MALARIA DETECTION USING LOW-RESOLUTION MICROSCOPES WITH DEEP LEARNING

My code is available on GitHub (https://github.com/maximevo/malaria). Please do not share the GitHub repository nor this report, since the project is in the process of being published.

1 Introduction

Malaria is a mosquito-borne infectious disease affecting 200 million humans world-wide and responsible for more than 400,000 deaths every year [1]. In Africa, a child dies almost every minute from malaria [2]. The WHO World Malaria Report 2015 [3] estimates that 3.2 billion people in 95 countries are at risk of being infected with malaria, and 1.2 billion are at high risk (over 1 in 1000 chance of getting malaria in the next year). Previous efforts to reduce malaria mortality have faced two major barriers. First, detecting malaria parasites in blood samples is traditionally done by fluorescence microscopy. This technique relies on expensive ($15,000) microscopes to create high-resolution images (1000$\times$ of magnification) of blood samples. Second, these high-resolution images need to be analyzed by a medical expert to identify the presence or absence of malaria parasites. Access to medical experts is constrained in some low-resources regions affected by malaria. Even in regions with unconstrained access to medical experts, microscopic diagnostics are not always standardized and depend heavily on the experience and skill of the microscopist. This leads to incorrect diagnostic decisions in the field [1].

Recent work[1] tackled the first barrier to malaria diagnosis. Indeed, Hongquan Li[2], Manu Prakash[2] & al proposed a novel technique to identify malaria parasites using cheap ($200), low-resolution (100$\times$ of magnification) microscopes[1]. The authors discovered that if fluorescent blood samples are examined using a microscope equipped with RGB cameras - instead of monochromatic cameras - blood platelets appear slightly blue whereas malaria parasites appear slightly green. This phenomenon is referred to as a spectral shift. Leveraging this spectral shift, it is possible to distinguish platelets from malaria parasites based on color. It is thus no longer necessary to look at the DNA of platelets and malaria parasites to distinguish them. As a result, it is possible to distinguish platelets from malaria parasites using cheap, low-resolution microscopes. This discovery removes the first barrier to malaria diagnosis - medical experts can now detect malaria parasites by using cheap, low-resolution microscopes.

In this paper, we partner with the aforementioned authors[2]. We aim to build on top of their previous work in order to alleviate the second barrier to malaria diagnosis - providing accurate diagnostic information in places with constrained access to medical experts. To do so, we develop an algorithm which automatically detects malaria parasites in low-resolution images of blood samples.

2 Data

Our goal is to detect malaria parasites in blood samples. Each blood sample is examined using a low-resolution microscope which generates two images. The first image (Figure 1) is a low-resolution microscopic view of the blood sample, before adding fluorescent agents to the blood sample. The second image (Figure 2) is a low-resolution microscopic view of the blood sample, after adding fluorescent agents to the blood. Red blood cells are visible in the first, non-fluorescent image (Figure 1). Platelets and malaria parasites are visible in the second, fluorescent image (Figure 2). The lab provided 15,000 blood samples. The blood samples are split into a training, validation and test set. We ensure that there is no subject overlap between these sets.

3 Approach Overview

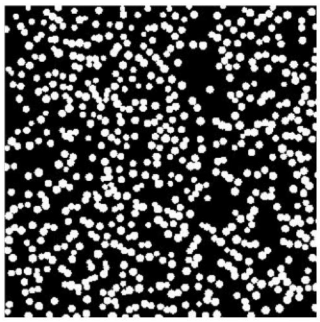
For each blood sample, our goal is to detect all malaria parasites. To do so, we aim to distinguish malaria parasites from platelets in the fluorescent image. Our approach consists of two stages.

