



INSTITUT
POLYTECHNIQUE
DE PARIS



Design Project

Fourier Ptychography Microscopy for malaria detection

Realized by :

Slimane Baamara and Weiqiang Liu

Under the supervision of:

Prof. Patrick Horain Telecom SudParis

October 1, 2021

Contents

1	Introduction	2
2	Method	4
2.1	Fourier ptychography microscopy	4
2.2	Image reconstruction	6
2.2.1	Image reconstruction using Fourier Ptychography	6
2.2.2	Pupil reconstruction	8
2.2.3	LED position optimization (self-calibration)	9
2.3	Image segmentation	9
2.4	Image Classification	10
3	Integrated interactive interface	11
3.1	Image preprocessing plugin	11
3.2	Image segmentation	13
3.3	Image segmentation plugin	13
3.3.1	Image segmentation results	14
3.4	Cells extraction	15
3.5	Image reconstruction using (FPM)	16
3.6	Image classification	18
3.6.1	Image classification plugin	18
3.6.2	Image classification results	19
3.7	Image saving and loading plugin	20
3.8	Contour assist plugin	20
4	Conclusion	22

Abstract

Fourier ptychography microscopy (FPM) is a novel imaging method. It obtains a series of low-resolution photos by adding a movable light source to an ordinary microscope, and then calculates and reconstructs the high-resolution images. Malaria parasites are a common red blood cell infection. Since red blood cells do not have a nucleus, the presence of parasites can be easily observed under a microscope. The use of FPM to observe parasites is simple and convenient, and suitable for use in resource-poor areas.

In this report, we first explain the imaging principle of FPM and introduce the steps of image reconstruction, LED position optimization (Self Calibration), Image segmentation, Image Classification.

Then, finally we introduce the Graphical User Interface. Our first mission is to understand the FPM technique and the work already done by the master's students (first part) and reproduce the results found, and propose additional improvements that could improve the results.

Our second mission is to develop an interactive graphical user interface that implements previous work (image reconstruction, segmentation, classification), visualizes the results, and allows individual control of blood cells at different images.

1 Introduction

Malaria is a serious and sometimes fatal disease caused by a parasite that commonly infects a certain type of mosquito which feeds on humans. People who get malaria are typically very sick with high fevers, shaking chills, and flu-like illness. Malaria remains the most important parasitic disease worldwide. Millions of travellers from non-endemic areas visit malaria-endemic regions each year, which is responsible for the death of 445,000 people worldwide in 2017 according to WHO [2].

Malaria is caused by single-celled microorganisms of the Plasmodium group, which contaminates red blood cells and destroys them. Malaria is usually confirmed by the microscopic examination of blood films or by antigen-based rapid diagnostic tests (RDT). The disease can also be detected by polymerase chain reaction (PCR) [3]. However, this method is very expensive and the main affected countries do not have the means to generalize this kind of method. Despite the widespread usage of microscopes, diagnosis by microscopy suffers from two main drawbacks: many rural areas are not equipped to perform the test, and the accuracy of the results depends on both the skill of the person examining the blood film and the levels of the parasite in the blood. The sensitivity of blood films ranges from 75 to 90% in optimum conditions, to as low as 50%.

There are several key factors to consider in the inspection of parasites: 1. Identification of the different malaria parasites. Four kinds of malaria parasites infect humans: Plasmodium falciparum, P. vivax, P. ovale, and P. malariae. Different parasites have different infection ability, transmission ability, and different symptoms, which also correspond to different treatment options.

Therefore, distinguishing the types of parasites is very important for clinical diagnosis. 2. Numbers of parasites. A parasite count is a useful indicator of disease severity. There is also a large gap in the number of different parasites that cause disease. 3. The size of the parasite. Different types of parasites also have differences in size, which will affect some detection methods that rely on naked eyes, thereby affecting false negatives.

