
Automated Image Segmentation of Liver Stage Malaria Infection

Ava Soleimany
Divya Shanmugam
Harini Suresh
Jose Javier Gonzalez Ortiz

ASOLEI@MIT.EDU
DIVYAS@MIT.EDU
HSURESH@MIT.EDU
JJGO@MIT.EDU

Abstract

Global eradication of malaria depends on the development of drugs effective against the liver stage of the disease. A major bottleneck in this drug development pipeline is manual segmentation of microscopy images of liver stage infection. Though rule-based approaches for automated segmentation of these images have been developed, they have yet to gain headway, due to their lack of generalizability and reliance on hand-crafted features. To address this need, we developed a convolutional neural network based architecture for automated segmentation of microscopy images of liver stage malaria infection. Our pipeline produces high-precision segmentations nearly identical to the human gold standard and generalizes well on a diverse dataset. This tool should facilitate greater throughput in screening of anti-malarials with efficacy against liver stage malaria infection.

1. Biological Motivation

Malaria remains a major global health scourge, with nearly half of the world's population remaining at risk (Organization, 2016). The disease may be broken into two phases: the liver-stage infection, followed by the blood stage infection. Chronic, relapsing bouts of malaria result from dormant parasites formed during the liver stage of *Plasmodium vivax* infection (Figure 1). Despite this fact, there is only one anti-malarial with efficacy against dormant parasites in the liver stage of *P. vivax* infection, and therefore there is a pressing need for more drug development in this area (Wells et al., 2010; Alonso et al., 2011).

Proceedings of the 30th International Conference on Machine Learning, Atlanta, Georgia, USA, 2013. JMLR: W&CP volume 28. Copyright 2013 by the author(s).

2. Problem Statement

Drug development for liver stage anti-malarials involves complex, large-scale testing of many different drugs on *in vitro* cultures of liver cells that have been infected with malaria parasites, as seen in Figure 1 (March et al., 2013). To determine which drugs were effective, researchers manually segment microscopy images of infected liver cells to identify and size parasites that remain post-treatment. This presents a significant bottleneck in the screening pipeline. Still, manual segmentation of these immunofluorescent microscopy images remains the gold standard, but takes approximately 20 hours for an experienced research for a typical batch of images (over 3-4 experiments).

In this paper, we aim to provide a reliable, accurate method for automated segmentation of parasites in liver stage malaria infection. Beyond alleviating the manual burden of segmentation, an automated method would enable more rapid developmental cycles and higher-throughput screening for identifying drugs and drug targets for liver stage malaria. To the best of our knowledge, an automated annotation pipeline for liver stage malaria infection that performs comparably to human annotation has yet to be developed.

Object detection and segmentation in these immunofluorescent images presents a unique set of challenges. These images of malaria infected liver cells have significant variations in cell and parasite shape, density, and size, particularly in regards to the presence of both dormant parasites, which are very small, and actively growing parasites, which are large (Figure 1). The class distribution in this dataset is also highly imbalanced, due to the inherent fact that uninfected liver cells significantly outnumber infected liver cells.

In this paper, we develop a neural network based pipeline for automated image segmentation of liver stage malaria infection. We leverage a convolutional U-Net architecture to achieve this goal and demonstrate that our method outperforms our baseline for

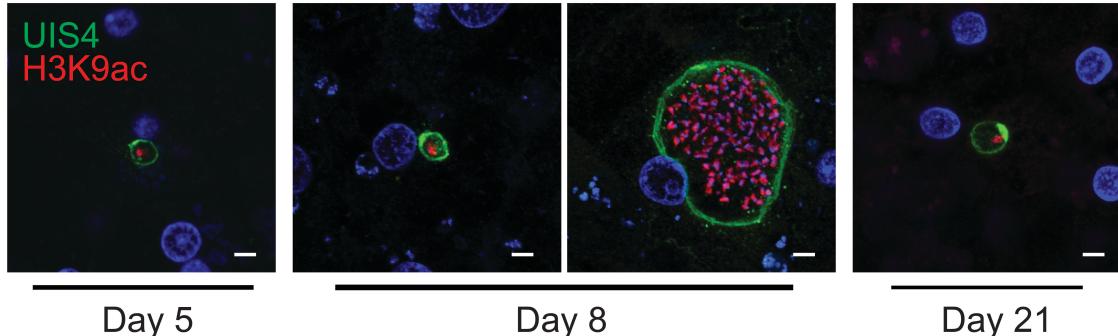


Figure 1. Confocal microscopy images of *P. vivax* infection in *in vitro* culture from 5, 8, and 21 days post infection, with the green and red stains indicating *P. vivax* parasites. On Day 8, both the small dormant forms (left) and the large mature forms (right) are present. The presence of small forms 21 days post infection suggests that the *in vitro* culture supports the full liver stage phase of *P. vivax*, including the long term maintenance of dormant forms. Scale bars: 5 μ m.

both detection and segmentation, achieving high precision and producing segmentations nearly identical to the human gold standard.