In the first stage, we focus on the non-fluorescent image (Figure 1). We attempt to segment red blood cells using deep learning. The output of this first stage is a segmentation map of red blood cells. It will be used in the second stage as a

[1] Paper in the process of being published

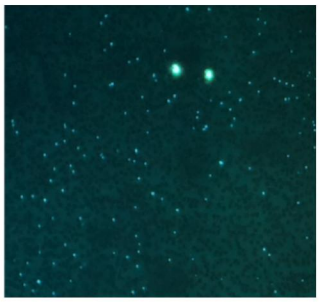
[2] Department of BioEngineering, Stanford University

(a) Raw microscopic view

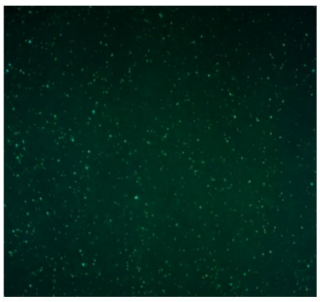


(b) Segmentation mask of red blood cells

Figure 1: Microscopic view of a blood sample, before adding fluorescent agents to it. Our goal is to segment red blood cells in this low-resolution image.



(a) Spectral shift leads blood platelets to appear slightly blue



(b) Spectral shift leads malaria parasites to appear slightly green

Figure 2: Microscopic view of a blood sample, after adding fluorescent agents to it. Our goal is to distinguish malaria parasites from platelets in this low-resolution image.

feature to distinguish platelets from malaria parasites. Indeed, malaria parasites tend to penetrate inside red blood cells, whereas platelets do not.

In the second stage, we focus on the fluorescent image (Figure 2). Each fluorescent cell is either a platelet or a malaria parasite. We attempt to distinguish platelets from malaria parasites. To do so, we crop the fluorescent image into small patches centered around each fluorescent cell. We extract expert features about the size, shape and color of the fluorescent cell at hand. We then combine these visual features extracted from the fluorescent image, with the segmentation mask of red blood cells - computed in the first stage. This equips us with two sets of features to classify whether each fluorescent cell is a platelet or a malaria parasite.

4 Segmentation of Red Blood Cells (Stage 1)

In this section, we describe the models trained for the first stage of our system - segmentation of red blood cells.

4.1 Problem Formulation

Our goal is to segment non-fluorescent images to identify red blood cells (Figure 1). For each non-fluorescent image X, our segmentation model outputs a binary mask Y indicating whether each pixel corresponds to a red blood cell. We optimize the binary cross-entropy objective function

$$\mathcal{L} = -\sum_{i=1}^{N}\sum_{h=1}^{H}\sum_{w=1}^{W} Y_{h,w}^{(i)} \log p(Y_{h,w} = 1 \mid X^{(i)}) + (1 - Y_{h,w}^{(i)}) \log p(Y_{h,w} = 0 \mid X^{(i)}),$$

where i is the index of the non-fluorescent image (there are N images in total), where (h, w) is the position of the pixel,

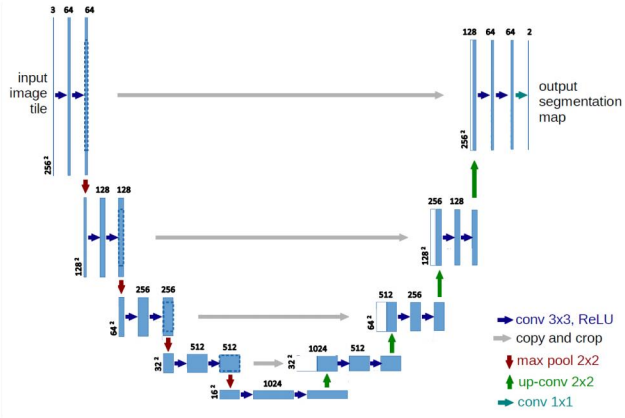


Figure 3: The architecture of the network. The network consists of 19 layers of convolution followed by a sigmoid.

and where $p(Y_{h,w} = 1 \mid X^{(i)})$ is the probability that the network assigns to label l given the input non-fluorescent image $X^{(i)}$.

4.2 Data Preprocessing and Class Imbalance

Non-fluorescent blood images (Figure 1) originally have a size of 820×820 pixels. We split them into multiple smaller images of size 128×128 pixels. The 128×128 images are provided as inputs to the neural network for segmentation. This reduces the dimensionality of the inputs and lowers the number of parameters of the neural network, while maintaining the resolution of the input image. Besides, in terms of class imbalance, it is worth noting that 42% of the pixels of the test images correspond to red blood cells.