There are many ways to diagnose parasites, which can be divided into optical detection, chemical detection and antibody detection. Here we briefly introduce a few:

1. Conventional light microscopy with Giemsa stained blood films is the most common method, which uses a nucleic acid stain to locate the genetic material of parasites, which has been used for more than a century;
2. Fluorescent microscopy. Mature erythrocytes do not normally contain DNA and RNA, while malaria parasites do. This has led to the use of fluorescent dyes, mainly acridine orange (AO), to detect parasites.
3. Fluorescent microscopy after centrifugation. Centrifugation can separate genetic material and achieve a sensitivity greater than 90% when compared to microscopy. However, its shortcomings are also obvious: high cost, difficulty in species identification and quantification;
4. Antigen detection assays in dipstick format. The identification of histidine-rich protein 2 (HRP-2) in P. falciparum infections has led to a simple, rapid dipstick assay, however, they only detect infections

with *P. falciparum*;

5. Molecular methods: Polymerase chain reaction (PCR) . PCR assays can detect fewer parasites. Another advantage of PCR is the significantly improved species identification in mixed infections when compared with microscopy. The time needed to perform most PCR assays prohibits their use in routine diagnostic situations. From the perspective of detection difficulty, microscope observation is simple and convenient, and can provide the number of parasites, but the accuracy is not as high as that of chemical and biological detection. Therefore, improving the accuracy of the microscope and simplifying the process of counting parasites through image processing can greatly increase the utilization rate of the microscope.

Computer-aided diagnosis has been significantly contributing for quantitative characterization of diagnostic markers in medical imaging field and subsequently augmented decision making towards disease diagnosis and prognosis. In recent times, machine learning techniques play important role in strengthening diagnostic precision using medical imaging informatics protocols. For automatic detection of malaria pathogens from the microscopic images, Convolutional Neural Network (CNN) [45] received much attention from the researchers in recent times.

Patrick Horain has compared Deep Convolutional Neural Networks (DCNN) frameworks, namely AlexNet and VGGNet, for the classification of healthy and malaria-infected cells in large, gray-scale, low quality and low resolution microscopic images, in the case only a small training set is available. Experimental results deliver promising results on the path to quick, automatic and precise classification in untrained images. However, CNN requires powerful CPU and GPU, and it takes a very long time to get the detection results, which is usually impossible to achieve in remote areas.

Tanzilur Rahman proposed a deep learning based automatic malaria parasite detection from blood smear and proposed its smartphone based application. Their deep learning-based model can detect malarial parasites from microscopic images with an accuracy of 99.23% while requiring just over 4600 floating point operations, which is suitable for mobile phones and a server-backed web application. In the near future, we can look forward to a more efficient and fast mobile inspection method.

The detection method described in this report is based on the deep learning algorithm. Unlike other models, we innovate in sampling. Ordinary machine learning is based on images obtained directly from a microscope. They usually use image enhancement methods to improve the recognition of parasites, but the images used in this article come from a technique called Fourier ptychography microscopy, which is better than ordinary microscope images. The clarity is higher and the field of view is larger. Fourier ptychography microscopy (FPM) is a novel method of microscopic image processing [1], which iteratively accumulates high frequency information from a number of variably illuminated, low-resolution intensity images in Fourier space to produce a wide-field, high-resolution complex sample image. FPM enjoys simple apertures and high resolution and large field of view, thus it is promising for malaria parasite diagnosis in resource-poor areas.

In this report, we will introduce parasite detection based on Fourier ptychography microscopy. The following text is divided into five parts, the first part introduces the imaging principle of FPM, and the second part we will introduce how to reconstruct the high-quality image based on the image data obtained by FPM.

In the third part, we will talk about image segmentation, how to segment a complete microscope observation image into images containing only one cell according to the size of red blood cells. In the fourth part, we will discuss classification. Through machine learning, the obtained red blood cell images are automatically classified into infected and uninfected. Then we will discuss the integrated interactive application we designed. We will integrate the three image processing processes mentioned above into one app.

In this interactive program, users can freely read and store images and perform simple image processing, for example, select channel, rotate, crop. Users can also read FPM images and perform image segmentation on them. In particular, thanks to the interactive page, users can customize the range of images that need to be segmented, thereby reducing the time for image processing. The segmented image will be divided into two labels of health and disease by the classification program.

2 Method

The second part mainly introduces the detection process of reconstructed red blood cell image. This process can be divided into sampling, image reconstruction, segmentation, and classification.