3. Related Work

Approaches to biomedical segmentation dichotomize into rule-based methods and network-based methods. We review recent work in both areas and expand upon key insights crucial to our work. The research regarding the application of these neural networks to liver stage malaria is sparse and current rule-based implementations fail to meet the accuracy standards necessary. Thus, a method for liver stage parasite identification and segmentation that meets human accuracy remains to be developed.

3.1. Rule-Based

We elaborate upon the two most popular rule-based approaches to segmentation in microscopy images: thresholding and the watershed algorithm.

Thresholding involves selecting a threshold intensity at which pixels are tagged as objects. This method is simple but suffers in the face of noise and non-uniform stain intensities in the images. Phansalkar et al. propose a local thresholding segmentation method to address the issue of variable stain intensity, but the method remains sensitive to noise (Phansalkar et al., 2011).

The watershed algorithm maps the boundary between objects in images to watersheds, or ridges that separate water reservoirs (Beucher & Meyer, 1992). The algorithm places a water source at each regional minima and floods each of these water sources. Places where disparate water sources meet are designated as

boundaries. Although widely effective, this algorithm is prone to over-segmentation. Moreover, in our application context, this algorithm would segment false positives (stains that do not correspond to a parasite).

CellProfiler, a popular open-source software, uses hand crafted features in conjunction with the aforementioned algorithms to perform object identification for cell image analysis tasks (Carpenter et al., 2006). However, due to the fact that it relies on pre-specified features, this approach is limited in its generalizability, accuracy, and robustness, as a new pipeline must be created for every distinct processing task. In the context of parasite identification, researchers indicate that there is up to a 20% discrepancy between manual and CellProfiler segmentations, and speak to CellProfilers tendency towards false negatives over false positives (Miller, 2017).

3.2. Network-Based

Convolutional neural networks have emerged as an attractive approach for the segmentation of biomedical images due to their strength on visual recognition tasks.

Ciresan et al. (Ciresan et al., 2012) used a convolutional network with a sliding-window setup to predict the class label of each pixel in an image, where a patch surrounding the pixel was provided as input. However, this approach suffers from computational inefficiency and exhibits a tradeoff between accuracy and the context provided by the input patches.

Fully convolutional networks, such as that proposed by Long et al. (Long et al., 2015), use a traditional contracting network architecture supplement with successive layers that use upsampling rather than pooling operations to increase the resolution of the layer out-

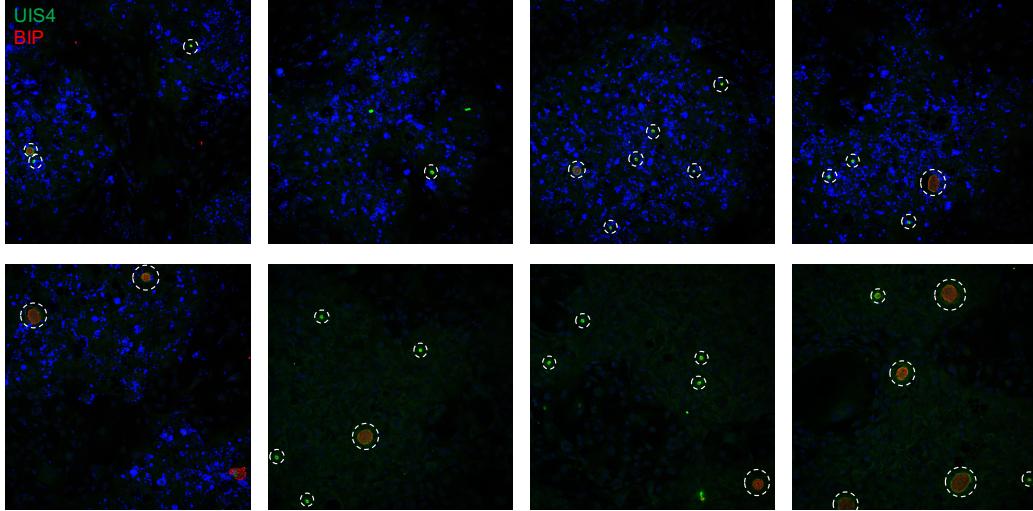


Figure 2. Representative images from our dataset of approximately 250 3-color images of *P. vivax* infected cultures, 8 days post infection, with the green and red stains corresponding to *P. vivax* parasites. Image set shows the presence of both small and large forms and the diversity in size, shape, and density of the images. White circles indicate true positives (parasites) as annotated by experienced researchers.

put. Combining features from the contracting path with the output of upsampling layers allows for localization.

