in assessing large number of microscope images of blood samples. In this paper, we propose a novel approach to improve the performance of deep learning models through consistent labelling of ground truth bounding box for the task of pathogen detection on microscope images of thick blood smears. Recommendations are made on the reliability and repeatability testing of the trained models. A custom deep learning architecture (YOLO-mp) is developed based on the design criteria of optimizing accuracy and speed of detection with minimal resources. The custom three-layered YOLO-mp-3l and four-layered YOLO-mp-4l models achieved the best mAP scores of 93.99 (@IoU=0.5) and 94.07 (@IoU=0.5), respectively outperforming standard YOLOv4 (mAP 92.56 @IoU=0.5) for detection of malaria pathogen on a public dataset of thick blood smear microscope images captured using phone camera. YOLO-mp-3l (BFLOPs = 21.8, model size = 24.5Mb) and YOLO-mp-4l (BFLOPs=24.477, model size = 25.4Mb) outperformed standard YOLOv4 (BFLOPs=127.232, model size = 244Mb) in terms of computation and memory requirements proving them suitable to run on low resource devices.

INDEX TERMS Custom YOLO, deep learning, medical imaging, microscope images, object detection, thick blood smear images.

I. INTRODUCTION

Malaria disease is caused by plasmodium parasite species. There are different species of human malaria plasmodium such as *P. falciparum, P. vivax, P. ovale, P. malariae and*

The associate editor coordinating the review of this manuscript and approving it for publication was Gina Tourassi.

P. knowlesi. *P. falciparum* is the most virulent species responsible for the majority of the severe malaria complications and death [1]. While microscopic examination of the thick and thin blood smears taken from suspected malaria patients under a microscope is considered as the gold standard test to identify these malaria parasites [2], [3], [4]; the rapid diagnostic tests (RDT), is the mainstay for diagnosis of malaria across all healthcare sectors and in field settings. Nonetheless, the accuracy of these antigen-based diagnoses by RDT could become compromised due to the emergence of parasite antigen gene deletion(s) as per recent global literature [5], [6]. On the other hand, the lack of highly skilled expertise in microscopy and prolonged turnover time of reading the blood smears for an accurate malaria parasite detection, deems it most challenging to be used in rural and remote areas which impede the diagnosis and treatment of malaria at large.

Thick blood smears contain many blood cells on multiple layers and therefore usually contain high parasitemia which is suitable and sensitive for diagnosis or detection of malaria infection. Thin blood smears contain fewer blood cells in a single layer and allow a clear view for identification and classification of malaria parasite species which is necessary for providing correct treatment [4]. A repetitive test is required several times a day for several days to assess the change in the parasitemia levels throughout the treatment [4]. However, microscopic examination is laborious, subjective, time-consuming and requires expert microscopists [2], [3], [7]. Malaria is more prevalent in developing countries and there is a shortage of expert microscopists [2] and even if they are available the results are subject to the expert's judgment [7].

Quinn *et al.* [2] have reported increasing interest in using computer vision to automate the process of malaria detection in microscope images to compensate for the shortage of expert microscopists. Alternatively, decision support tools based on computer vision can also speed up the diagnosis by speeding up the pathogen detection task while allowing the experts to make final judgements. In object classification tasks, deep learning has previously been known to surpass human-level performance [8].

Deep learning is an emerging field under machine learning and have an ability to automatically learn important features from large amounts of data to produce accurate results. Deep learning builds on neural network which simulates the human brain and works on analysis and learning via input perception data into a mechanism of deep neural networks. However, training a supervised model requires ground-truth labels with the need for identifying target objects (e.g., parasites) in images by an expert. Large number of overlapping cells with different orientations, and unclear morphological features such as shape, color, and size of blood cells in thick blood smear microscope images make it difficult to identify malaria parasites from artifacts compared to thin blood smears. These challenges bring difficulties in precisely drawing annotations for rectangular bounding boxes or polygons around the perimeter of the target parasites on microscope images for the purpose of training object detection or segmentation models. Object detection methods rely on both localization (also called the position of objects in images), and classification (also called as the category of the detected object). Therefore, the performance of such methods is sensitive to the consistency of labelling or drawing annotations and the size of target objects in the images. Moreover, the ambiguity in ground-truth labelling can result in missing object labels in the training datasets that can adversely affect the performance of supervised object detection models [9].

The number of parasites on an image indicates the severity of infection in a blood sample. With deep learning-based object detection methods, not only the objects can be localized in images but also their number can be counted. Parasites and their morphologies are clearly visible in thin blood smears but, contain a smaller number of parasitemia compared to thick smears. However, a large number of parasites and their morphologies can be seen in thick blood smear. Therefore, examining a thick blood smear is recommended over a thin blood smear [10] as thick smears allow more efficient detection of parasites with increased sensitivity.

In general, there are two deep learning detection frameworks in the field of object detection, one is one-stage object detection, and the other is two-stage object detection. Faster Regional Convolutional Network (Faster R-CNN) [11] is one of the widely used two-stage object detection frameworks based on deep learning CNN. You Only Look Once (YOLO) [12] and Single Shot Detector (SSD) [13] are among the few most popular single-stage object detection frameworks. Single-stage object detectors are very fast compared to two-stage detectors and find their use in real-time applications.

YOLO has officially evolved from version 1 to version 4 [12], [14], [15], [16] and recently version 7 [17] with substantial changes making it better and one of the state-of-the-art algorithms in object detection. The performance of such object detection frameworks is benchmarked against large image datasets of day-to-day objects covering a large portion of objects in images, such as Visual Object Challenge (PASCAL VOC) [18] and Common Objects in Context (COCO) [19] containing 20 and 80 object categories, respectively. The standard CNN architectures and models designed and trained for such datasets may require modifications and tuning to work on the datasets from other domains e.g., medical x-ray images. Therefore, depending on the application type there is a need for modification in the classification part of CNN architecture or the detection part of the pipeline.

In this paper, a method to improve the performance of deep learning models through consistent labelling of the ground truth bounding boxes is proposed for the task of malaria parasite detection on microscope images of thick blood smears. Recommendations are made on the reliability and repeatability testing of the trained models. A custom deep learning architecture (YOLO-mp) was developed on the design criteria of accuracy and speed for automation of current application in a low resource setting to be used predominantly in rural areas.

A. PUBLISHED DATASETS USED IN THIS STUDY

We have used two published open access malaria datasets from the Makerere AI Lab, Makerere University, Uganda (http://air.ug) and named as Dataset A, and Dataset B for our study.

1) DATASET A [2]

Dataset A is a ''plasmodium-images.zip'' dataset and contains 2703 color images, taken from 133 thick blood smears treated with field stain, and all the images have a resolution of 1024 × 768 pixels (http://air.ug/datasets/; accessed on 20/06/2022). Each image has an accompanying annotation file containing the coordinates of bounding boxes around any visible plasmodium. Images were captured using a Motic MC1000 camera mounted on a Brunel SP150 microscope at 1000× magnification.

2) DATASET B [10]

Dataset B is a ''plasmodium-phonecamera.zip'' dataset and the images were collected using a smartphone camera attached to a microscope's eyepiece at x1000 magnification. Dataset B contains 1182 color images of thick blood smears treated with field stain, and all the images have a resolution of 750 × 750 pixels. It contains 948 malaria-infected images with 7628 *P. falciparum* parasites and 234 normal (negative) images with artifacts due to impurities (http://air.ug/datasets/; accessed on 20/06/2022). Each image has an accompanying annotation file containing the coordinates of bounding boxes around any visible *P. falciparum* parasite.

II. RELATED WORKS IN THIS FIELD

Depending on the applications sometimes there arises a need to run an object detection model in a real-time setting on a low-resource hardware device. In such low-resource settings, the model size and detection speed are also as important as the model's accuracy. Quinn *et al.* [10] have reported a significant improvement in the model performance with the use of deep learning CNN for malaria pathogen detection on microscope images. A sliding window (SW) approach was used to pull out the overlapping patches from the original image and classify each patch using methods such as the Extremely Randomized Trees (ERT) classifier [2] or using CNN [10]. However, both, [2] and [10], reported their model performance on the classification accuracy of patches and not for object detections on the full image.

Chibuta and Acar [3], experimented with YOLOv3 [15] for malaria pathogen detection on both datasets from studies conducted by Quinn *et al.* [2] and [10]. Chibuta and Acar [3] re-implemented the Quinn *et al.* [10] (SW +CNN) for evaluating detection performance and achieved very poor Mean Average Precision (mAP) 0.515 on Dataset A and 0.685 on Dataset B. the modified YOLOv3 achieved the best mAPs 0.887 and 0.902 for Dataset A and Dataset B, respectively, and a detection speed of about 0.42 seconds per image (800 x 800 pixels) using a CPU computer [3].

Abdurahman *et al.* [20] reported a very good detection accuracy from their modified YOLOv4 model (mAP=96.32 @ inference IoU 0.3, mAP=89.73 @ inference IoU 0.5). Results from study [20] showed that all YOLO models outperformed Faster RCNN [11] (two-stage detector) (mAP = 71.0 @ inference IoU 0.3) and SSD [13] (one-stage detector) (mAP=71.4 @ inference IoU 0.3). The authors also argued that the modifications by extending feature scales and introducing an additional detection layer to the standard YOLOv3 and YOLOv4 models have improved the capability to detect small objects on images.

III. RESEARCH GAP

Chibuta and Acar [3] modified YOLOv3 to be very small and tiny for increased detection speed but with decreased performance for the detection of malaria on microscope images. Abdurahman *et al.* [20] modified YOLOv4 to achieve higher performance for the detection of malaria parasites but with added detection layers making the models very complex and computationally expensive. Since Abdurahman *et al.* [20] reported YOLO models outperformed Faster R-CNN and SSD for detecting malaria parasites on thick blood smear microscope images, we used a similar line of investigation and conducted experiments based on the YOLO object detection framework. Bochkovskiy *et al.* [16] have also reported that YOLOv4 has a promise for both speed and accuracy. Our previous study [21] in which standard YOLO architecture was re-designed for fruit detection tasks to run on lower memory and higher speed through reduced computation without compromising the detection accuracy, established a starting point for pathogen detection with low hardware resources. Therefore, in this study, we aimed to experiment with YOLOv4 models with the following objectives.

