



# COMPARISON OF FASTER-RCNN vs YOLOv3 FOR MALARIA DETECTION AND COUNTING

Dewi Endah Kharismawati

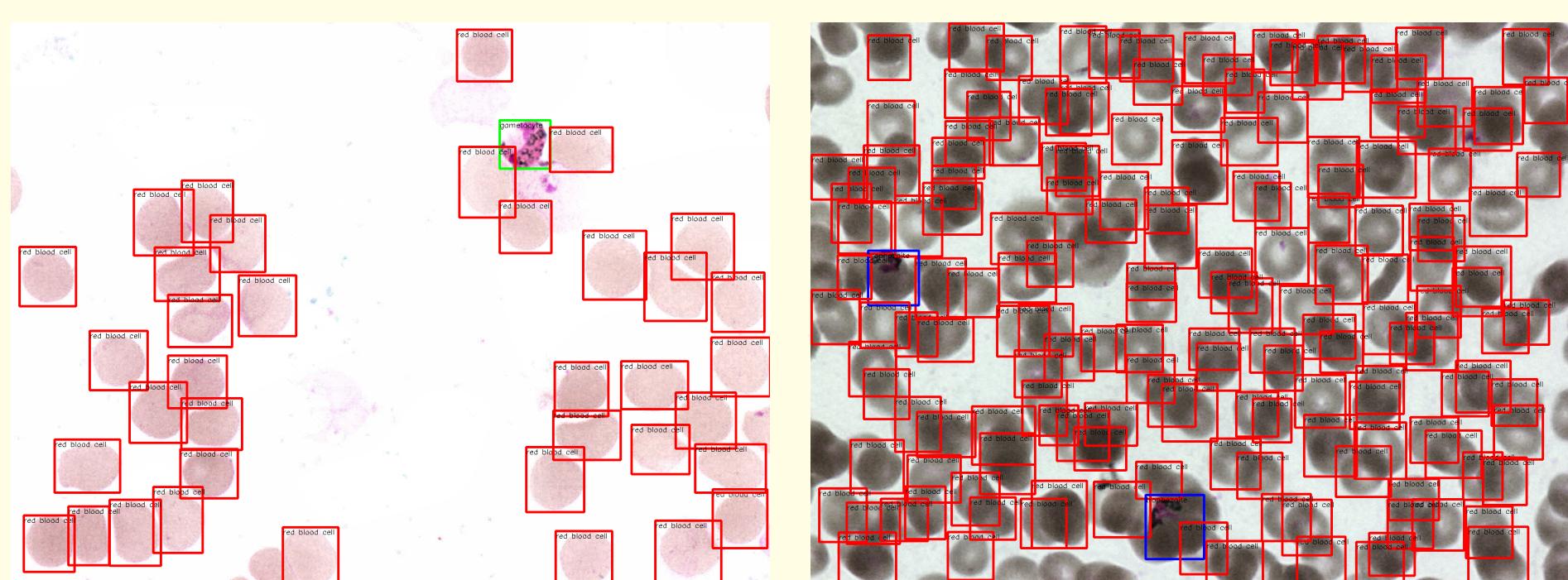
Department of Electrical and Computer Engineering,  
University of Missouri, Columbia, U.S.A.

Malaria is a life threatening disease caused by Plasmodium parasite that is transferred to human by infected female Anopheles mosquito [1]. Malaria has reached epidemic, especially in developing country. It kills approximately 400,000 people each year. To diagnose, a clinician examine thin blood smear that stained with contrasting agents under a microscope and visually searching and counting for infected cells (~5000 cell samples), an extremely time consuming, labor intensive, and challenging process. There are multiple factors to be considered, such as variations of cell shape, density, color, and some more uncertainty. On the other hand, deep learning based architecture have been great success in object detection. We use faster Region-based Convolution Neural Network (Faster R-CNN) and You Only Look Once version 3 (YOLOv3), two of the top performing object detection architectures in recent years. This project goal is to compare which architecture can provide a more accurate and efficient model to detect and count Malaria infected cell.