The U-Net architecture developed by Ronneberger and colleagues (Ronneberger et al., 2015) expands upon this notion. U-Net uses a convolutional autoencoder structure, where the input image is compressed and then decompressed through successive convolutional layers to produce a final output. Skip-like connections transmit information from the encoder portion of the network to the decoder portion. Notably, the U-Net architecture can be trained with very few images, is quite fast, segmenting a 512×512 image in less than a second on a standard GPU, and outperforms the state-of-the art sliding-window network from Cirsean et al. (Ciresan et al., 2012; Ronneberger et al., 2015). The U-Net architecture has proven effective across many medical domains, from CT scans to skin cancer images (Dong et al., 2017; Chang, 2017).

4. Dataset

Our experimental dataset was generated by Sangeeta Bhatia’s lab and consists of approximately 750 1080×1080 pixel images of cultured human liver cells that have been cultured with *P. vivax* parasites. Approximately 0.2 – 1% of the liver cells are infected based on the typical infectivity of *P. vivax*.

Eight days post infection, the *P. vivax*-infected culture plate was preserved, and subsequently stained with a

stain for cellular nuclei (DAPI, blue), for a parasite membrane protein (UIS4, green), and a parasite cytoplasmic (cell internal) protein (H3K9ac or BIP, red). This staining scheme allows for visualization of parasites, as seen in Figures 1 and 2. 150 images across three colors, corresponding to these three distinct immunostains, are taken over the area of each well in a culture plate. The three images corresponding to a particular area in the plate are then stacked to generate a 3-color RGB image. We use a set of approximately 250 3-color images as our dataset.

Our image dataset is annotated with the pixel coordinates corresponding to segmentations of parasites present in the image. The dataset was annotated and curated by practiced researchers in the Bhatia lab. Representative images from our dataset are shown in Figure 2.

5. Experimental Setup

The segmentation task is to determine all pixels in an image that are part of a parasite. The workflow used may also be found in Figure 4.

5.1. U-Net Architecture

We use a U-Net architecture (Figure 5) for segmentation. U-Net rests on a convolutional autoencoder framework with the addition of skip-like connections to transmit information from the encoder portion of the network to the decoder portion.

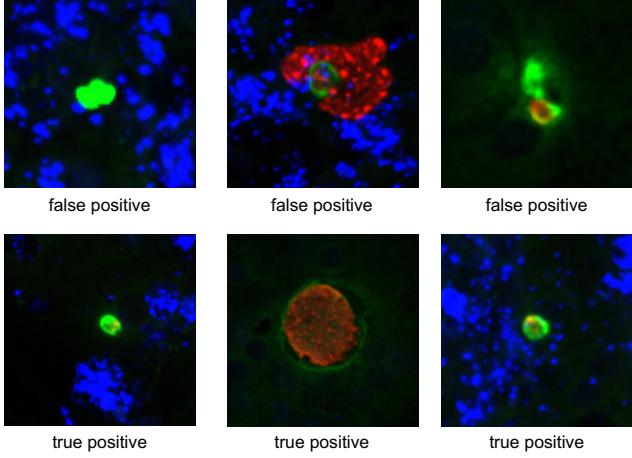


Figure 3. Examples of true and false positives.

Figure 5 visualizes the architecture in more detail. Note that all convolutional layers preserve the image dimensions (stride 1). In the upsampling phase this was done on purpose since strided deconvolutions (transposed convolutions) can produce checkerboard patterns in the output.

5.2. U-Net Objective Functions

We experimented with several loss functions to achieve stable training of the U-Net.

5.2.1. PIXELWISE CROSS-ENTROPY

The pixel-wise cross-entropy loss for an image is the sum of the cross-entropy on each pixel. The true label is 0 or 1 and the predicted label is a number in [0,1], since the output is the result of a sigmoid function on the last layer. Therefore, the pixel-wise cross-entropy for a single image takes the following form:

$$E(I) = - \sum_{x \in I} y \log \hat{y} + (1 - y) \log(1 - \hat{y}) \quad (1)$$

where I is the image (consisting of the set of all pixels), and \hat{y} is the output of the U-Net for point x .