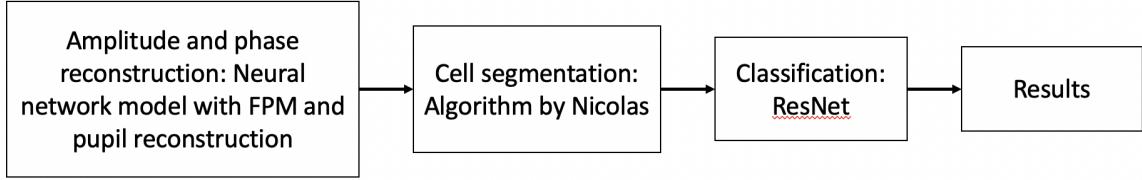


Figure 1: Workflow of malaria detection using Fourier ptychography microscopy.

2.1 Fourier ptychography microscopy

The throughput of the imaging platform is basically limited by its space-bandwidth product (SBP) in optical system, which is defined as the number of degrees of freedom it can extract from an optical signal. Increasing the SBP of the microscope fundamentally confuses the scale-dependent geometric aberrations of its optical components, resulting in a compromise between achievable image resolution and FOV. However, under the microscope, a large SBP is very ideal for biomedical applications, such as digital pathology, hematology, immunohistochemistry, and neuroanatomy. The strong demand of biomedicine and neuroscience for digital imaging of a large number of tissue slices for analysis has prompted the commercial development of complex mechanical scanning microscope systems and lensless microscope devices.

Artificially increasing the SBP of the imaging system by mechanical means is sub-optimal because it requires precise control of drive, optical alignment, and motion tracking. In addition, the mechanical solution only accepts the inherent resolution limit and SBP of traditional microscope optics, while ignoring the computational addressability problem of resolution enhancement. Lensless microscopy methods, such as digital online holography and contact imaging microscopy, provide unique imaging capabilities, but they also have certain shortcomings. For example, digital coaxial holography cannot be used well for continuous samples, and contact imaging microscopes require the sample to be very close to the sensor. Fourier ptychography is a computational imaging technique based on optical microscopy that consists in the synthesis of a wider numerical aperture from a set of full-field images, which provides a way to resolve the contradiction between high resolution and large field of view. FPM only needs a small numerical aperture to obtain high-resolution images. In the reconstruction process, we can obtain both the intensity and phase images of the sample through numerical calculations, such as deep learning.

The setup of FPM imaging system and operation of the Fourier ptychographic microscope prototype is shown in Fig 2. At the bottom is an LED light source, which can emit a single light source according to the code. The light source at different positions emits light, passes through the sample above, and then passes through the prism to reach the detection in image plane.

The process of FPM is straightforward:

- (1) An LED array sequentially illuminates the sample with angle-varied plane waves. The constraint from the objective's coherent transfer function is digitally panned across the Fourier space to reflect the angular variation of its illumination.
- (2) A two-dimensional sample is placed at the focal plane of a low numerical aperture microscope objective with an LED light source, thus a low resolution image is obtained.
- (3) LED moves back and forth and a sequence of low resolution images are collected, each of them are under one different LED illumination with different angles.
- (4) Based on a set of collected low-resolution intensity images, we computationally reconstruct a high-resolution image of the sample following the recovery procedure in Fourier domain, as shown in Fig.7.