## Method Overview

### Dataset

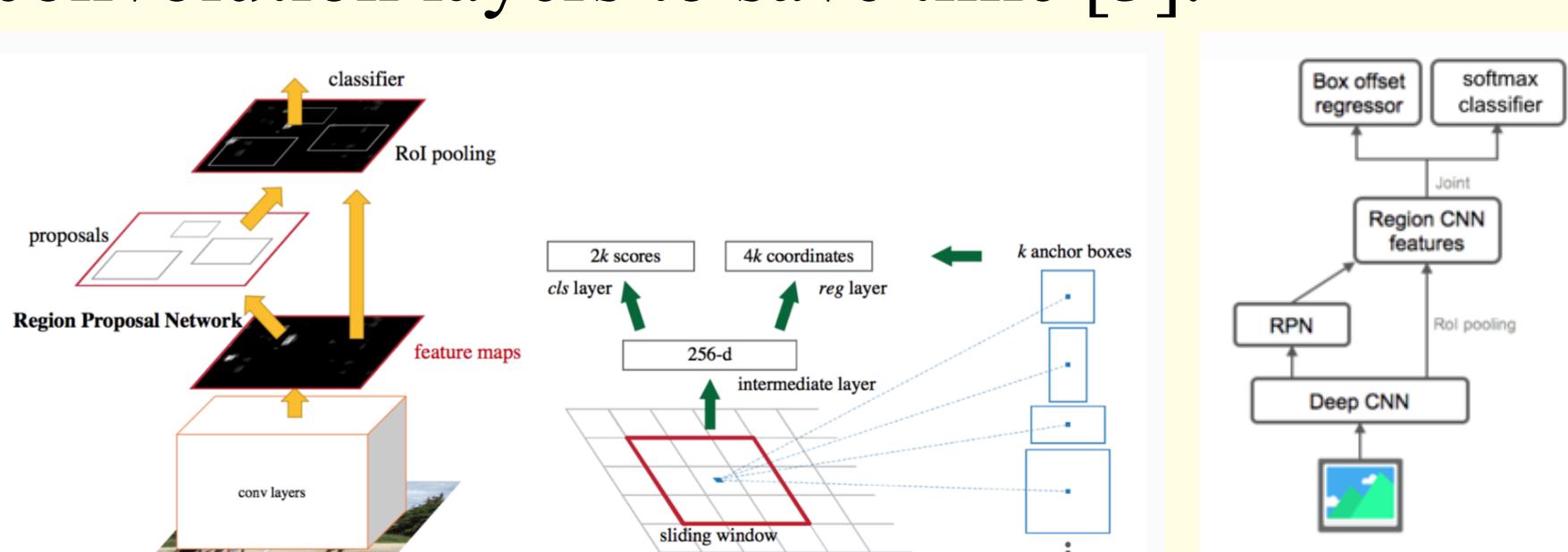
Dataset used is from Broad Bioimage Benchmark Collection [2]. It contains 1062 images for training and 267 images for testing. There are overall 7 classes with ground truth.



The dataset is naturally imbalanced over the object classes. In training, uninfected red blood cell is dominated with 97%, which lead to some challenge in training.

### Faster-RCNN

Faster-RCNN is the third evolutions of RCNN. It takes an image as input and produce bounding boxes around objects of interest with their labels. Faster-RCNN takes an input image to 2 modules: region proposed network (RPN) that proposed some high confidence regions and object detector adapted from Fast-RCNN to classify the region proposed. Both module shared the same convolution layers to save time [3].



Multitask loss function combines losses of classification and bounding box regression