5.2.2. PIXELWISE WEIGHTED CROSS-ENTROPY

A weighted form of cross-entropy was tested as an alternative to upsampling positive patches (section 5.3.1). In this case, the cross-entropy for each pixel is multiplied by a weight corresponding to the inverse frequency of that label across all images.

$$E(I) = - \sum_{x \in I} w_1 y \log \hat{y} + w_0 (1 - y) \log(1 - \hat{y}) \quad (2)$$

$$w_y = \frac{|I| * N}{\sum_I \sum_{x' \in I} \mathbb{1}(y' = y)} \quad (3)$$

where y is the true label corresponding to point x , \hat{y} is the output of the U-Net for point x , $|I|$ is the size of image I , and N is the total number of training images.

5.2.3. MEAN ABSOLUTE ERROR

The mean absolute error averages the absolute difference between the true and predicted values for each pixel. This loss function encourages sparsity and sharper segmentations.

$$E(I) = - \sum_{x \in I} \frac{|y - \hat{y}|}{|I|} \quad (4)$$

5.2.4. DICE LOSS

We also test a dice loss with a differentiable formulation:

$$E(I) = \frac{2 * \sum_{x \in I} y * \hat{y}}{\sum_{x \in I} y + \sum_{x \in I} \hat{y}} \quad (5)$$

5.2.5. TOTAL VARIATION LOSS

The total variation loss (TV) is the difference between an image and the image shifted by one pixel both horizontally and vertically. TV can be added as an additional term to any of the previous losses, and has a regularizing effect (Strong & Chan, 2003).

5.3. Data Pre-Processing and Augmentation

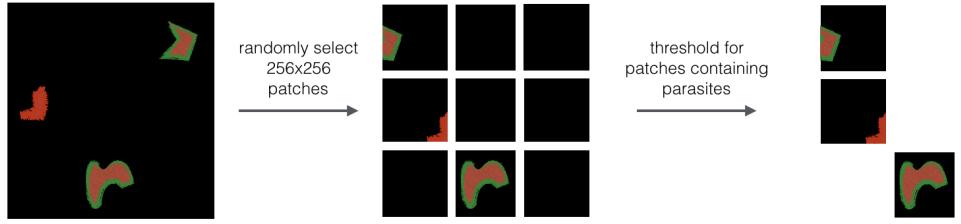
5.3.1. GENERATING PATCHES

Approximately 64%, 16%, and 20% of the 250 1080 × 1080 images was used for training, validation, and test cohorts, respectively. Since the memory needed to train with images of this size would be prohibitive, we initially trained on randomly selected patches of 256 × 256. This turned out to be much more manageable, with the patch size being large enough for generalization later with windowed application of the network.

5.3.2. BALANCING CLASSES

However, the initial performance of the U-Net architecture on the segmentation task was quite poor. We

Preprocessing: Raw Data —> Training Data



Training: Preprocessed Data —> Segmentation

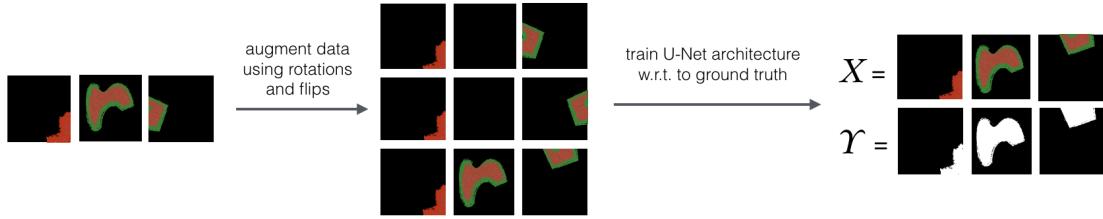


Figure 4. Data pre-processing and training pipeline. A threshold of 0.33 is used for the final step of pre-processing.