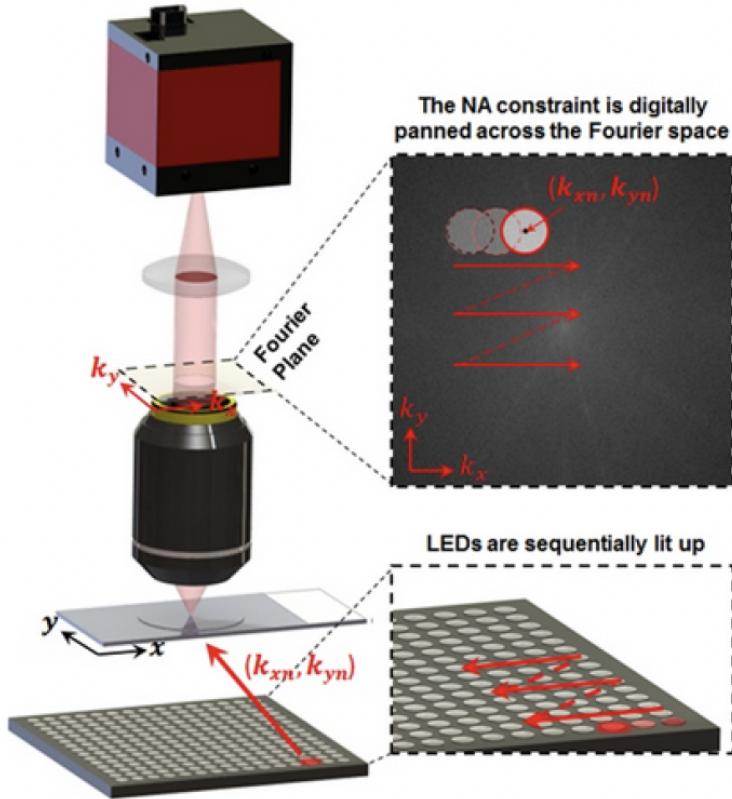


Figure 2: Iterative recovery procedure of FPM.

We first note that our recovery process alternates between the spatial and Fourier domains. Second, we assume that illuminating a thin sample by an oblique plane wave with a wavevector is equivalent to shifting the centre of the sample's spectrum in the Fourier domain. Third, we assume that the filtering function of our objective lens (that is, coherent optical transfer function) in Fourier space is a circular pupil. FPM generates a high-resolution image from a set of N low-resolution measurements.

The FPM method begins by making an initial guess of the high-resolution object function in the spatial domain. A good starting point is to select phase 0 and intensity of image 0 as any upsampled low-resolution image (an initial guess with constant value also works). The Fourier transform of the initial guess creates a broad spectrum in the Fourier domain. Then a small subregion of this spectrum was selected, equivalent to a low-pass filter, and we apply Fourier transformation to generate a new low-resolution target image. The applied lowpass filter shape is a circular pupil, given by the coherent transfer function of the objective lens. The position of the low-pass filter is selected to correspond to a particular angle of illumination. For example, the subregion enclosed by the red circle in Fig. 7 corresponds to an image collected under normally incident illumination. Third we replace the target image's amplitude component obtained under one illumination angle with an updated, low-resolution target image. We then apply Fourier transformation to this updated target and replace its corresponding subregion of the high-resolution Fourier space. We repeat to select a small, circular region in space and update it with measured image data for other plane wave illuminations. Examples are represented by the green and blue circles in Fig. 7. Each shifted subregion corresponds to a unique, low-resolution intensity measurement, and each subregion must overlap with neighbouring subregions to assure convergence. This iterative update continues for all N images, at which point the entire high-resolution image in Fourier space has been modified with data from all low-resolution intensity measurements. At the end of this iterative recovery process, the converged solution in Fourier space is transformed to the spatial domain to recover a high-resolution image with a dramatically increased SBP (high-resolution and wide-FOV).