- Diagnosis of malaria under microscopy through visual inspection of blood film is considered a gold standard but it is laborious, time-consuming, and requires an expert microscopist. Therefore, a model for automated detection of malaria pathogens in microscopy images is desirable.
- Malaria is mostly prevalent in less developing countries which face poor health facilities and a shortage of expert microscopists. When there are large number of patients it is difficult for the limited number of experts to do timely diagnoses. Therefore, it is desirable that the trained model be able to run on real-time in low-resource setting devices.

IV. MATERIALS AND METHODS

The research study by Chibuta and Acar [3] has established a baseline for what we could expect from the trained models to benchmark against human performance for pathogen detection on Dataset A and Dataset B images. Dataset B is chosen in our study for training and validation of object detection models because of two reasons, firstly, this dataset is produced under a low-cost setting by microscope image captured using a general phone camera, and secondly, Field stain is used for quick smear preparation. Both are suitable for developing practical applications targeting malaria-endemic regions of the globe.

Moreover, Dataset B contains images of *P. falciparum* which is the deadliest and most prevalent species of malaria parasite in endemic regions such as Africa. All images and XML annotations from Dataset A and Dataset B were uploaded to the Roboflow website (https://roboflow.com/; accessed on 20/06/2022) and the annotations were exported in darknet format for YOLO model training, validation, and testing. All models were trained using the official darknet framework for YOLOv4 (https://github.com/AlexeyAB/darknet; accessed on 20/06/2022).

Transfer learning is commonly used in deep learning to initialize a model with weights from pre-trained models usually trained on large datasets such as PASCAL VOC [18], COCO [19], and ImageNet [22]. Such a strategy allows to train large models on relatively small datasets of similar applications through re-using previously learned weights from a larger dataset. Transfer learning was used for all YOLOv4 standard models used in this study through initialization with weight files from COCO pre-trained models (https://github.com/AlexeyAB/darknet/wiki/YOLOv4-model-zoo; accessed on 20/06/2022).

A. TRAIN-VALID SET

Dataset B was randomly split into a train-valid set constituting 90% of images in the training set and 10% of images in the validation set. The validation dataset was not used to control the training behavior in our current study and was solely used to observe the model's performance. In this context, more data would be available for training. Alternatively, a k-fold, where k=10, cross-validation is performed to test the repeatability of trained models on Dataset B.

B. TEST SET

Ideally, a test set should be an independent set that is other than the split of the current dataset and can be used for testing model robustness for accuracy and generalizability. Dataset A is used as a test set in our current study to assess the robustness of models trained on Dataset B images.

C. MODEL CONFIGURATION

Default training parameters in the YOLO configuration file were updated with new parameters. The new parameters used are learning rate = 0.001, momentum = 0.9, max batches = 4000, steps = 3200, 3600 as specified by Abdurahman *et al.* [20].

In the YOLO configuration file, parameters 'max_batches' specify the total number of iterations while the 'steps' learning policy updates the starting learning rate (0.001) at specified iterations 3200 (i.e., 80% of total iterations) and 3600 (i.e., 90% of total iterations) with new learning rates 0.0001 and 0.00001, respectively calculated using scale values ('scales = 0.1, 0.1') during model training.

YOLO uses a set of prior/anchor boxes known as 'masks' defined in the configuration file as initial sizes of height, and width to regress the bounding box around detections. The 'calc_anchors' command from darknet was used to determine anchors for our training set.

D. MODEL EVALUATION METRICS

1) INTERSECTION OVER UNION (IOU)

IoU is a metric whose value is between 0 and 1. IoU of 0 indicates no overlap and IoU of 1 indicates complete overlap between two bounding boxes. For detection algorithms, a box will be treated as true detection for both model training and inference, if the overlap between the detected box and ground truth box is above the set IoU threshold. Using a lower IoU threshold during inference allows to increase True Positive (TP) by accepting boxes with small overlaps as true detection.

For each detection the trained model also returns a confidence score based on how accurate the prediction is. Detections can be filtered out by thresholding inference-time confidence scores. It is possible to detect more objects with high chances of False Positives (FP) when the confidence threshold is set to lower values.

2) F1 SCORE

F1 score is a harmonic mean between precision and recall therefore, F1 will be maximum when both precision and recall are maximum. Depending on the application we can trade-off precision and recall of a trained model for detection task by adjusting IoU and confidence threshold values which will affect the F1 score.

3) MEAN AVERAGE PRECISION

Mean average precision mAP quantifies the performance of the model by summarizing the precision-recall curve. mAP is affected by changes in IoU threshold values because an IoU determines whether detection is to be considered true or false. For, a fixed IoU threshold, the change in confidence threshold value will not affect mAP but F1 score. Therefore, in this study we will report only the mAP values for performance evaluation of trained models.

V. BASELINE YOLOv4 ARCHITECTURES

A. YOLOv4 ARCHITECTURE

- YOLOv4 backbone:
  - YOLOv4 uses CSPDarknet53 backbone as feature extractor which has 53 convolutional layers arranged as dense blocks with Cross-Stage-Partial-connections (CSP) [23]. A better result was obtained with CSPDarknet53 and "Mish" [24] activation function. Misra [24] in his study claimed that "Mish" outperformed many other activation functions in various datasets.
- YOLOv4 neck:
  - YOLOv4 also used Spatial Pyramid Pooling (SPP) [25] block before the first YOLO detection head. In YOLOv4's SPP block the convolutional Kernels of different sizes (1×1, 5×5, 9×9, 13×13) are slid

on a feature map with maximum pooling operation and the features finally concatenated together to get output of same spatial size. It is multi-scale max-pooling.

- Full-YOLOv4 finally implements Path aggregation Net (PAN) [26] as the neck after the backbone and just before the YOLO detection head. Unlike PAN [26], which adds neighbor layers together, YOLOv4 concatenates feature maps together in its PAN implementation.

- YOLOv4 head:
  - For network input resolution of 608 × 608 pixels YOLOv4 uses three detection heads on feature maps of 19 × 19, 38 × 38 and 76 × 76 pixels.

B. YOLOv4-TINY ARCHITECTURE

YOLOv4-tiny uses the tiny version of CSPDarknet53 feature extractor as backbone. There are only three CSP Nets in CSPDarknet53-tiny with "leaky" activation functions. Unlike full-YOLOv4 as shown in Figure 1, SPPNet and PANet are not part of the YOLOv4-tiny architecture as shown in Figure 2. This tiny architecture design reduces the computational cost of YOLOv4 which makes YOLOv4-tiny the best model in terms of detection speed. The "yolov4-tiny.cfg" and "yolov4-tiny-3l.cfg" come with two and three detection heads, respectively.

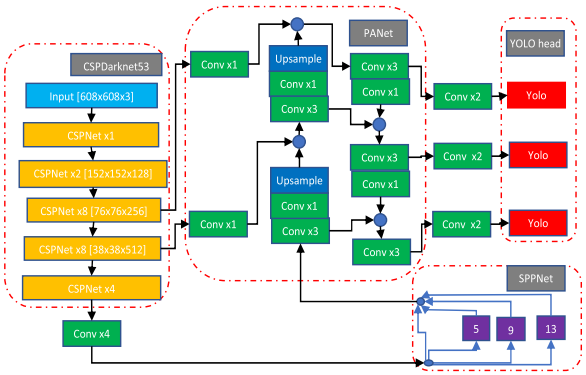

FIGURE 1. Architecture of original YOLOv4.

VI. STUDY CONDUCTED USING BASELINES

A. PRELIMINARY STUDY

Full version of standard YOLOv4 model was used for baseline study.

1) FULL-YOLOv4 MODEL

YOLOv4 (yolov4-custom.cfg) was trained on Dataset B with all parameters specified in section 'Model configuration'. Individual model was trained with varying train IoU thresholds of 0.3 to 0.6 in steps of 0.1, and detection mAP obtained for varying inference IoU thresholds for 0.3 and 0.5, as shown in Table 1.

2) FULL-YOLOv4 PERFORMANCE ANALYSIS ON DATASET B

The best mAP and average IoU was obtained for train and inference IoU of 0.5 and 0.3, respectively as shown in Table 1.

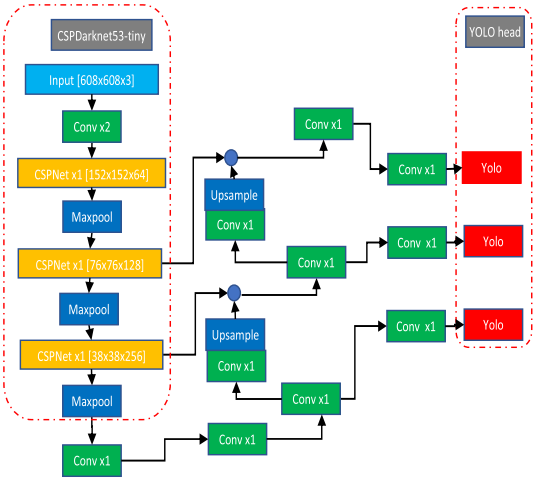

FIGURE 2. Architecture of original YOLOv4-tiny-3l.

TABLE 1. Full-YOLOv4 performance on validation set for inference IoU 0.5 and IoU 0.3 for models trained using different train IoU (0.3 to 0.6). Avg. IoU is the average overlap between predicted and ground truth bounding box for validation set images.