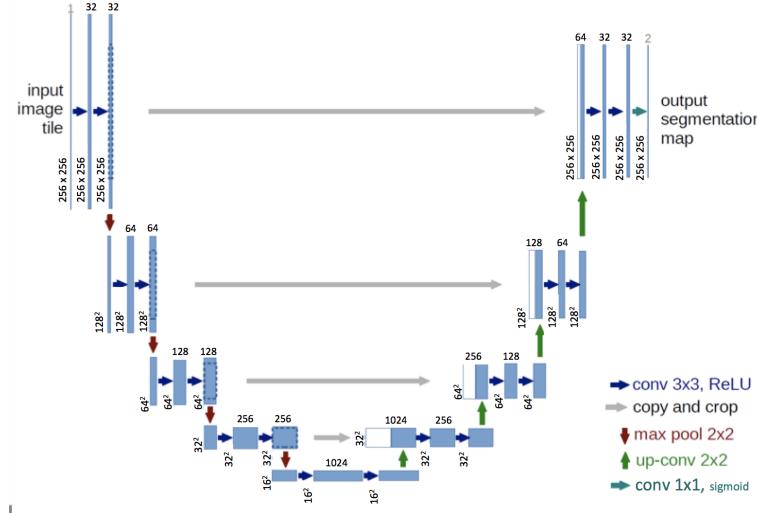


Figure 5. The U-Net architecture takes in an image of cultured liver cells infected with *P. vivax* and outputs the segmentation map of the given image. The key to the U-Net architecture is its convolutional autoencoder structure, which involves successive compression (encoding) and decompression (decoding) layers connected by skip-like connections.

reasoned that the issue could be either with the U-Net implementation or the dataset itself, and proceeded to test our architecture on a color ellipse dataset and a road segmentation dataset (ZFTurbo; CS-433). Here, the U-Net architecture performed very well, suggesting that the issue was with our dataset.

We hypothesized that the issue was due to the class imbalance in our dataset—a very small portion of a given image, if any, contains a malaria parasite, while in the

ellipse and road datasets the regions of interest are quite prominent. To address this issue, we filtered the images used to train the U-Net.

We checked for the presence of the stains for UIS4 (green, stains parasite membrane) and BIP (red, stains internal parasite protein). The presence of these dyes does not necessarily imply the presence of a malaria parasite (there are lots of false positives), but a positive example always has the presence of these dyes in

its vicinity.

The dyes are green and red fluorescent stains, so we detect whether there are pixels in those channels of the image with value above a threshold T determined via cross-validation. If so, we feed that patch into the model. Other methods for detection, such as a CNN, could also be used to solve this classification problem, with positive classified patches being then fed into the U-Net for segmentation. In addition to the patch generation, we further augmented the data by applying random rotations and symmetries to the patch images, since the data is invariant under these transformations.

5.4. Baselines

5.4.1. THRESHOLDING

As a baseline, we check for the presence of the stains for UIS4 (green, stains parasite membrane) and BIP (red, stains internal parasite protein) with the thresholding method described in Section 5.3.2. We determine $T = 0.33$ using cross-validation.

In order to produce a segmentation, we return a pixel-wise thresholding of the green and red channels for an entire image. This gives an image of equal size to the input image with binary labels for each pixel indicating whether it is part of the segmentation or not.

5.4.2. LOGISTIC REGRESSION

We also test a L2-regularized logistic regression trained on class-balanced patches (using the preprocessing methodology described in Section 5.3). Each patch was split into vectors of size 27, representing a 3x3 box of 3 channels with a given pixel at the center. The model then takes as input this size-27 vector and predicts a probability of the center pixel being positive for malaria. When predicting a segmentation for an entire image, we simply make a prediction for each pixel as described, and combine them.

The model was implemented using sklearn SGDClassifier with logistic loss to enable partial training with the patch generator.

5.5. Training

All units in our U-Net architecture (Figure 5) are ReLU, with the exception of the output layer which uses sigmoid activation as we are outputting a binary segmentation. All weights are initialized to be He Normal (He et al., 2015). For convolutional layers, all kernel sizes are 3×3 , and convolutions are always applied twice before and after every downsampling and

upsampling operation, which is specific to the U-Net architecture (Ronneberger et al., 2015). We found that batch normalization (Ioffe & Szegedy, 2015) improved convergence and used it in our final implementation. Our implementation uses an Adam optimizer (Kingma & Ba, 2014) with an initial learning rate of 10^{-3} and a minibatch size of 8, where in the case of a mini-batch containing positive examples 5 were random and 3 were known true segmentations. A learning rate schedule is used, where after 20 epochs the learning rate is decreased without an observed increase in the validation loss. We used a variety of loss metrics, described further in Section 5.2, for our experiments. We trained our model on approximately 200 RGB images. Our model ran for approximately 20 epochs before convergence. We performed one-fold cross validation using the validation data, due to limited computational resources and richness of the labels. All models were written in Keras and Tensorflow and trained in 1-2 hours on a NVIDIA Titan X GPU.

6. Results

6.1. Evaluation Metrics

6.1.1. AVERAGE PRECISION SCORE

Evaluation of our model is sensitive to the threshold chosen to distinguish segmented pixels. We employ the average precision score due to its ability to summarize threshold sensitivity.