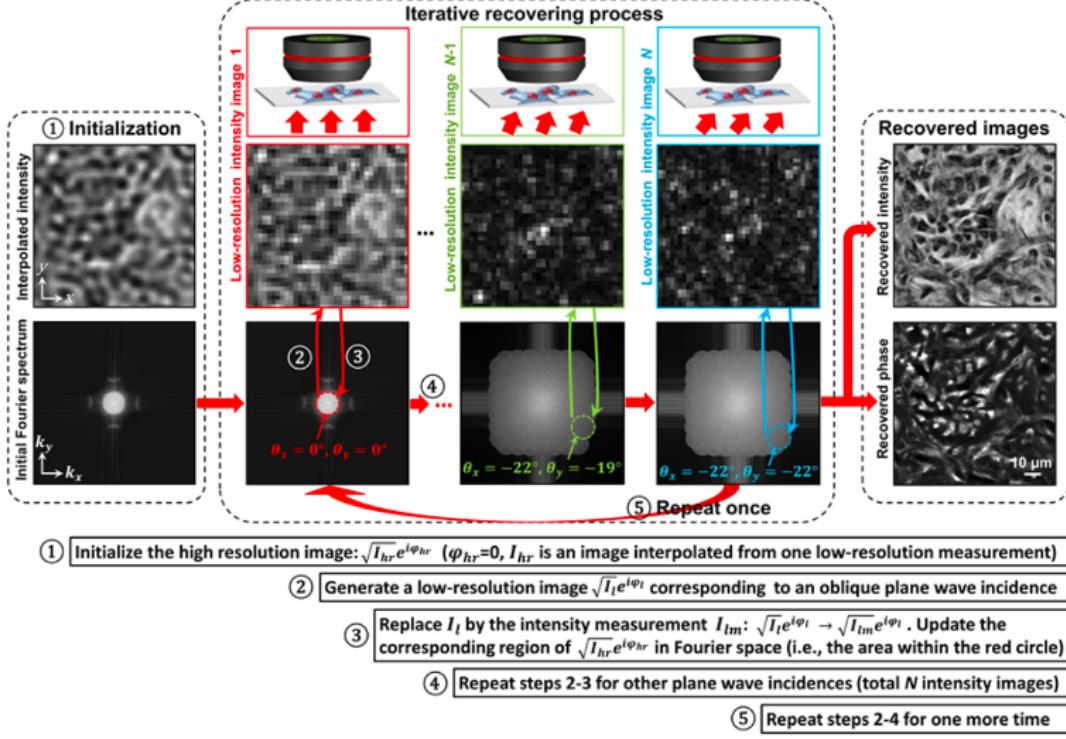


Figure 3: Iterative recovery procedure of FPM.

2.2 Image reconstruction

2.2.1 Image reconstruction using Fourier Ptychography

The key information about an image is intensity and phase. From each imaging in FPM, we can get the intensity information of the photo, that is, the intensity distribution of the light, but we don't know the phase of the picture. We use Fourier transform and inverse Fourier transform to reconstruct the image.

The Fourier Transform is an important image processing tool which is used to decompose an image into its sine and cosine components. The output of the transformation represents the image in the Fourier or frequency domain, while the input image is the spatial domain equivalent.

In the Fourier domain image, each point represents a particular frequency contained in the spatial domain image.

First, we select an initial guess, that is, assume that any picture is a high-definition picture, and extract its intensity distribution. Then we perform Fourier transform on the intensity to get the complex Fourier transform, and the phase is still unknown at this time.

The obtained Fourier transform needs to meet low pass constraint (pupil) in the Fourier plane, thus the TF is extended from the multiple captures.

By inverse Fourier transform, the object function is obtained. The object functions should also meet some restrictions, such as the size of the aperture, focal length, etc. These restrictions will further adjust the object function. Through iteration of this process, we finally get the image parameters as close to the original object as possible.

The aberration correction process discussed in the last section is based on the known aberrated pupil function. One question is that, if the pupil aberration is unknown, can we still compensate for the aberrations? We have two approaches for correcting unknown pupil aberrations in FP.

The first approach is to use the concepts in adaptive optics for aberration correction [4] and the second approach is to recover both the high-resolution complex object and the unknown aberrated pupil function in the iterative process [5]. The details of these aberration is not described in this report.

After aberration, in this project we use neural network model for image reconstruction with FPM,