Train IoU	mAP@0.5	Avg. IoU@0.5	mAP@0.3	Avg. IoU@0.3
0.3	79.57	43.44%	85.97	44.85%
0.4	82.64	47.60%	87.97	48.77%
0.5	**84.72**	**47.67%**	**89.60**	**48.90%**
0.6	82.00	43.35%	87.77	44.75%

Increased IoU threshold values during model training will allow only the objects detected with greater overlaps as true positives therefore the model will detect less false positives. However, if an object was truly classified but failed to meet the IoU threshold then the object will not be considered as detection by YOLO method.

The significant differences (∼5% difference) between columns 2 and 4 of Table 1, for different training IoU thresholds (column1), indicates that the trained model was unable to fit bounding box properly on the detected objects. This is supported by the fact that the average IoU of the models is very low and less than 50% (<50%). Lower average overlap between detected and ground truth boxes due to model's localization error can come from improper ground truthing.

B. PATHOGEN IDENTIFICATION CHALLENGES ON IMAGES

Object detection involves both localization and classification of the target object in images. The goal of the model training is to minimize training loss which is a weighted combination of the localization and the classification error/loss. The measure of overlap between ground truth and predicted bounding box in images determines if a prediction can be considered as a detection. Similarly, the predicted class/category of the detected target is assessed against ground truth label to determine if the detection is a true positive or a false positive. Therefore, if the ground truth boxes are not tight and not

consistent around the target object it will be difficult for an object detection algorithm to train an accurate model.

In general an IoU threshold of 0.5, it is about 52.5% overlap, and is used for detecting objects on image datasets like PASCAL VOC [18] that consists of images of the general objects but this threshold can be tuned specific to the applications. However, Chibuta and Acar [3] have reported better mAP can be obtained from the trained models when using IoU threshold of 0.3 for inference on test sets of Dataset B that is consistent to the report in [20]. The authors of [3] argued that IoU of 0.3 was suitable to account for inconsistencies in placing pathogen in the center of ground truth bounding box.

C. FURTHER INVESTIGATION ON DATASET B

Chibuta and Acar [3] reported annotation error on images of current Dataset B. A closer look on the image datasets revealed that the boxes were not consistently drawn around the pathogen. There were many tiny boxes of about 1 or 2 pixels. Some boxes contain nothing, some boxes were too big around the object, and some boxes only covered part of the pathogen. This report from [3] along with our results shown in Table 1 warranted for further investigation on the current training dataset Dataset B itself.

We observed following annotation errors in Dataset B:

- 7 images contained at least one duplicate bounding box: (image names: 0044, 0208, 0239, 0578, 0967, 1083, 0545).
- 21 images contained at least one very tiny bounding box (1-2 pixels): (image names: 0051, 0081, 0089, 0091, 0389, 0410, 0547, 0567, 0682, 0762, 0797, 0907, 0951, 1019, 1029, 1113, 0433, 0479, 0707, 0940, 1024).
- 8 images contained at least one blank bounding box: (image names: 0110, 0163, 0197, 0234, 0819, 0889, 1063, 1148).

D. DATASET B-CENTERED

Ground truth annotations in Dataset B were shifted in position such that the chromatin was aligned towards the center of the bounding boxes in images as in figure 3. The translation was done without resizing. A graphical annotation tool ''labelImg'' (https://pypi.org/project/labelImg/; accessed on 20/06/2022) was used for adjusting the annotation box. This process created a new dataset, and we named it as Dataset B-centered which is used in the rest of the paper. We did not attempt to completely re-annotate the dataset but rather tried to reduce training noise to make the annotations more consistent.

Following treatment was carried out in Dataset B-centered to remove annotation errors of Dataset B.

- Duplicate boxes were removed to keep one box per object.
- Tiny boxes of 1-2 pixels were removed because they can't contribute to training as they don't contain any useful information. Most of these tiny boxes were on top of the pathogen-like objects.

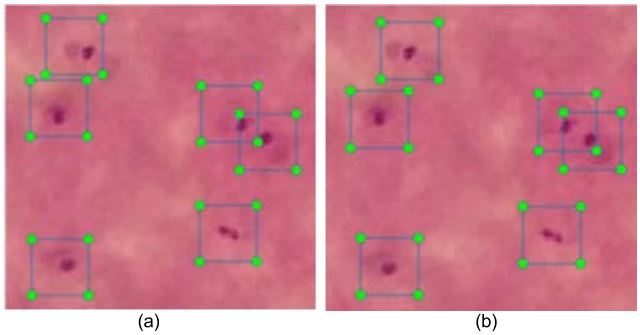

FIGURE 3. Example annotations from Dataset B (left (a)) and Dataset B-centered (right (b)).

- Some bounding boxes drawn around plain background of the image were deleted because they don't contain any useful feature information about the target pathogen class.

Chromatin is present in all pathogens irrespective of the morphologies such as shape, size, and colors. Kaewkamnerd *et al.* [27] studied the detection and classification of *p. falciparum* and *p. vivax* based on the size of chromatin on the microscope images and reported that the color and edge features of the chromatin can be easily detected on Giemsa-stained thick blood films while the cytoplasm edge can blend with the background making it difficult to determine. Therefore, in this study an attempt was made to re-position chromatin towards the center of ground truth bounding boxes.

Centering of chromatin on ground-truth bounding boxes is proposed to improve quality of data labelling and following hypotheses are established.

- Hypothesis 1 (H1): Centering chromatin on the bounding boxes could enhance the consistency of data labelling and thus improve performance of bounding box-based object detection methods through better localization and classification capabilities.
- Hypothesis 2 (H2): Through consistent labelling based on chromatin centering it is possible to craft relatively smaller models with similar or better accuracies in comparison to large and complex models while it can run in real-time under low resource settings.

E. MODEL TRAINING AND VALIDATION ON DATASET B-CENTERED

Standard full and tiny versions of YOLOv4 models were trained and assessed against custom YOLOv4 models on dataset B-centered.

All models were trained and tested on 90-10 (Train-valid) split of Dataset B-centered which is the same image list as train-valid split of Dataset B. The IoU threshold was set to 0.5 for training. Full-YOLOv4 is standard YOLOv4 model as shown in Figure 1.

1) CRAFTING CUSTOM FULL VERSION OF YOLO MODELS

The standard YOLOv4 architecture is redesigned to create custom model architectures as follows.

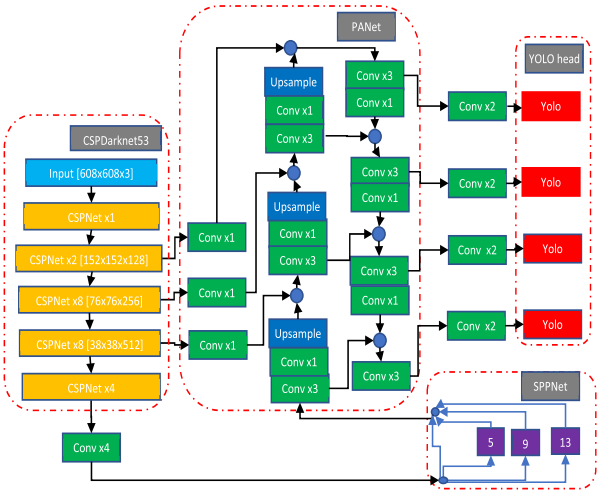

FIGURE 4. Architecture of YOLOv4-4l.

Full-YOLOv4-det104

- Input to the first detection layer (layer-54, feature map 76 × 76 pixels) of full-YOLOv4 was moved to early layer (layer-23, feature map 152 × 152 pixels i.e., 104 × 104 pixels @416 × 416 input resolution) to extract features from early stages of the CNN backbone. This new design is named Full-YOLOv4-det104.

Full-YOLOv4-4l

- One more detection layer was introduced inside PANet of full-YOLOv4 making 4 detection layers (feature maps 19 × 19, 38 × 38, 76 × 76, 152 × 152 @ input resolution 608 × 608 pixels) and used 12 cluster of anchors (3 anchors per detection layer). This new design is named Full-YOLOv4-4l as shown in Figure 4 in this study.

2) PERFORMANCE OF FULL-YOLOv4 MODELS ON DATASET B-CENTERED

Using early feature maps as input to one of the detection layers slightly improved the mAP of Full-YOLOv4 as shown in Table 2. There was no performance advantage with 4 detection layers and had an increased computational cost.

TABLE 2. Performance of full-YOLOv4 and custom full-version models trained and validated on 90-10 split Dataset B, using IoU of 0.5 for training and inference.

Model name	Transfer learning	mAP@0.5 IoU
Full-YOLOv4 (standard)	yes	90.61
Full-YOLOv4-det104	yes	**91.06**
Full-YOLOv4-4l	yes	91.00

3) CRAFTING SMALLER VERSION OF YOLO MODELS

The aim of this exercise was to re-design YOLO models to obtain smaller and faster model without compromising the accuracy that can be obtained from larger full version model. Yolov4 tiny models are the smallest and fastest model in the YOLO family of YOLOv1-v4. Therefore, yolov4-tiny-3l.cfg model configuration file from the repository was chosen as a base architecture to start crafting a custom plasmodium pathogen detection model.

The following models were trained on the 90-10 train-valid split of Dataset B-centered which has the same image list as train-valid split of Dataset B. IoU threshold for training was set to 0.5 for all models.

Yolov4-tiny-3l:

- Yolov4-tiny-3l is same as standard yolov4-tiny-3l.cfg model.

4) CRAFTING CUSTOM TINY VERSION OF YOLOv4 MODELS

Several experiments (not reported) were iteratively carried out before finally naming few model variants as follows- pertaining to the modifications that produced significant improvement in detection performance.

YOLOv4-tiny-3l-det104:

- One of the inputs to the first YOLO detection layer (@76 × 76 feature map) of yolov4-tiny-3l was moved to earlier layer (@152 × 152 feature map i.e., 104 × 104 pixels @416 × 416 input resolution) creating YOLOv4-tiny-3l-det104. This is in attempt to utilize the information about the target object learned by CNN backbone in the earlier layers which could have not propagated in the following layers- specifically for smaller objects in our study.