The average precision score is calculated by taking the weighted sum of precisions at each threshold, where the weight is the increase in recall; concretely, this may be described as follows:

$$APS = \sum_n (R_n - R_{n-1})P_n$$

where R_n and P_n are the recall and precision at threshold n .

Precision is the percentage of segmented pixels that correspond to a parasite, and recall is the percentage of ground truth segmented pixels identified by our model. At a low threshold, we may identify all segmented pixels at a cost to the overall precision. Conversely, a high threshold will maintain high precision but demonstrate low recall. Ideally, as the threshold is lowered, the precision remains high and the recall increases.

6.1.2. DICE SIMILARITY COEFFICIENT (DSC)

The Dice Similarity Coefficient (DSC) is a statistic that computes the similarity between two samples, and

Model	Average Precision Score	Dice Similarity Coefficient
Thresholding Baseline	N/A	0.359
Logistic Regression	0.807	0.623
U-Net, cross entropy loss	0.984	0.882
U-Net, mean absolute error loss	0.895	0.934
U-Net, cross-entropy + total variation loss	0.956	0.897

Table 1. Accuracies achieved by various models on the test dataset. U-Net with cross entropy loss and U-Net with mean absolute error loss achieve the highest performance in regards to average precision score and dice coefficient, respectively.

takes on a value between 0 (no similarity) and 1 (identical). Intuitively, the DSC can be thought of as:

$$DSC = \frac{2TP}{2TP + FP + FN}$$

where TP, FP, and FN are the number of true positives, false positives, and false negatives, respectively. The DSC is equivalent to the F1 score (harmonic mean of precision and recall), as evidenced by its formulation. The DSC is widely used as a statistic for measuring precision of image segmentation (Zou et al., 2004; Taha & Hanbury, 2015). In this context, the DSC is a measure of spatial overlap, where 0 indicates no spatial overlap between two sets of binary segmentation results and 1 indicates complete overlap.

Here, we compute a pixel-wise DSC to measure the precision of the segmentations returned by each of the models we test:

$$DSC = \frac{2(|S_t \cap S_r|)}{|S_t| + |S_r|}$$

where S_t is the pixel set of the true segmentation and S_r is the pixel set of the returned segmentation.

6.2. Performance

We trained our U-Net architecture as described in Section 5.5 with each of cross entropy loss, mean absolute error loss, and (cross entropy+total variation) loss (Section 5.2), as the U-Net model did not train to convergence with the weighted cross entropy or Dice loss objective functions. We used the thresholding baseline, logistic regression, and U-Net models to perform the segmentation task on our test dataset of 50 RGB images.

We measure the precision of each of the working methods relative to the ground truth segmentation using both the APS and DSC (Section 6.1) evaluation metrics, as shown in Table 1.

Considering the average precision scores and DSCs for each of the models, we see that the U-Net architecture with any of the three objective functions significantly

outperforms the logistic regression and thresholding baseline methods in terms of both APS and DSC.

A difference between the DSC and APS evaluations is that the DSC operates over a set threshold (in practice, we determine this threshold with cross-validation), whereas the APS computes a weighted sum of the precision at all thresholds. Because we average over all thresholds, APS will tend to give a more optimistic score than DSC (consider how at very low thresholds, precision will be close to one, and we are including those scores in the average). Interestingly, the cross entropy loss proved optimal for the APS metric, while the mean absolute error loss was optimal for the DSC metric. The DSC quickly becomes unforgiving as the prediction around the boundary becomes worse. We hypothesize that the mean absolute error does better on the DSC because it penalizes outliers more heavily. The total variation term does not appear to improve the result. We interpret this result as an indicator that the network without this regularization term still generalizes well.

In the end, the ultimate goal here is not necessarily to produce a segmentation that is as close to the human segmentation as possible on a pixel-wise basis. Rather, what is more valuable to researchers is an automated segmentation that robustly, quickly, and reliably identifies all the parasites in a given image without picking up on false positives. To see how well our model performs in this sense, we visually compared the segmentations returned by each of U-Net with cross entropy loss, thresholding baseline, and logistic regression to images from the dataset with the true positives highlighted (Figure 6). The baselines appear unable to distinguish malaria cells from other stains of similar brightness, or recognize that two cells of quite different brightnesses could both be malaria cells. The logistic regression is somewhat able to catch a few more of the latter case, but the U-Net model uniformly outperforms it. These results suggest that our U-Net architecture is both sensitive and specific in its detection and segmentation of parasites in these images.