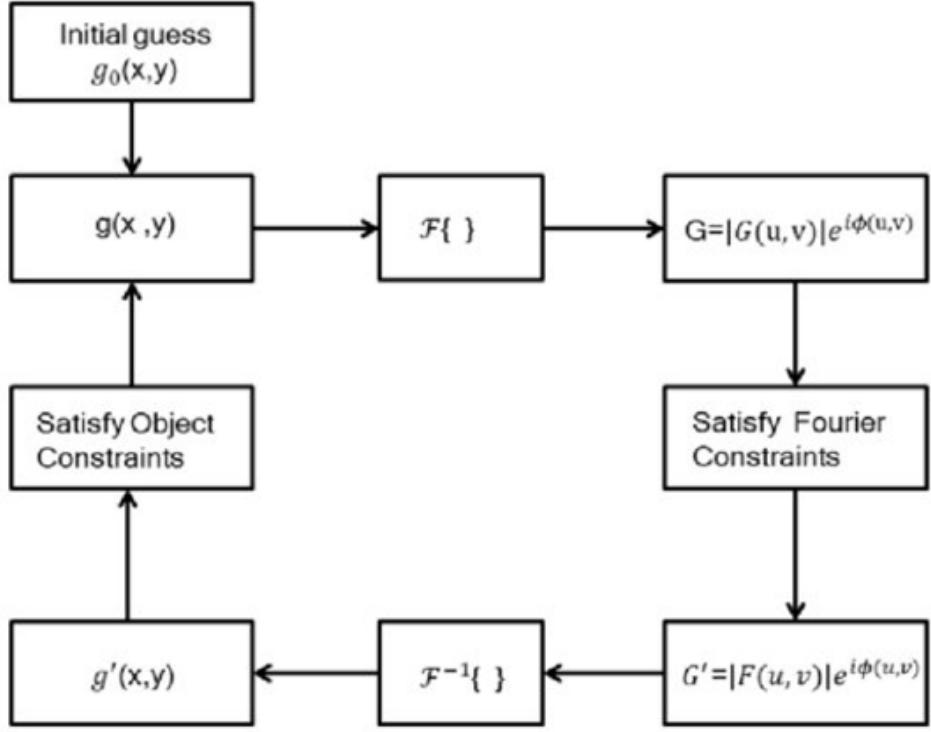


Figure 4: Flowchart of the interactive Gerchberg-Saxton algorithm

using Coherent Transfer Function (CTF) of the microscope pupil and low resolution images as input, based on [6]. Since the reconstruction process involves linear operation, it can be implemented as successive convolutions in a Convolved neural network. As shown in Fig. 9, the neural network is composed of several layers.

Here we describe the pupil, the center circle in the Fourier space, as a linear combination of Zernike polynomials. In the entries of the neural network we provide the first 10 polynomials of Zernike type and the CTF as the input. In this network, the layers of the complex object O_r and O_i will be trained by gradient back-propagation. And also the weights of the linear combination layer of the Zernike polynomials will be driven by the back-propagation of the gradient.

Therefore, the layers of the complex object and the linear combination of the Zernike polynomial are initialized, the complex object is initialized as described in the previous model. And the weights of the layers of the Zernike linear combination polynomial are all initialized with zeros.

Then the pupil is separated into two layers: one corresponding to its real part and the other to its imaginary part. We perform the multiplication between the complex object and the complex pupil. In the next layer, the result of this multiplication is sliced according to the size of the low-resolution image, then the reverse transformation is performed to obtain the low-resolution image generated by the grating. An intensity correction is made in the last layer.

By employing a lens to switch the spatial and Fourier domain, FPM has several unique advantages:

- (1) FPM reduces the requirement of spatial coherence as the images are captured in spatial domain [8]. In FPM, we only need to maintain the spatial coherence over the scale of the point spread function at the object plane. As such, we can use partially coherent LED illumination in FPM settings.
- (2) By scanning the aperture in the Fourier domain, FPM is able to recover an image with a resolution beyond the frequency limit of the employed lens. Aperture scanning in the Fourier domain can be implemented by using a simple LED array for angle-varied illumination, and thus, no mechanical scanning is needed in an optical FPM platform.
- (3) The Fourier spectrum of an image typically has a very high dynamic range. The signal strength at

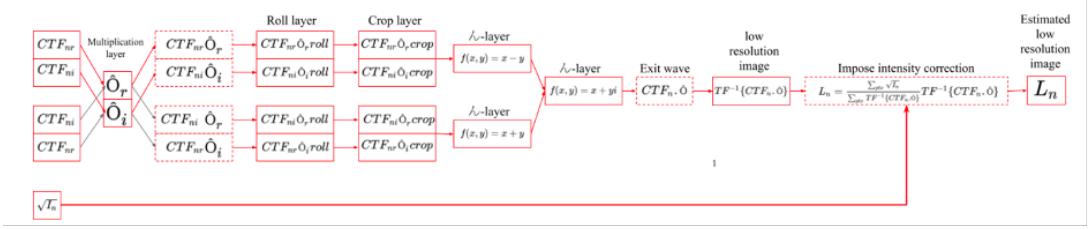


Figure 5: Neural network model for image reconstruction with FPM, using CTF and low resolution images as input, based on [7]

the center of the Fourier spectrum is typically orders of magnitude higher than that at the edge. As a result, real-space ptychography requires a detector with a very high dynamic range for capturing diffraction patterns at the Fourier space. FPM, on the other hand, directly captures images in the spatial domain.

2.2.2 Pupil reconstruction

The pupil function or aperture function describes how a light wave is affected upon transmission through an optical imaging system such as a camera, microscope, or the human eye. More specifically, it is a complex function of the position in the pupil or aperture (often an iris) that indicates the relative change in amplitude and phase of the light wave.

As we mentioned in the previous model, as the pupil reconstruction was not performed, the tests on the data set did not work well, which demonstrates the importance of pupil reconstruction for quality FPM image reconstruction. This phenomenon is due to the fact that we have optical aberrations in the imaging system, and these aberrations vary depending on the complexity of the lens used in the microscope, so correcting these aberrations allows the reconstruction of images by FPM with aberrant lenses (15).