$$\mathcal{L} = \mathcal{L}_{cls} + \mathcal{L}_{box}$$

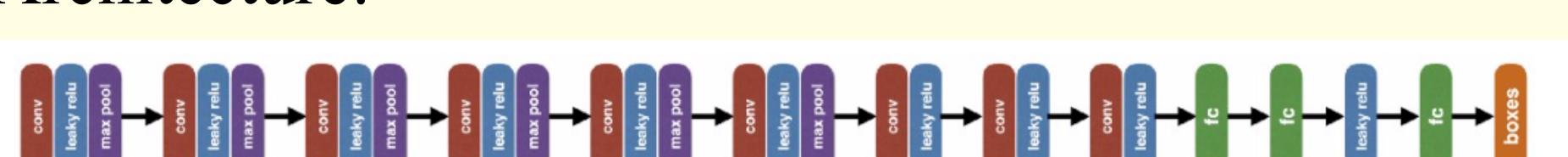
$$\mathcal{L}(\{p_i\}, \{t_i\}) = \frac{1}{N_{cls}} \sum_i \mathcal{L}_{cls}(p_i, p_i^*) + \frac{\lambda}{N_{box}} \sum_i p_i^* \cdot L_{smooth}(t_i - t_i^*)$$

$$\mathcal{L}_{cls}(p_i, p_i^*) = -p_i^* \log p_i - (1 - p_i^*) \log(1 - p_i)$$

### YOLOv3

Yolo uses only a single CNN network for both classification and bounding boxes localizing. The input image is divided into an SxS grid. Each grid responsible for predicting the object, the cell where the center of an object falls into [4].

Architecture:



Loss Function:

$$\text{Classification} \quad \sum_{i=0}^{S^2} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2$$

Localization

$$\lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{obj} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2] + \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{obj} [(\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2]$$

Confidence loss

There is object in the box

$$\sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{obj} (C_i - \hat{C}_i)^2$$

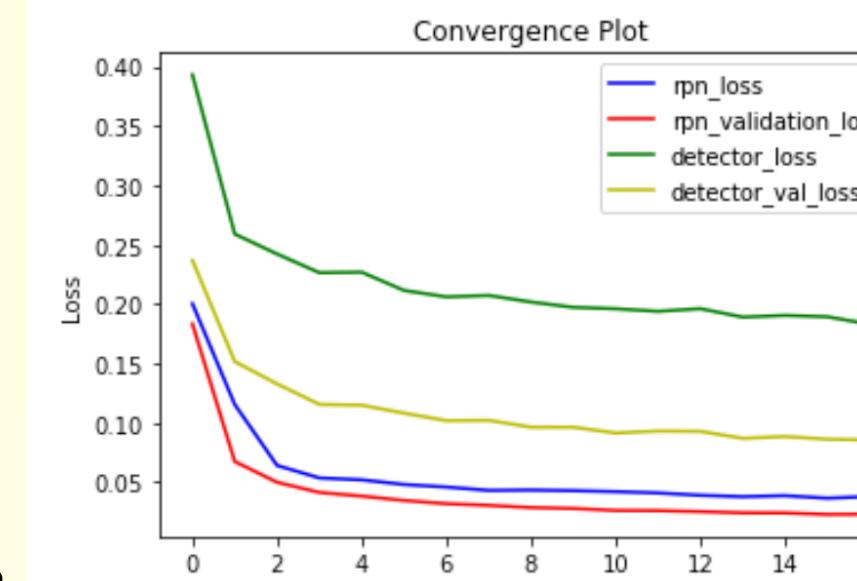
No object in the box

$$\lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{noobj} (C_i - \hat{C}_i)^2$$