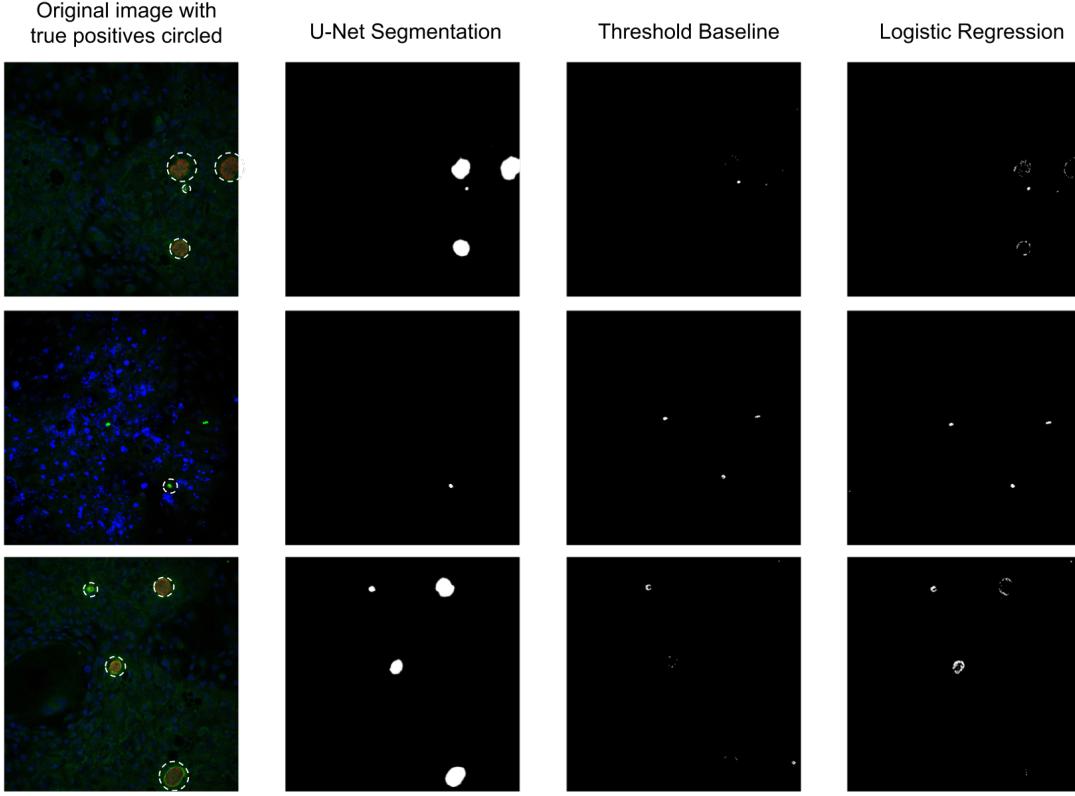


Figure 6. Results for the segmentation task, comparing the segmentations returned by the U-Net with cross entropy loss, threshold baseline, and logistic regression models. We see examples of where the baselines are not able to detect infected cells in the first and third rows, and an example of where the baselines falsely detect infected cells in the second row.

7. Future Work

Our work supports the potential for neural network-based methods to automate previously time-intensive processes of the anti-malarial drug development pipeline. In particular, the U-Net demonstrates its strength in segmentation of parasites during the liver stage of malaria with near-human accuracy. In future work, we hope to further develop our model along three axes: efficiency, generalization, and usability.

7.1. Efficiency

Training time for the model takes on the order of 1-2 hours, and inference on a given image takes 3-5 seconds, a drastic improvement to the manual tagging approach. Because of its efficiency, our method has the potential to enable the rapid research necessary to iterate towards liver stage anti-malarials.

7.2. Generalization

Currently, we employ thresholding of the randomly sampled inputs to address class imbalance in the training data. This choice is specific to our problem space

and may not generalize well to cases in which class imbalance is more complex. Instead of a rule-based approach to mitigating a skewed train distribution, we believe a robust detection network could accomplish the same task with a higher level of generalization. Furthermore, this separation of tasks results in a constrained input domain for the second network, leading to a much more learnable task.

7.3. Usability

Competitive precision scores and Dice coefficients alone are not enough to alter the drug development cycle. The resulting software must also be accessible to researchers. Towards that end, we plan to produce a well-documented executable to deploy to researchers. We have also identified two areas of further automation: parasite counting and sizing. These extensions will enable a rigorously quantitative characterization of liver-stage malaria.

In summary, our method addresses a significant bottleneck in the drug development pipeline for malaria, a parasite which affects over 1 million people every year. We use real data and achieve state-of-the-art results on

a critical step towards global eradication.