YOLOv4-tiny-3l-det104-SPP:

- Introduced Spatial Pyramid Pooling (SPP) network to YOLOv4-tiny-3l-det104 as in full YOLOv4 to create YOLOv4-tiny-3l-det104-SPP. This is an attempt to pool the spatial information about a target object on the features extracted by the CNN-backbone using kernels of different sizes (5 × 5, 9 × 9 and 13 × 13 pixels) as used in standard YOLOv4 full model.

YOLOv4-tiny-3l-det104-mish:

- Replaced 'leaky' activation function of YOLOv4-tiny-3l-det104 with 'mish' activation function to create YOLOv4-tiny-3l-det-104-mish. This is an attempt to switch to a newer and better activation function 'mish' which is used by standard YLOv4 full model.

YOLOv4-tiny-3l-det104-SPP-mish (YOLO-mp-3l):

- Introduction of both SPPNet and 'mish' activation function in YOLOv4-tiny-3l-det104 architecture to create YOLOv4-tiny-3l-det104-SPP-mish. This is an attempt to combine the best of both 'SPP' and 'mish' into a single model.

YOLOv4-tiny-3l-det104-SPP-mish with SPP network and 'mish' activation was the best performing model as shown in Table 3. YOLO-mp-3l also outperformed the best performing full YOLOv4 models as shown in Table 2. YOLOv4-tiny-3l-det104-SPP-mish model was chosen as the best and final tiny architecture in this study and renamed as YOLO-mp-3l as shown in Figure 5 which is short for YOLO malaria

parasite detection model having three detection layers and implemented in YOLO framework.

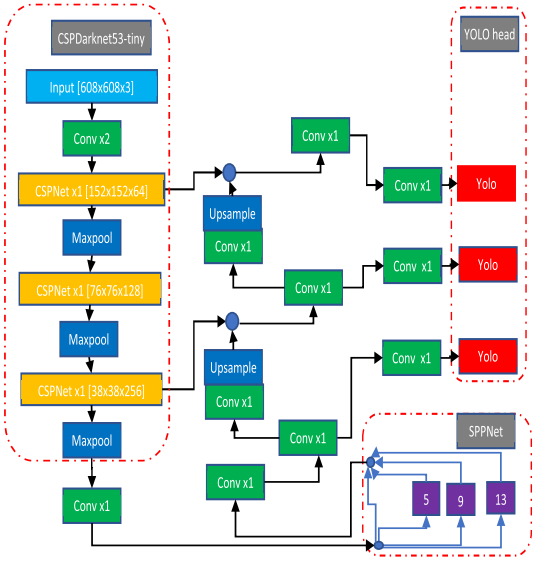

FIGURE 5. Architecture of YOLO-mp-3l.

TABLE 3. Performance of standard YOLOv4-tiny-3l and custom tiny models trained and validated on 90-10 split Dataset B, using IoU of 0.5 for training and inference.

Model name	Transfer learning	mAP@0.5 IoU
YOLOv4-tiny-3l (standard)	no	89.47
YOLOv4-tiny-3l-det104	no	90.33
YOLOv4-tiny-3l-det104-SPP	no	91.13
YOLOv4-tiny-3l-det104-SPP-mish (YOLO-mp-3l)	no	**91.56**
YOLOv4-tiny-3l-det104-mish	no	90.25

5) REPEATABILITY TEST ON DATASET B-CENTERED

K-fold validation:

To verify the consistency of model performance the Dataset B-centered consisting of 1182 images was sorted according to image names and split into 10-folds. Folds roughly 10% of dataset were created sequentially without shuffling. Fold-1 and fold-2 consisted of 119 validation images while remaining folds fold-3 to fold-10 consisted of 118 validation images each as shown in Table 4. Each fold was considered as validation set while remaining folds mixed into train set which resulted in 10 different models.

TABLE 4. Number of ground truth bounding boxes in validation set of each fold.

Fold-1	Fold-2	Fold-3	Fold-4	Fold-5	Fold-6	Fold-7	Fold-8	Fold-9	Fold-10
757	777	692	830	749	718	743	841	737	746

YOLO-mp-3l obtained best mAP of 93.44 and 93.81 on fold-5 validation set as shown in Table 5. Similarly, YOLO-mp-3l achieved mAP of 93.08% (@ 0.5 IoU) and mAP of 93.53% (@ 0.3 IoU) on fold-5 training set as shown in Table 5.

TABLE 5. Performance (mAP) of YOLO-mp-3l models trained and validated separately on set of each fold.

	Fold-1	Fold-2	Fold-3	Fold-4	Fold-5	Fold-6	Fold-7	Fold-8	Fold-9	Fold-10
@0.5 IoU	90.90	90.03	89.47	92.89	**93.44**	92.18	90.97	91.12	88.77	91.60
@0.3 IoU	91.80	90.17	89.8	93.48	**93.81**	92.33	91.25	92.27	89.33	92.25

6) HUMAN BENCHMARK ON DATASET A AND DATASET B-IMAGES

Chibuta and Acar [3] have reported mAP of 92.3% and 91.2% @ IoU 0.3 by two independent human experts on their test for Dataset A and Dataset B, respectively. In our current study, the performance average mAPs of 91.137% @ 0.5 IoU and 91.657% @0.3 IoU of YOLO-mp-3l model trained from scratch with no transfer learning and averaged across 10-fold validation set (from Table 5) is at the same level to human performance on Dataset B as reported by Chibuta and Acar [3]. Therefore, YOLO-mp-3l is all ready to be used for automation of pathogen detection on microscope images of thick blood smears.

With the success of YOLO-mp-3l architecture for pathogen detection a four-layered YOLO-mp-4l model as shown in Figure 6 was created by adding an extra detection layer on YOLO-mp-3l model for further experiments.

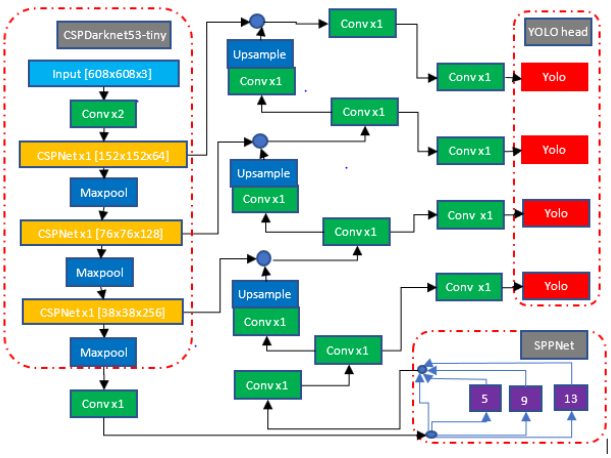

FIGURE 6. Architecture of YOLO-mp-4l.

As the best mAP was achieved on fold-5 validation set of Dataset B-centered as shown in Table 5, comparison of YOLO-mp-3l model with full-YOLOv4 model was done on same train-valid set as shown in Table 6.

For transfer learning COCO pre-trained weights file for Yolov4-tiny and Yolov4 were downloaded from model ZOO (https://github.com/AlexeyAB/darknet/wiki/YOLOv4-model-zoo; accessed on 20/06/2022) and initialized for training YOLO-mp-3l and full-YOLOv4 models, respectively.

TABLE 6. Performance of YOLO-mp-3l and full-YOLOv4 on fold-5 validation set (118 images, 749 ground truth boxes) of Dataset B-centered trained using 0.5 IoU threshold and a range of IoUs (0.3-0.6) for inference.

Model name	Transfer learning	mAP@0.3 IoU	mAP@0.4 IoU	mAP@0.5 IoU	mAP@0.6 IoU
YOLO-mp-3l	No	93.81	93.81	93.44	92.94
YOLO-mp-3l	Yes	94.20	94.20	93.99	**93.54**
YOLO-mp-4l	yes	**94.25**	**94.25**	**94.07**	93.39
Full-YOLOv4 (standard)	Yes	92.67	92.61	92.56	91.86
Full-YOLOv4-det104	Yes	92.95	92.88	92.84	92.53
Full-YOLOv4-4l	Yes	93.12	93.07	92.82	92.58

7) MODEL WITH FOUR DETECTION LAYERS

Improved performance of full-YOLOv4-det104, YOLO-mp-3l and YOLO-mp-4l models can be attributed to moving the feature input of first detection layer from deeper 52 × 52 feature map referred to 416 × 416 input resolution layer in their original configurations to early layer 104 × 104 feature map referred to 416 × 416 input resolution as shown in Table 6. Models with 4-detection layers 13 × 13, 26 × 26, 52 × 52 and 104 × 104 in reference to 416 × 416 network input resolution, however, showed insignificantly small increase in performance but additional computation cost as compared to their 3-detection layered counterparts as shown in Table 6.

Table 7 shows the performance comparison of custom 3-layered and 4-layered models against standard full-YOLOv4 models trained and validated on 90-10 split Dataset B Centered using IoU of 0.5 for training and a range of IoU 0.3-0.6 for inference.

8) MODEL ROBUSTNESS TESTING

Use Test against other public datasets:

Trained model obtained from our current study on Dataset B-centered 90−10 split and fold-5 was used to do inference on Dataset A images with 2704 microscope images and 49900 ground truth boxes.

Better results obtained for lower inference IoU threshold value on Dataset A as shown in Table 8 and Table 9 compared to higher IoU values, which can be attributed to localization errors that can occur due to the difference in the ground truth annotation boxes between train and test data. Moreover, we have noticed annotation errors (like those reported for Dataset B in our study) on Dataset A which can be attributed to poor results obtained for higher inference IoU threshold (Table 8 and Table 9). YOLO-mp-3l and YOLO-mp-4l outperformed other full versions of YOLOv4 models at 0.3 inference IoU threshold as shown in Table 8 and Table 9.