4.3 Architecture Search and Parameter Turning

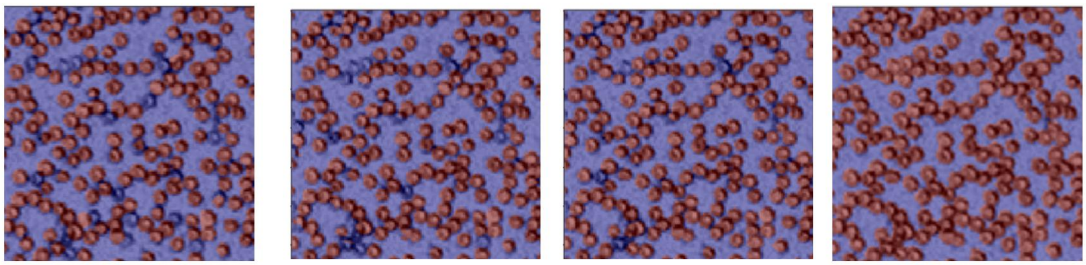
The segmentation network is a 19-layer U-Net [4] convolutional network. The architecture is analogous to that of an encoder-decoder. The contracting path consists of 2×2 max-pooling layers. It captures context by learning a compact representation of the data. The expansive path consists of 2x2 up-convolution layers. It enables precise localization, by retaining spatial information in spite of the max-pooling performed in the contracting path (Figure 3). We experimented with other architectures, such as Fully Convolutional DenseNets architectures [5], and did not observe significant performance differences. We train the network from scratch and use He initialization for the weights of the convolutional layers [6]. We use the Adam optimizer [7] with the default parameters and minibatches of size 8. These hyperparameter choices are commonly adopted by previous works in the field of segmentation [4, 5]. The optimization of the neural network was tricky and we therefore had to fine-tune the learning rate carefully. The band of learning rates leading to the best neural network optimization was between 1e-6 and 5e-5. It is also worth noting that we fine-tuned the number of convolutional layers in the neural network. The choice of 19 layers was a compromise between segmentation accuracy and inference time on CPU. Indeed, the end goal is have the neural network running in real-time on portable CPUs in remote areas, therefore we decided to opt for relatively shallow neural networks. Finally, we save the best model as evaluated on the validation set during the optimization process.

4.4 Data augmentation

In this section, we explain how we obtain segmentation labels for our dataset. We run a Hough Transform [8] algorithm to automatically segment red blood cells in the blood samples. The Hough Transform provides relatively accurate labels for blood samples where red blood cells are round and do not overlap. However, the Hough Transform provides noisy labels for blood samples where red blood cells are not round or overlap.

Our deep learning algorithm is trained on these labels. Hence, it segments well blood samples where red blood cells are round and do not overlap, but it fails to segment blood samples where red blood cells are not round or overlap.

We improve the quality of our automatic data labeling by using data augmentation on the training set. Indeed, rather than running a noisy Hough Transform on blood samples where red blood cells are not round or overlap, we opt for the following procedure. First, we run an accurate Hough Transform on blood samples where red blood cells are round and do not overlap. Then, we distort both the image and the accurate segmentation map. This provides an accurate segmentation map for distorted blood samples, where red blood cells are not round anymore. Besides, we superpose multiple accurate segmentation maps, in order to obtain an accurate segmentation map for blood samples where red



(a) Hough Transform, used for automatic labeling, leads to many false negatives

(b) Deep Learning without data augmentation nor without pixel weights

(c) Deep Learning with data augmentation, without pixel weights

(d) Deep Learning with data augmentation and with pixel weights

Figure 4: Test samples of red blood cell segmentation. In red, the red blood cells detected by the algorithm. In blue, the red blood cells undetected by the algorithm. The Deep Learning algorithm with data augmentation and pixel weights outperforms other variants and outperforms the Hough Transform (used for automatic labeling)

blood cells overlap. Using this data augmentation technique on the training set, we obtain accurate segmentation maps for blood samples where red blood cells are not round or overlap. This reduces noise in our training labels.