## Experiment and Result

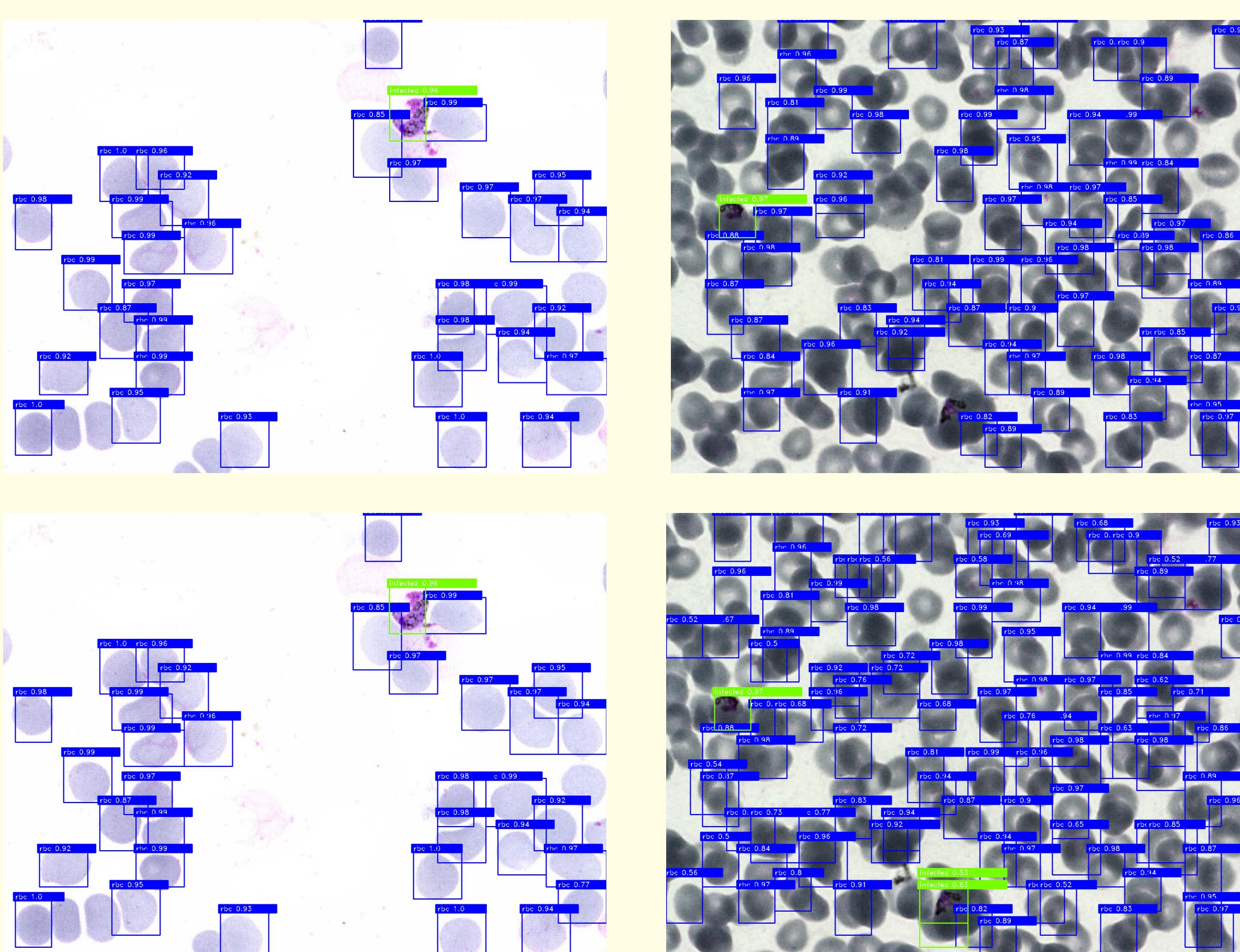
### FASTER-RCNN [5]

Training dataset was 1062 images, divided into training set of 902 and validation set of 160. The backbone is using resnet. It was optimized with Adam with learning rate 0.00001. Batch training with 16 successful epoch, each epoch has 1000 steps. The training has to stop after 25 hours.



Testing There are 266 image tested using retrained model, with 16,615 count of red blood cell and 616 count of infected cell. Each image feed into the network outputs bounding boxes with its class prediction and confidence value. I experiments on threshold of 0.8 and 0.5. The result was somewhat good. Below is the matching matric for both threshold. Overall 0.8 threshold achieve F1 score of 63.2%, whereas 0.5 achieve 67.3%. This is better compare to Hung, J [8] result, that achieve 59%.