8. Division of Labor

Jose constructed the U-Net and experimented with the aforementioned loss functions. Harini implemented the baselines and performed experiments comparing them to our final implementation. Ava was our primary point of communication with the Bhatia lab and produced the data pre-processing pipeline. Divya implemented the evaluation portion of the pipeline. Each team member took responsibility for one milestone and collaboratively wrote the final report.

References

- Alonso, Pedro L, Bassat, Q, Binka, F, Brewer, T, Chandra, R, Culpepper, J, Dinglasan, R, Duncan, K, Duparc, S, Fukuda, M, et al. A research agenda for malaria eradication: drugs. *PLoS medicine*, 8(1):e1000402, 2011.
- Beucher, Serge and Meyer, Fernand. The morphological approach to segmentation: the watershed transformation. *Optical Engineering-New York-Marcel Dekker Incorporated-*, 34:433–433, 1992.
- Carpenter, Anne E, Jones, Thouis R, Lamprecht, Michael R, Clarke, Colin, Kang, In Han, Friman, Ola, Guertin, David A, Chang, Joo Han, Lindquist, Robert A, Moffat, Jason, et al. Cellprofiler: image analysis software for identifying and quantifying cell phenotypes. *Genome biology*, 7(10):R100, 2006.
- Chang, Hao. Skin cancer reorganization and classification with deep neural network. *arXiv preprint arXiv:1703.00534*, 2017.
- Ciresan, Dan, Giusti, Alessandro, Gambardella, Luca M, and Schmidhuber, Jürgen. Deep neural networks segment neuronal membranes in electron microscopy images. In *Advances in neural information processing systems*, pp. 2843–2851, 2012.
- CS-433, EPFL. Epfl ml road segmentation. <https://www.kaggle.com/c/epfml-segmentation/data>.
- Dong, Hao, Supratak, Akara, Mai, Luo, Liu, Fangde, Oehmichen, Axel, Yu, Simiao, and Guo, Yike. Tensorlayer: A versatile library for efficient deep learning development. In *Proceedings of the 2017 ACM on Multimedia Conference*, pp. 1201–1204. ACM, 2017.
- He, Kaiming, Zhang, Xiangyu, Ren, Shaoqing, and Sun, Jian. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034, 2015.
- Ioffe, Sergey and Szegedy, Christian. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, 2015.
- Kingma, Diederik and Ba, Jimmy. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- Long, Jonathan, Shelhamer, Evan, and Darrell, Trevor. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3431–3440, 2015.
- March, Sandra, Ng, Shengyong, Velmurugan, Soundarapandian, Galstian, Ani, Shan, Jing, Logan, David J, Carpenter, Anne E, Thomas, David, Sim, B Kim Lee, Mota, Maria M, et al. A microscale human liver platform that supports the hepatic stages of plasmodium falciparum and vivax. *Cell host & microbe*, 14(1):104–115, 2013.
- Miller, Alex, 2017.
- Organization, World Health. *World malaria report 2015*. World Health Organization, 2016.
- Phansalkar, Neerad, More, Sumit, Sabale, Ashish, and Joshi, Madhuri. Adaptive local thresholding for detection of nuclei in diversity stained cytology images. In *Communications and Signal Processing (ICCSP), 2011 International Conference on*, pp. 218–220. IEEE, 2011.
- Ronneberger, Olaf, Fischer, Philipp, and Brox, Thomas. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 234–241. Springer, 2015.
- Strong, David and Chan, Tony. Edge-preserving and scale-dependent properties of total variation regularization. *Inverse problems*, 19(6):S165, 2003.
- Taha, Abdel Aziz and Hanbury, Allan. Metrics for evaluating 3d medical image segmentation: analysis, selection, and tool. *BMC medical imaging*, 15(1):29, 2015.
- Wells, Timothy NC, Burrows, Jeremy N, and Baird, J Kevin. Targeting the hypnozoite reservoir of plasmodium vivax: the hidden obstacle to malaria elimination. *Trends in parasitology*, 26(3):145–151, 2010.

ZFTurbo. https://github.com/ZFTurbo/ZF_UNET_224_Pretrained_Model.

Zou, Kelly H, Warfield, Simon K, Bharatha, Aditya, Tempczyk, Clare MC, Kaus, Michael R, Haker, Steven J, Wells, William M, Jolesz, Ferenc A, and Kikinis, Ron. Statistical validation of image segmentation quality based on a spatial overlap index 1: Scientific reports. *Academic radiology*, 11(2):178–189, 2004.