4.5 Pixel weights

Data augmentation reduces noise in the segmentation labels of the training set (see Section 4.4). Yet, these segmentation labels still contain many False Negative pixels. Interestingly, they contain very few False Positive pixels. In other words, labels are accurate for pixels labeled positively (red blood cell), while they are noisy for pixels labeled negatively (no red blood cell). We leverage this observation by adding pixel weights in the training loss. Pixels labeled positively (red blood cell) are given a high weight $w_+ = 10$, while pixels labeled negatively (no red blood cell) are given a low weight $w_- = 0.1$. As a result, the neural network learns to privilege pixels labeled positively which have accurate labels. The training loss becomes

$$\mathcal{L} = -\sum_{i=1}^{N}\sum_{h=1}^{H}\sum_{w=1}^{W} w_+ Y_{h,w}^{(i)} \log p(Y_{h,w} = 1 \mid X^{(i)}) + w_-(1 - Y_{h,w}^{(i)}) \log p(Y_{h,w} = 0 \mid X^{(i)}),$$

where i is the index of the non-fluorescent image (there are N images in total), where (h, w) is the position of the pixel, and where $p(Y_{h,w} = 1 \mid X^{(i)})$ is the probability that the network assigns to label l given the input non-fluorescent image $X^{(i)}$.

Besides, this impacts our choice of metrics. We report the Intersection over Union (IoU) of our model, since it is the standard metric used in the field of segmentation [4, 5]. However, given that labels are accurate for pixels labeled positively and noisy for pixels labeled negatively, we mostly care about reporting recall - the proportion of pixels labeled positively which were identified correctly.

4.6 Results and Error Analysis

We train three variants of our model. First, the model without data augmentation nor training weights (see Section 4.3). Second, we train the model with data augmentation but without training weights (see Section 4.4). Finally, we train the model with data augmentation and training weights (see Section 4.5).

The best performing model leverages both data augmentation and training weights. Qualitative results on the test set are shown in Figure 4. The model segments accurately red blood cells. It has very few false positives and very few false negatives. The model achieves an intersection over Union (IoU) score of 0.70 on the test set. This score is artificially low, since we are comparing accurate model predictions to noisy labels (obtained by Hough Transform). The recall of the model is more pertinent, since only the positive labels (obtained by Hough Transform) are accurate. The model reaches a recall score of 0.73 on the test set. Again, the recall is artificially low, since even positive labels can be noisy.

To the best of our knowledge, no benchmark was established by previous works for the segmentation of red blood cells, in terms of IoU nor recall. We suspect that our quantitative metrics widely underestimate the actual performance of our segmentation algorithm, since our automatic labels are noisy. We take comfort in the excellent qualitative performance of our algorithm, as shown in Figure 4. More importantly, we study the quality of our red blood cell segmentation by analyzing whether it helps improve the performance of a downstream task - classifying malaria parasites vs platelets (see Section 5).

5 Segmentation of Red Blood Cells helps improve the Classification of Malaria Parasites (Stage 2)

In this section, we use the segmentation of red blood cells obtained in Stage 1 (see Section 4) as a feature to distinguish malaria parasites from platelets. Indeed, malaria parasites tend to penetrate inside red blood cells, whereas platelets do not.

5.1 Problem Formulation

Our goal is to distinguish platelets from malaria parasites in the fluorescent image. The fluorescent image is cropped into multiple patches, each centered on a cell - which could be either a platelet or a malaria parasite. For each patch, we extract 6 visual features corresponding to the size, shape and color intensity of the patch. The 6 features are concatenated with the segmentation mask obtained in Stage 1 (see Section 4). This set of features, X, is provided as inputs to a classification model. We opt for a Linear Discriminant Analysis (LDA) [9], a simple algorithm which looks for a linear combination of features that separates malaria parasites from platelets. Note that 48% of the samples are malaria parasites, whereas 52% are platelets.