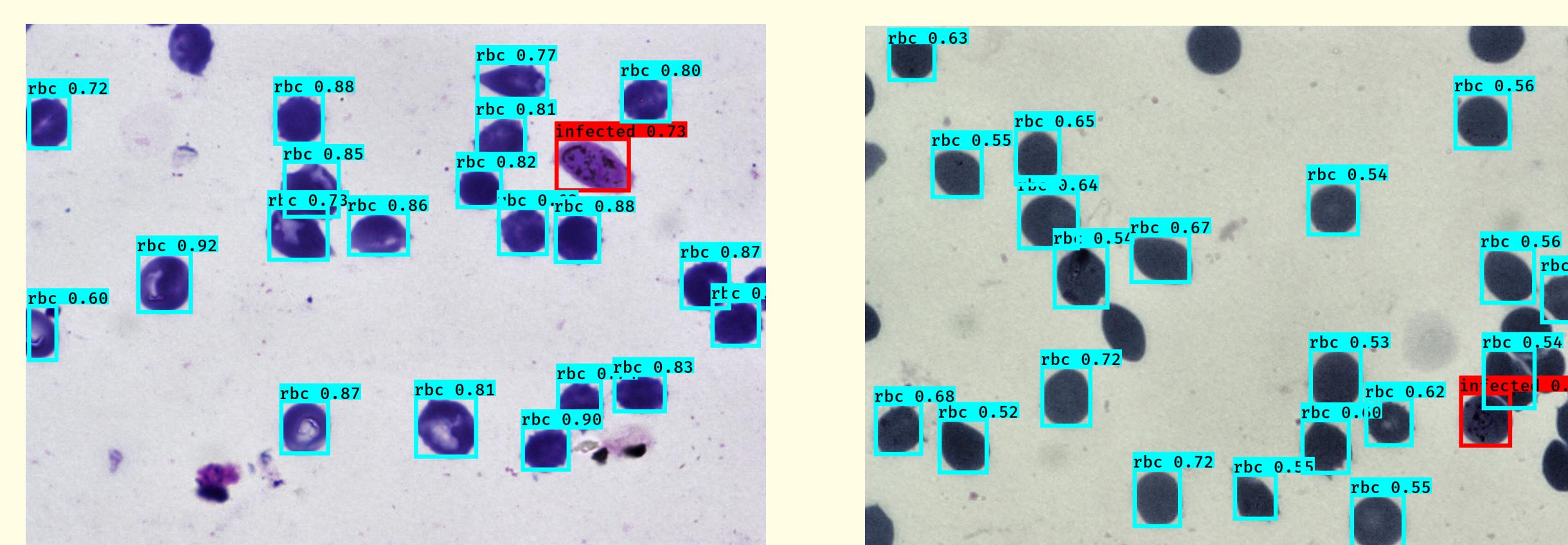
	Model Count	Ground Truth		Model Count	Ground Truth
RBC	10,262	16,615	RBC	12,452	16,615
Infected Cell	624	616	Infected Cell	776	616