TABLE 7. Performance comparison of custom 3-layered and 4-layered models against standard full-YOLOv4 models trained and validated on 90-10 split Dataset B-centered, using IoU of 0.5 for training and a range of IoUs (0.3-0.6) for inference.

Model name	Transfer learning	mAP@0.3 IoU	mAP@0.4 IoU	mAP@0.5 IoU	mAP@0.6 IoU
YOLO-mp-3l	Yes	**92.13**	**92.13**	**91.99**	**91.50**
YOLO-mp-4l	yes	91.81	91.79	91.68	91.16
Full-YOLOv4 (standard)	yes	90.89	90.74	90.60	89.07
Full-YOLOv4-det104	yes	91.39	91.30	91.06	89.89
Full-YOLOv4-4l	Yes	91.37	91.26	91.00	90.25

TABLE 8. Performance of models trained on fold-5 of Dataset B-centered applied to infer on all images of Dataset A. All models have input resolution 608 × 608 pix and trained using 0.5 IoU threshold.

Model name	map@0.5	IoU@0.5	map@0.3	IoU@0.3
YOLO-mp-4l	62.26	51.09	**80.92**	55.83
YOLO-mp-3l	50.82	41.57	79.32	49.45
Full-YOLOv4	40.90	41.25	49.74	43.93
Full-YOLOv4-det104	61.30	55.82	74.38	59.14
Full-YOLOv4-4l	**62.59**	52.28	78.22	55.91

TABLE 9. Performance of models trained on split 90-10 of Dataset B-centered applied to infer on all images of Dataset A. All models have input resolution 608 × 608 pix and trained using 0.5 IoU threshold.

Model	map@0.5	IoU@0.5	map@0.3	IoU@0.3
YOLO-mp-4l	**66.0**	55.22	**82.01**	59.25
YOLO-mp-3l	65.98	55.38	81.27	59.16
Full-YOLOv4	58.37	54.91	74.04	58.65
Full-YOLOv4-det104	54.94	52.14	71.63	56.88
Full-YOLOv4-4l	57.30	51.91	73.46	56.37

9) MODEL SIZE, COMPUTATION, AND SPEED REQUIREMENTS

For a test on 118 images and network input resolution of 608 × 608 pixels the average detection-only time on a CPU computer with specification Intel®Core^{TM}i7-10700 CPU @ 2.90 GHz, 8 Cores, 16 Logical Processors were 11, 12 and 77 seconds for YOLO-mp-3l, YOLO-mp-4l and

full-YOLOv4 models, respectively. YOLO-mp-3l model is 7 times faster (93.22ms per image) compared to full-YOLO model (652.54ms per image) due to YOLO-mp-3l having fewer number of layers (51 layers) and require less computation (21.800 BFLOPS) compared to full-YOLOv4 model with 162 layers and 161.839 BFLOPS as shown in Table 10.

TABLE 10. BFLOPS of trained model and memory size.

Model name	BFLOPs	Model size (Mb)
YOLOv4-tiny-3l (original)	17.127	23.3
YOLO-mp-3l	21.800	24.5
YOLO-mp-4l	24.477	25.4
Full-YOLOv4 (original)	127.232	244
Full-YOLOv4-det104	161.839	244
Full-YOLOv4-4l	213.521	263

Larger model needs more storage space, longer downloading time and more computing resource such as processor and memory. Therefore, YOLO-mp-3l model is suitable for real-time mobile and embedded device applications. YOLO-mp-3l model's weight file size is 24.5MB which is about 10 times less than full-YOLOv4 model's weight size of 244MB as shown in Table 10.

VII. QUANTITATIVE ANALYSIS

Irrespective of diagnostic methods used for identification of malaria infection, all positive results should be accompanied by quantifying percentage parasitemia content in the blood films [4]. Trained models were used for detection and the number of detections per image was counted and assessed against the ground truth count of bounding boxes using regression analysis as shown in Table 11 and Table 12.

TABLE 11. Detection counts from models trained on trainset of fold-5 of Dataset B-centered applied to infer on validation set of fold-5 Dataset B-centered. All models have input resolution 608 × 608 pix and trained using 0.5 IoU threshold. Inference-time conf threshold of 0.25. "y" and "x" refers to predicted and ground truth counts, respectively.

	YOLO-mp-3l	Full-YOLOv4	YOLO-mp-4l
Regression equation	y=1.2295x	**Y=1.2214x**	y=1.2992x
R-squared	**0.932**	0.9304	0.9188
RMSE	3.242	**3.218**	3.923

Although, the confidence threshold value for each model should be determined separately to obtain best results, on a common confidence level of 0.25 and 0.5 the counts of pathogen from YOLO-mp-3l are similar to Full-YOLOv4 model as shown in Table 11 and Table 12.

With lower confidence threshold (Table 11) there is higher detection counts and higher RMSEs from all models compared to using higher confidence threshold (Table 12). At confidence threshold of 0.5 the counts form YOLO-mp-3l and Full-YOLOv4 are close to the ground truth counts with RMSEs of 2.163 and 2.190, respectively (Table 12).

TABLE 12. Detection counts from models trained on trainset of fold-5 of Dataset B-centered applied to infer on validation set of fold-5 Dataset B-centered. All models have input resolution 608 × 608 pix and trained using 0.5 IoU threshold. Inference-time conf threshold of 0.5. "y" and "x" refers to predicted and ground truth counts, respectively.

	YOLO-mp-3l	Full-YOLOv4	YOLO-mp-4l
Regression equation	y=1.0824x	**y=1.0332x**	y=1.1535x
R-squared	**0.9469**	0.9358	0.9404
RMSE	**2.163**	2.190	2.638

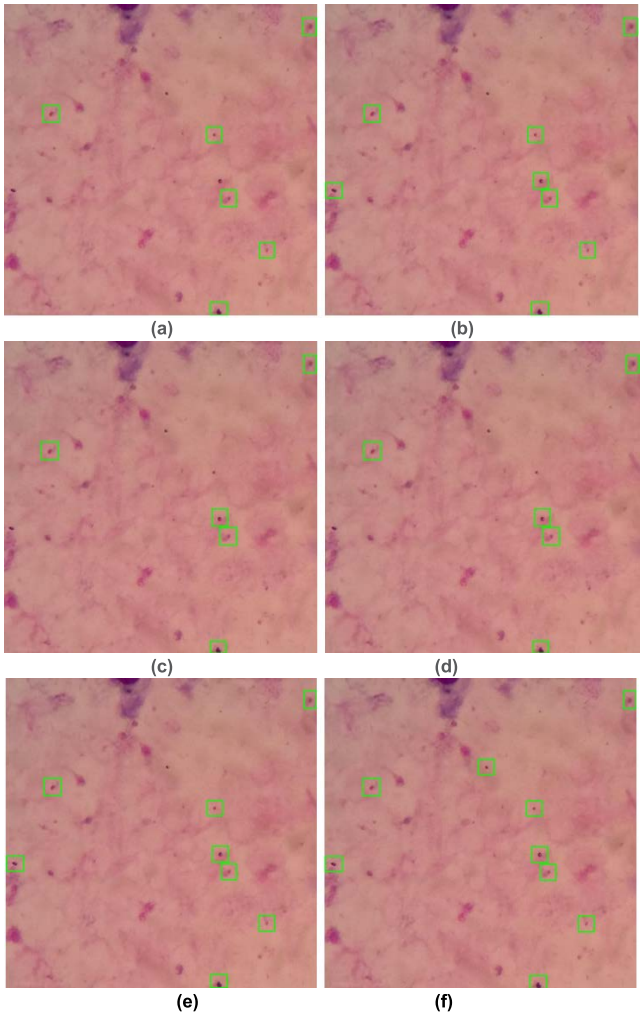

FIGURE 7. Pathogen detection results on image 0521.jpg for 0.5 confidence threshold left (a, c, e) and for 0.25 confidence threshold right (b, d, f) from YOLO-mp-3l, Full-YOLOv4, and YOLO-mp-4l models, respectively.

However, YOLO-mp-4l generated more detections than the other models compared (Table 11 and Table 12).

VIII. QUALITATIVE ANALYSIS

Models trained on fold-5 trainset were used to detect parasites on fold-5 validation set images of Dataset B-centered and bounding box visualized for qualitative analysis as shown in Figure 7-12.

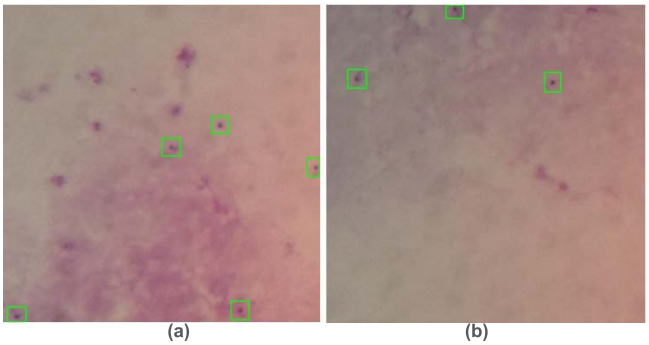

FIGURE 8. Pathogen detection results on image 0555.jpg (a) and image 0568.jpg (b).

Example detections on negative images:

Fold-5 validation set consisted of 24 negative images. There was no detection on most of the negative images from all 3 models YOLO-mp-3l, Full-YOLO, and YOLO-mp-4l. However, pathogens were detected on some negative images analyzed below.

In Figure 7, we have a negative image (0521.jpg), where YOLO-mp-3l, full-YOLOv4 and YOLO-mp-4l detected 6, 5, and 8 pathogens (left image conf thresh 0.50), and 8, 5, and 9 pathogens (right image conf thresh 0.25), respectively.

However, YOLO-mp-4l detected all the possible pathogens in Figure 7, that were either missed by YOLO-mp-3l or full-YOLOv4.