5.2 Results

Our classification model reaches a test AUC of 99.98% without the red blood cell segmentation. It reaches a test AUC of 99.998% with the red blood cell segmentation. Figure 5 compares the two performances. Our red blood cells segmentation therefore improves the detection of malaria parasites by an order of magnitude.

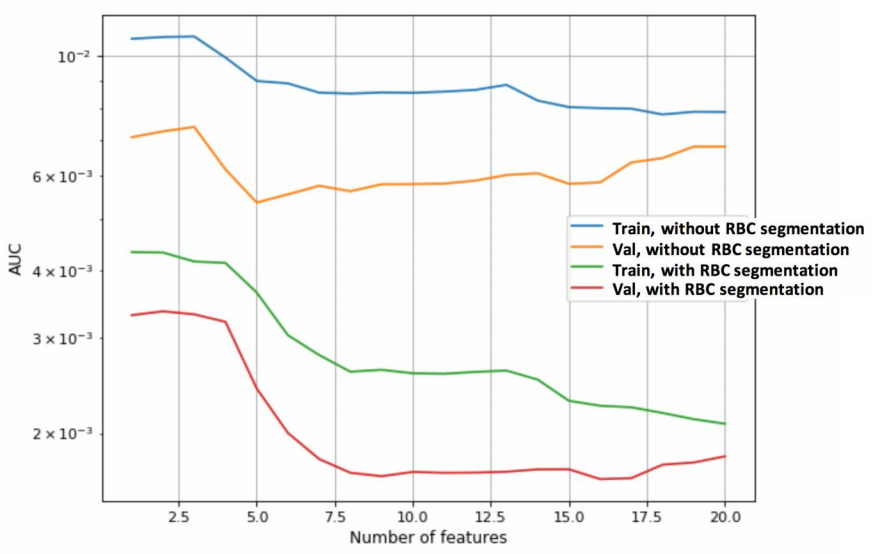


Figure 5: The train AUC and validation AUC of classification models trained respectively with and without the red blood cell segmentation features. The performance of the classification of malaria parasites improves by an order of magnitude if we incorporate the segmentation of red blood cells obtained in Stage 1.

6 Conclusion and Next Steps

In this project, we develop an algorithm which automatically detects malaria parasites in low-resolution images of blood samples. To the best of our knowledge, this is the first paper to attempt classifying malaria parasites using such low-resolution microscopes. We hope this work will help provide accurate diagnostic information for malaria in places with constrained access to high-resolution microscopes and to medical expertise.

Future work should explore the deployment of this system on a portable chip. Indeed, places with constrained access to medical infrastructure and to medical expertise tend to lack internet connection.

7 Contributions

A huge thank you to Lucas Fuentes, Hongquan Li and Cristian Bartolome Aramburu who helped me collect data, preprocess data, train the LDA and mentor the project!

References

[1] Mahdieh Poostchi, Kamolrat Silamut, Richard J Maude, Stefan Jaeger, and George Thoma. Image analysis and machine learning for detecting malaria. *Translational Research*, 194:36–55, 2018.

[2] Kevin Marsh, Dayo Forster, Catherine Waruiru, Isiah Mwangi, Maria Winstanley, Victoria Marsh, Charles Newton, Peter Winstanley, Peter Warn, Norbert Peshu, et al. Indicators of life-threatening malaria in african children. *New England journal of medicine*, 332(21):1399–1404, 1995.

[3] World Health Organization. *World malaria report 2015*. World Health Organization, 2016.

[4] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[5] Simon Jégou, Michal Drozdzal, David Vazquez, Adriana Romero, and Yoshua Bengio. The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 11–19, 2017.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[7] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[8] John Illingworth and Josef Kittler. A survey of the hough transform. *Computer vision, graphics, and image processing*, 44(1):87–116, 1988.

[9] Jieping Ye, Ravi Janardan, and Qi Li. Two-dimensional linear discriminant analysis. In *Advances in neural information processing systems*, pages 1569–1576, 2005.