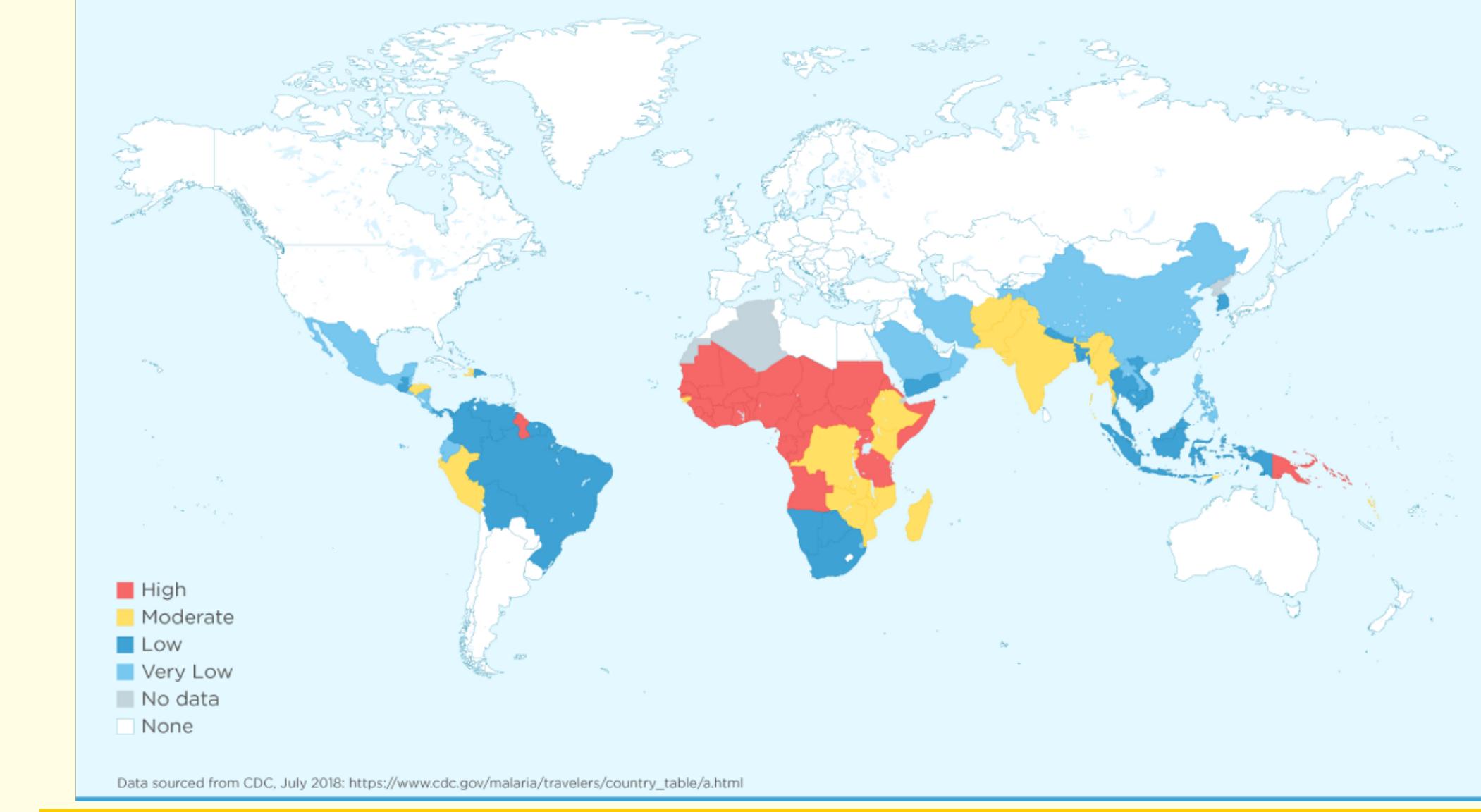
### YOLOv3 [6]

Training training dataset was 1062 images in total, 827 for training and 91 for validations. With batch size of 32, it has 29 steps per epoch. The network stops at epoch 381 after 25 hours (same constraints time as Faster-RCNN). Training is using adam optimizer with learning rate of 0.0001.

Testing There are 266 image tested using retrained model, with 16,615 count of red blood cell and 616 count of infected cell. Each image feed into the network outputs bounding boxes with its class prediction and confidence value. The confidence values turns out quite low, so I set the threshold to be 0.5