In Figure 8, we have negative images where all models (YOLO-mp-3l, full-YOLOv4 and YOLO-mp-4l) detected 5 pathogens in image 0555.jpg (left) and 3 pathogens in image 0568.jpg (right). For both images of Figure 8, the detections were on the same objects for detection confidence thresholds of either 0.5 or 0.25.

In Figure 9, we have a negative image (0540.jpg) in which YOlO-mp-3l, full-YOLOv4 and YOLO-mp-4l detected 2 objects each for with confidence threshold of 0.50. Although all models agreed on number of detections on images of Figure 9, but the detections were for different objects.

In Figure 10, we have a negative image (0562.jpg) in which all models (YOLO-mp-3l, full-YOLOv4 and YOLO-mp-4l) detected same object with confidence scores of greater than 90%. A zoomed-in view of the detected pathogen is provided in figure 10.

In Figure 11, we have a negative image (0537.jpg) in which YOLO-mp-3l, full-YOLOv4 and YOLO-mp-4l models detected 2, 2 and 3 objects at confidence threshold of 0.50 and 3,3 and 4 objects at confidence threshold of 0.25, respectively. However, in image of Figure 11 the detections from models are different objects.

The trained models from our study have picked up pathogen like objects in negative images having no boxes in the ground truth, but it still begs the question whether the detections were really a false detection or an ambiguity in ground truth labelling.

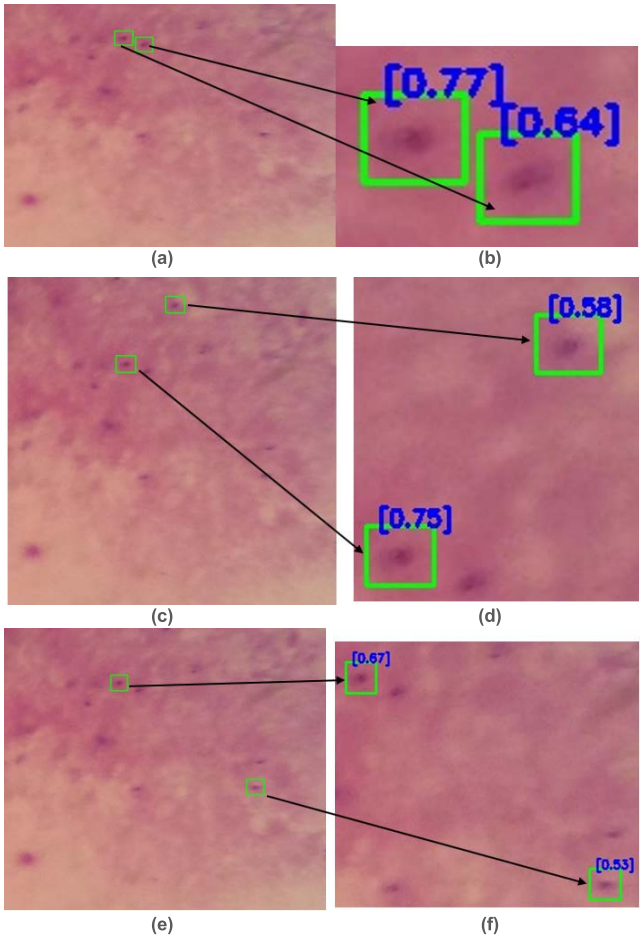

FIGURE 9. Pathogen detection results on image 0540.jpg left (a, c, e) and cropped detection with confidence score displayed right (b, d, f) from YOLO-mp-3l, Full-YOLOv4, and YOLO-mp-4l models, respectively.

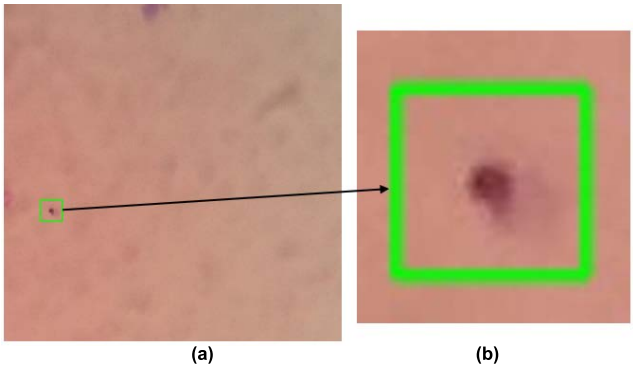

FIGURE 10. Pathogen detection results on image 0562.jpg left (a) and zoomed in display right (b).

Example detection on positive image:

We have considered as positive image where the difference between prediction and ground truth object counts was high.

In Figure 12, we have a positive image (546.jpg), where both YOLO-mp-3l and full-YOLOv4 models detected 8 pathogens where the ground truth count being

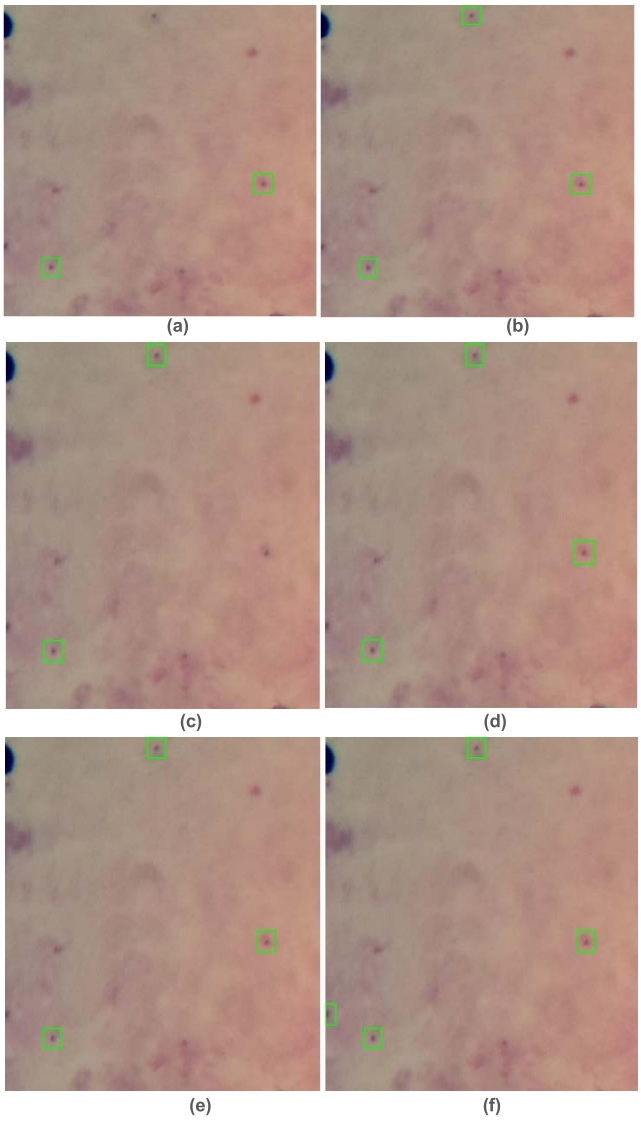

FIGURE 11. Pathogen detection results on image 0537.jpg for 0.5 confidence threshold left (a, c, e) and for 0.25 confidence threshold right (b, d, f) from YOLO-mp-3l, Full-YOLOv4, and YOLO-mp-4l models, respectively.

13 pathogens. However, YOLO-mp-4l model for the same image of Figure 12 detected all the 12 possible pathogens that were either missed by YOLO-mp-3l or full-YOLOv4.

Although the model prediction of number of infected pathogens can be very close to the ground truth number of objects in the image as shown in Table 11 and Table 12, the total count can have false positives counts. Qualitative analysis in Figure 7-11 showed that although, models detected similar number of pathogens for the image but few of the detections were from different objects. Therefore, we cannot judge the model performance only based on the detection counts. Average precision on the other hand considers the overlap of detected object with the ground truth box to determine true positives and is therefore a better metrics for model performance assessment.

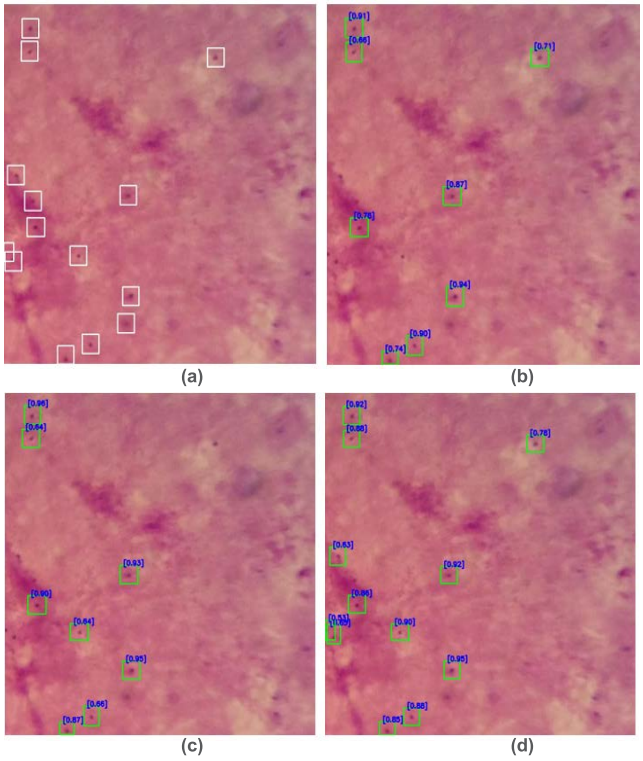

FIGURE 12. Pathogen detection results on image 0546.jpg for 0.5 confidence threshold (b, c, d) from YOLO-mp-3l, Full-YOLOv4, and YOLO-mp-4l models, respectively. Dataset B ground truth boxes for 0546.jpg (a).

IX. VALIDITY OF PUBLIC DATASET

Missing ground truth boxes and error in annotations were reported for Dataset B by [3]. Form the previous section on qualitative analysis of Dataset B, we observed our trained models were detecting more malaria pathogens that did not have ground truth annotation boxes. Interestingly, such detections were clearly observed on negative images of Dataset B.

We randomly selected eight images (0097.jpg, 0322.jpg, 0604.jpg, 0617.jpg, 0652.jpg, 0997.jpg, 1138.jpg, and 1149.jpg) from Dataset B but ensuring they include negative images as well as images with medium and high level of parasite infection as per the ground truth count of Dataset B. On those sampled eight images we draw bounding box on all possible pathogen-like objects (141 annotation boxes in total) and sent to our two independent expert microscopists ('expert-1' and 'expert-2') for scoring each box for presence or absence of malaria parasite. For comparison, existing Dataset B annotator was considered as 'expert-3'. Any boxes (out of 141 boxes annotated by us in this study) that were not labelled as pathogen in sampled images of Dataset B was considered as scored no-pathogen by expert-3.

To test the similarity in labelling between any two experts, we calculated the Jaccard similarity score between labelled sets using ''jaccard_similarity_score'' function of ''scikit-learn'' machine learning package (https://scikit-learn.org/stable; accessed on 20/06/2022). Jaccard similarity

index (0∼1) is a statistical measure of similarity between two sample sets. Jaccard similarity scores of 0.6, 0.52, and 0.59 were obtained for expert-3 vs. expert-1, expert-3 vs. expert-2, and expert-1 vs. expert-2, respectively. The low similarity scores between any of the two expert microscopists on sampled images of Dataset B indicates the high level of disagreement between experts' judgement. Therefore, if we need to deploy low-cost healthcare solution in pathogen detection through computer vision then there is also a need to establish standards and quality control on data sets.

X. DISCUSSION

Chibuta and Acar [3] studied the images on their test sets of Dataset B where the performance of trained model was very poor and reported that in most cases there were few parasites that were not ground truth labelled but detected by the trained model. Authors of [3] have also reported some annotation errors in randomly sampled images from the Dataset B. For supervised learning such subjective errors along with non-appropriate and loose bounding box where objects are not centered properly would significantly affect the model training and detection performance. In our study of dataset B for images containing tiny (1∼2 pixel) bounding boxes, it is unclear if the labelling expert missed to enlarge the boxes or that it was mistakenly annotated. Supervised learning mostly depends on the quality of training data i.e., correctness of labelling and consistency of drawing bounding box on ground truth target objects [9]. Similarly, on some images of dataset B we observed that few ground truth bounding box were unnecessarily large compared to other boxes in the same image. Moreover, some bounding boxes contained only part of the pathogen although the box were big enough to cover whole object.

For our new dataset we observed that in most cases the original bounding boxes from Dataset B were large enough in size to include the morphologies such as shape features around the centered chromatin on a re-adjusted dataset Dataset B-centered. Therefore, no attempt was made to resize the bounding box. However, it can be argued that if an expert microscopist can resize the boxes to properly fit around the object the overall quality of the dataset could improve.

Abdurahman *et al.* [20] reported that on Dataset B, the performance of YOLO models was sensitive to change in inference IoU threshold values. A 6.59% to 17.5% improvement in mAP was observed on Dataset B images for different YOLO models studied by changing IoU threshold from 0.5 to 0.3. Significant difference in mAP was observed on Dataset B for full-YOLOv4 model in our current study for two different inference threshold values as shown in Table 1, which is consistent to the results of [20].

In contrast, we observed that the performance mAP of YOLO models is not very sensitive to the change in IoU threshold values on Dataset B-centered as shown in Table 6 and Table 7 which indicates better object localization and better overlap between predicted and ground truth bounding box. We argue that the localization capability of models was improved on Dataset B centered, following chromatin centering approach.

YOLO-mp-4l achieved the best performance and was slightly better than YOLO-mp-3l as shown in Table 6. In contrast YOLO-mp-3l was slightly better than YOLO-mp-4l on 90-10 split dataset as shown in Table 7. Interestingly, YOLO-mp-3l for both scratch and transfer learning, outperformed full-YOLOv4 on fold-5 validation set of Dataset B-centered as shown in Table 6. This result proved that customized smaller models could achieve similar or even better performance in comparison to large standard models.

XI. RECOMMENDATIONS

The importance of model training and method development for very sensitive applications like medical imaging diagnosis should not solely rely on the model accuracies (numbers) reported on any publication but also on repeatability and reproducibility of model on different train/test sets such that the method will have some practical importance. It is recommended that the authors of publications release their train/test/valid set (especially for public datasets) and include all technical information on their model implementation clearly on the paper such that other person can re-produce the results for benchmarking of new models. We argue that the experts in medical imaging (e.g., microscopists) and machine learning should sit together to properly define and validate some rules for creating better quality datasets to improve reliability and accuracy of supervised models. It is recommended for authors of publications to explore and understand the dataset (especially public dataset) before crafting complex algorithms to solve the problems.

XII. CONCLUSION

Recommendation has been made from machine learning viewpoint and are based on the publicly available datasets of malaria pathogens in thick blood smears, about the sensitivity of the dataset, methods of testing repeatability and reproducibility of trained models and what information to be released in publications for future benchmarking. It was demonstrated that having a cleaner and consistently labelled dataset would allow to craft smaller and less complex models that can achieve similar or even better results than heavier and more complex models with low resource such as less computation and smaller hardware. A gap between experts of two domains medical and machine learning was identified for application under current study and recommendations are made to close the gap. A tiny model based on YOLO object detection framework was designed which achieved the best result with mAP 94% at 0.5 IoU threshold on our test set for the current dataset compared to heavier standard YOLO-v4 models studied.

In summary, this paper discusses and describes a method to increase labelling consistency for improved model accuracy. The improvement in model performance validated the

proposed method of centering the ground truth bounding boxes around chromatin. Future developments and large-scale validations of this can possibly address the existing knowledge-gaps; and could augment the current efforts of several health implementations programs on strengthening Malaria Microscopy across remote and rural areas where it is warranted most. The custom three-layered YOLO-mp-3l and four-layered YOLO-mp-4l models achieved the best mAP scores of 93.99 (@IoU=0.5) and 94.07 (@IoU=0.5), respectively outperforming standard YOLOv4 (full-YOLOv4 model, mAP 92.56 @IoU=0.5) for detection of malaria pathogen on a public dataset of thick smear microscopic images captured using phone camera. YOLO-mp-3l (BFLOPs = 21.8, model size =24.5Mb) and YOLO-mp-4l (BFLOPs=24.477, model size = 25.4Mb) outperformed full-YOLOv4 (BFLOPs=127.232, model size = 244Mb) in terms of computation and memory requirements proving them suitable to run on low resource settings.

Based on our results we can conclude that computer vision and deep learning methods can certainly help in automation of pathogen detection task, but there is a growing need for standardization on health care data set labelling.

REFERENCES

[1] N. Tangpukdee, C. Duangdee, P. Wilairatana, and S. Krudsood, "Malaria diagnosis: A brief review," *Korean J. Parasitol.*, vol. 47, no. 2, pp. 93–102, Jun. 2009, doi: 10.3347/kjp.2009.47.2.93.

[2] J. A. Quinn, A. Andama, I. Munabi, and F. N. Kiwanuka, "Automated blood smear analysis for mobile malaria diagnosis," *Mobile Point Care Monitors Diagnostic Device Design*, vol. 31, p. 115, Sep. 2014.

[3] S. Chibuta and A. C. Acar, "Real-time malaria parasite screening in thick blood smears for low-resource setting," *J. Digit. Imag.*, vol. 33, no. 3, pp. 763–775, Jun. 2020.

[4] B. A. Mathison and B. S. Pritt, "Update on malaria diagnostics and test utilization," *J. Clin. Microbiol.*, vol. 55, no. 7, pp. 2009–2017, Jul. 2017.

[5] D. Gamboa, M.-F. Ho, J. Bendezu, K. Torres, P. L. Chiodini, J. W. Barnwell, S. Incardona, M. Perkins, D. Bell, J. McCarthy, and Q. Cheng, "A large proportion of *P. falciparum* isolates in the Amazon region of Peru Lack *pfhrp2* and *pfhrp3*: Implications for malaria rapid diagnostic tests," *PLoS ONE*, vol. 5, no. 1, p. e8091, Jan. 2010.

[6] M. Gendrot, R. Fawaz, J. Dormoi, M. Madamet, and B. Pradines, "Genetic diversity and deletion of *Plasmodium falciparum* histidine-rich protein 2 and 3: A threat to diagnosis of *P. falciparum* malaria," *Clin. Microbiol. Infection*, vol. 25, no. 5, pp. 580–585, May 2019.

[7] R. Nakasi, E. Mwebaze, A. Zawedde, J. Tusubira, B. Akera, and G. Maiga, "A new approach for microscopic diagnosis of malaria parasites in thick blood smears using pre-trained deep learning models," *Social Netw. Appl. Sci.*, vol. 2, no. 7, pp. 1–7, Jul. 2020.

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1026–1034.

[9] M. Xu, Y. Bai, B. Ghanem, B. Liu, Y. Gao, N. Guo, X. Ye, F. Wan, H. You, and D. Fan, "Missing labels in object detection," in *Proc. CVPR Workshops*, vol. 3, 2019, p. 5.

[10] J. A. Quinn, R. Nakasi, P. K. Mugagga, P. Byanyima, W. Lubega, and A. Andama, "Deep convolutional neural networks for microscopy-based point of care diagnostics," in *Proc. Mach. Learn. Healthcare Conf.*, 2016, pp. 271–281.

[11] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.

[12] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.

[13] W. Liu, "SSD: Single shot MultiBox detector," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 21–37.

[14] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 7263–7271.

[15] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.

[16] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," 2020, *arXiv:2004.10934*.

[17] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," 2022, *arXiv:2207.02696*.

[18] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and W. Zisserman, "The PASCAL visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Sep. 2010.

[19] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014, pp. 740–755.