With F1 score of 61.2%

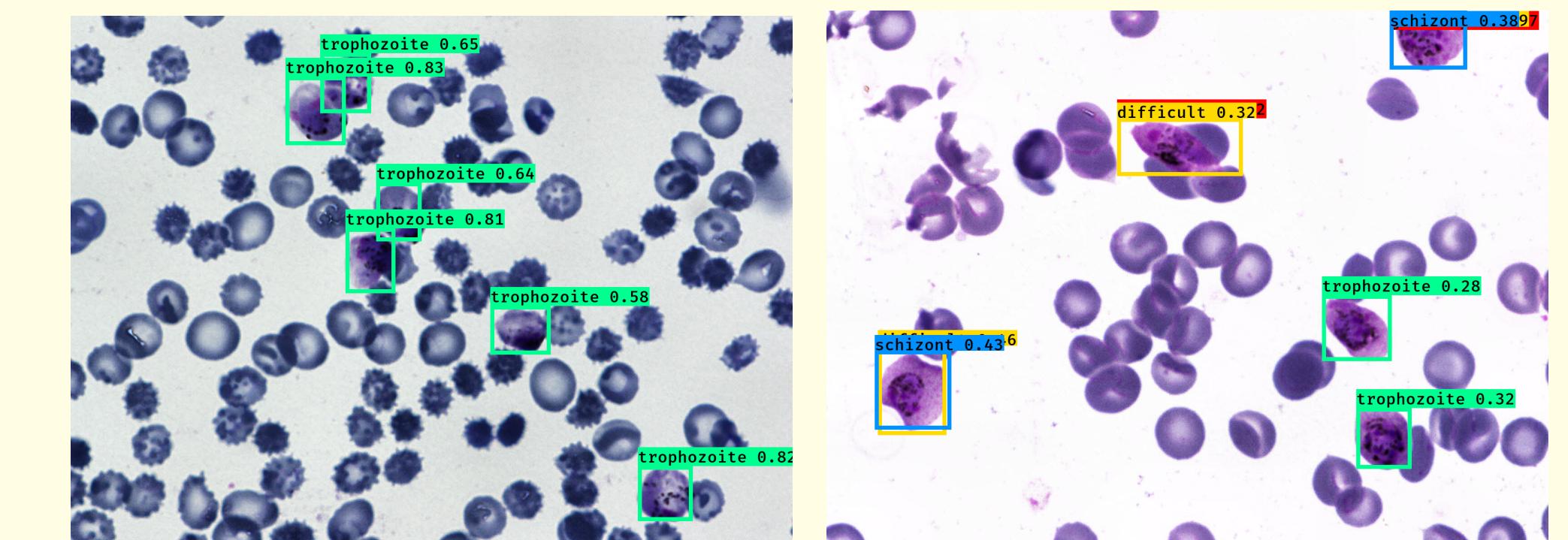


## Result

### Infected Only

In addition, I had a curiosity on how does these algorithm perform on detecting only the infected cells. So, I trained the network using YOLOv3 with identical configuration, except the classes.

It turns out that the result is pretty good.



	YOLO	GT	Model Count	Ground Truth Count
Trophozoite	447	561	trophozoite	664
Schizont	36	28	schizont	39
Ring	103	88	ring	227
Gametocyte	37	75	gametocyte	74
Leukocyte	27	28	leukocyte	49
Difficult	155	217	difficult	0

Compare to work by Hung, J [8] using Faster-RCNN, YOLOv3 result is actually perform better. In here, we are over detecting on ring and schizont, based on result that I examine, the difficult class is contribute to this error. When difficult is detected, they are sometimes overlap with schizont and ring at the same Location.

With this result, I was hoping that I can build a second layer of convolution network. We can do Faster-RCNN as the first infected and uninfected cells. Then, send the infected cells to YOLOv3 to detect what stage of plasmodium.

## Conclusion and Future Work

In conclusion, we have compare Faster-RCNN and YOLOv3 to detect and count malaria plasmodium vivax. Both methods are not very efficient and accurate to do the task. To detect between infected and uninfected cells, Faster-RCNN performs better compare to YOLOv3. This experiment is not inline with the theory. Based on the claim, YOLO should perform better. After reading some paper, this is because YOLO has a drawback when the object is overlap with each other, which in this malaria dataset it happens al lot. This problem also encountered in Faster-RCNN. Therefore, both architecture fails to detect a lot of uninfected red blood cell. However, to detect only infected red blood cell, YOLOv3 performs really good, it is even better compare to Faster-RCNN. Therefore, if I get a chance, my future work would be combining both Faster-RCNN and YOLOv3 to detect the whole uninfected and plasmodium stages.

[1] World Health Organization: World Malaria Report(2018)

[2] Malaria dataset retrieved from <https://data.broadinstitute.org/bbbc/BBBC041/>

[3] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems (pp. 91-99).

[4] Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767.

[5] Faster-RCNN code [https://github.com/jinfagang/keras\\_frcnn](https://github.com/jinfagang/keras_frcnn)

[6] YOLOv3 code <https://github.com/qzwvee/keras-yolo3>

[7] Hung, J., & Carpenter, A. (2017). Applying faster R-CNN for object detection on malaria images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (pp. 56-61).