[20] F. Abdurahman, K. A. Fante, and M. Aliy, "Malaria parasite detection in thick blood smear microscopic images using modified YOLOV3 and YOLOV4 models," *BMC Bioinf.*, vol. 22, no. 1, pp. 1–17, Dec. 2021.

[21] A. Koirala, K. B. Walsh, Z. Wang, and C. McCarthy, "Deep learning for real-time fruit detection and orchard fruit load estimation: Benchmarking of 'MangoYOLO,'" *Precis. Agricult.*, vol. 20, no. 6, pp. 1107–1135, Dec. 2019, doi: 10.1007/s11119-019-09642-0.

[22] O. Russakovsky, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

[23] C.-Y. Wang, H.-Y. Mark Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, and I.-H. Yeh, "CSPNet: A new backbone that can enhance learning capability of CNN," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2020, pp. 390–391.

[24] D. Misra, "Mish: A self regularized non-monotonic activation function," 2019, *arXiv:1908.08681*.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1904–1916, Sep. 2015.

[26] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8759–8768.

[27] S. Kaewkamnerd, C. Uthaipibull, A. Intarapanich, M. Pannarut, S. Chaotheing, and S. Tongsima, "An automatic device for detection and classification of malaria parasite species in thick blood film," *BMC Bioinf.*, vol. 13, no. S17, pp. 1–10, Dec. 2012, doi: 10.1186/1471-2105-13-S17-S18.

ANAND KOIRALA received the B.Eng. degree in electronics and communications from Tribhuvan University, Nepal, the Master of Engineering Science degree in electrical and electronics from the University of Southern Queensland, Australia, and the Ph.D. degree from CQUniversity, in 2020. He is currently working as a Research Officer with the School of Health, Medical and Applied Sciences, CQUniversity, Australia. He has provided machine vision solutions and consulting services for Australian industries (horticulture, aquaculture, medical, and automotive). His research interest includes the developing of computer vision decision support systems based on deep learning and AI.

MEENA JHA received the bachelor's and master's degrees in electronics and communication engineering and computer science in India and the Ph.D. degree in computer science and engineering in Australia. She has more than 20 years of experience in teaching and research. She is currently a Computer Science Academician with the School of Engineering and Technology (SET), Central Queensland University, Australia. She is also an Award Winning Academician and a Researcher in the area of education, virtual reality, artificial intelligence, learning analytics, and STEM. She was a recipient of the Vice Chancellor's Learning and Teaching Award at Central Queensland University. She has authored or coauthored many peer-reviewed papers in national and international journals, edited book chapters, and conference proceedings. She has completed many projects on virtual reality, artificial intelligence, and science, technology, engineering and math (STEM) in education. She has supervised many Ph.D. and master's students by research. She is a Co-Founder and the President of WinTECH Society at Central Queensland University, Sydney Campus, addressing gender inequality.

SRINIVAS BODAPATI received the B.Tech. degree (Hons.) in electrical engineering from the Indian Institute of Technology Kharagpur, India, and the Ph.D. degree in electrical and computer engineering from the University of Illinois at Urbana-Champaign, USA, in 2003. Since then, he has been with Intel Corporation, Santa Clara, CA, USA, working on statistical and machine learning methods for pre-silicon analysis and modeling of circuit reliability. His research interests include high-dimensional modeling and analysis, rare event prediction, machine learning algorithms in classification, clustering, and anomaly detection.

ANIMESH MISHRA (Member, IEEE) is currently a Distinguished Platform Architect with NVIDIA Corporation, Santa Clara, CA, USA, where he is involved in futuristic multi-core platform implementations. Prior to that, he spent almost 21 years at Intel Corporation, where he led several path finding and commercially successful products and technologies in the areas of RISC-V, several multi modal edge analytics SoCs transforming Intel's new generation platform Evo, three generations of Intel RealSense Stereo Depth Camera and strengthening Functional Safety (FuSa) implementation for automotive and IoT products. He was the Chief Architect for silicon and systems for computer vision (CV)/machine learning video analytics co-processor at Intel Corporation. He also held roles as a Principal Power Management and System Performance Architect for multiple generations of highly successful mobile platforms. He is credited with over 20 successful tapeouts, 45 patents granted or in pipeline, including Intel's key patent on Core Hopping technology for modern multicore systems. He has collaborated with Stanford University and USC for research/implementation in video analytics and AI/ML technologies. He is a member of IEEE Consumer Technology Society.

GIRIJA CHETTY (Senior Member, IEEE) received the bachelor's and master's degrees in electrical engineering and computer science in India and the Ph.D. degree in information sciences and engineering in Australia. She has more than 35 years of experience in industry, research, and teaching from universities and research and development organizations in India and Australia. She has held several leadership positions, including the Head of Software Engineering and Computer Science, the Program Director of ITS Courses, and the Course Director of Master of Computing and Information Technology Courses. She is currently a Full Professor in computing and information technology with the School of Information Technology and Systems, University of Canberra, Australia, and leads a research group with several Ph.D. students, post-docs, research assistants, and regular international and national visiting researchers. She is a Senior Member of Australian Computer Society and a member of ACM, and her research interests include the area of multimodal systems, computer vision, pattern recognition, data mining, and medical image computing.

PRAVEEN KISHORE SAHU received the B.Sc. and M.Sc. degrees in biology and biotechnology from Sambalpur University, in 2002 and 2004, respectively, and the Ph.D. degree in biotechnology with specialization on molecular microbiology and genomics from the University of Pune, Maharashtra, India, in 2012.

He is currently working with the Community Welfare Society Hospital (CWSH), Rourkela, and Ispat General Hospital (IGH) Rourkela, Odisha, India. He is also posted as a Visiting Professor in biotechnology with Sambalpur University, Odisha, with the teaching and active supervision of Ph.D. students in biotechnology. He is also a Senior Researcher (Molecular Biologist) in Odisha. He began working on malaria, since 2005, on the genetic diversity and drug resistance genes of the most virulent human malaria parasite *Plasmodium falciparum* at IGH Rourkela. His major areas of research and publications have focused on understanding the pathogenesis of *P. falciparum* in severe and cerebral malaria patients and the mechanisms exploring potential adjuvant therapy using a host of molecular, immunologic, cell-based, and machine-learning modeling approaches. He has also been involved with the genotype-phenotype characterizations of hemoglobinopathies and thalassemias in Indian patients; besides the genomics of multi-drug resistant (MDR) medical biofilms to explore novel biocontrol measures.

SANJIB MOHANTY received the M.B.B.S. and M.D. degrees in medicine from SCB Medical College and Hospital, Cuttack, Odisha, India, in 1978 and 1983, respectively.

He is currently a Physician and a Researcher in Odisha, and the former Director of Ispat General Hospital (IGH) Rourkela, Odisha. He has made major contributions in malaria research in India since past 35 years. He led several clinical trials on new antimalarials and adjuvants at IGH Rourkela for severe malaria treatment and management. He was a former WHO Faculty for "Management of Severe Malaria" for training doctors from South-East Asia; and a Senior Member of the Expert Group on National Drug Policy on Malaria, Government of India; and an Expert Member of the Indian Council of Medical Research (ICMR) Consortium Malaria Elimination Research Alliance (MERA-INDIA) for evaluating scientific proposals on malaria elimination in India. His current area of research is to unravel the pathophysiology of cerebral malaria and brain swelling using neuroimaging like MRI brain-scanning. His research led to the discovery of a novel phenomenon called posterior reversible encephalopathy syndrome (PRES) in cerebral malaria patients in India for the first time, and more recently, the silent features of severe non-cerebral malaria in Indian patients that could open new therapeutic avenues for cerebral malaria patients.

TIMIR KANTA PADHAN received the bachelor's degree in pharmacy and the master's degree in pharmacology from the Biju Patnaik University of Technology (BPUT), India. He is currently working as a Project Coordinator with the Community Welfare Society Hospital, Rourkela, Odisha, India. He has worked for several pharma industries in India in formulation and development section for the better quality and cost effective of medicine to the society. His main focus and research interest is to solve the rural and tribal health problem and give them a better life in the society.

JYOTI MATTOO is working as the Director of Intel AXG/XCE Business Group. She has been instrumental in building SoC design teams at Intel for Intel's IP silicon verification, securing multiple first-time silicon success for Intel products. In addition to SoC design, she is also responsible for operations management at Intel. She started her career at STMicroelectronics, in 1998, where she was primarily responsible for the SoC development of set-top boxes and other STMicroelectronics' SoC devices. She has also instrumental in building a silicon design ecosystem in UAE working for Dubai Circuit Design. Here, her role was to generate design services' business while interacting with product companies in the USA. She then moved to AMD, India, heading SoC design for AMD's GPU chips and later to South Korea working with Synopsys for GPU chipset professional services.

AJAT HUKKOO started his career at Intel in 1990, and he left for Broadcom, in 2000, where he was responsible for the cellular SoC development of 2G, 3G, and 4G/LTE SoC devices, shipping 300M units in his career at Broadcom. He joined Intel again, in 2014, and drove the nascent BaseIA technology to multiple A∗ PRQ's with Spreadtrum. He has held roles in the Design Development Group, running the Modem Engineering Group (MEG), and in 2019 became the Director of Advanced Technology Development (renamed to Emerging Technologies Engineering Group) in CIG. XCC is on a mission to bring in >$2B in revenue in five years by going after markets, such as blockchain, edge visual and super compute, and advanced technology design services. He is currently the Vice President of AXG and the General Manager of Accelerated Custom Engineering (XCE) focusing on blockscale and fully homomorphic encryption. He has been instrumental in charting new design and innovation since 2019, spearheading Intel's blockchain business efforts securing multiple first tape-in successes in record time. We expect this business to cross $150M in 2022 and is expected to grow at a much steeper rate in 2023. He has also nurtured DPRIVE into a valuable technology showcase and will now be responsible for growing it further.