Previously we assume the system is perfect and described the binary pupil as:

$$\begin{cases} P(u, v) = 1, & \forall u, v : \sqrt{u^2 + v^2} \leq R \\ P(u, v) = 0, & \forall u, v : \sqrt{u^2 + v^2} > R \end{cases}$$

And now we start to describe the pupil as a linear combination of Zernike polynomials as can be seen in equation:

$$\begin{cases} P(u, v) = \exp(iZ_{comb}), & \forall u, v : \sqrt{u^2 + v^2} \leq R \\ P(u, v) = 0, & \text{otherwise} \end{cases}$$

P represents the complex pupil and the term Z_{comb} is a linear combination of Zernike polynomials. We can describe the workflow of this neural network as follows: in the inputs we provide the first 10 polynomials of Zernike type. We also provide the CTF as an input but with a difference compared to the previous model, because this CTF is centered at the point (0, 0), that is to say that it is not moved according to the positioning of the LED. We also have as inputs the wave vectors k_x and k_y for the n different intensity measurements and the low resolution acquisition images. In this network, the layers of the complex object \hat{O}_r and \hat{O}_i will be trained by gradient back propagation.

And also the weights of the linear combination layer of the Zernike polynomials will be driven by the backpropagation of the gradient. Therefore, the layers of the complex object and the linear combination of the Zernike polynomial are initialized, the complex object is initialized as described in the previous model. And the weights of the layers of the Zernike linear combination polynomial are all initialized with zeros. After initialization, the result of the linear combination is used to define the pupil, then the pupil is separated into two layers: one corresponding to its real part and the other to its imaginary part. On the side of the complex object after initialization, we move that object relative to the wave vector, for each measurement of different intensity. We then perform the multiplication between the complex object and the complex pupil.

In the next layer, the result of this multiplication is sliced according to the size of the low-resolution image, then the reverse transformation is performed to obtain the low-resolution image generated by the

grating. And as in the previous model, an intensity correction is made in the last layer. Neural network model for FPM image reconstruction and pupil reconstruction.

2.2.3 LED position optimization (self-calibration)

In our database, we do not have the exact position of the LEDs, but only information on their format. The positioning of the LEDs is important information, as mentioned in [9]. The imprecision of the positioning of the LEDs, mainly in the matrices made on the measurements, can lead to a degradation of the reconstruction.

The positions of LEDs have been initialized as:

- The center LED to the position (0: 0),
- Divide the angle of 360 degrees by the number of LEDs in the circle and assign a radius (Ex: in the circle with 6 LEDs, each LED has been positioned with a difference of 60 degrees and a radius of 11mm).

The genetic algorithm was used to perform the optimization for each LED separately, using the result of the neural network loss function of the FPM image reconstruction model as a fitness function. The result of using the genetic algorithm was good for some LEDs and not for others.

The Ant colony optimization algorithm was also used to optimize the specific location of the LED. The ant colony optimization algorithm (ACO) is a probabilistic technique for solving computational problems which can be reduced to finding good paths through graphs. The main idea of this approach is to calculate the best position for an LED light source by iteration in order to minimize a loss function.

However, the result is not satisfying enough for now: still some differences in the Fourier frequency domain, but the differences produced by this method for image reconstruction are small, so we will further improve this method by integrating self-calibration together with image reconstruction in the same neural network.

2.3 Image segmentation

Segmentation is the process of partitioning digital images into meaningful regions. The analysis of biological high content images often requires segmentation as a first step. Since the parasites are almost always found inside cells, it is useful to segment the cells to guide detection. For training, we need to label each cell as infected or healthy. The different types of patients dataset are:

- CAT01: patient infected with large parasites.
- KPJ0: patient infected with small parasites.
- AD: healthy patient with a lot of platelets.
- LE: healthy patient with a normal platelet rate.

Automatic labeling : As only 4 images were available, it was unlikely to use neural networks for segmentation, so classical methods of segmentation and mathematical morphology were used as follow:

- Otsu method was used to perform automatic image thresholding locally on disks with a radius 4 times that of a cell, to threshold the gray levels corresponding to the background which may vary over the course of the image.
- The watershed algorithm was used to separate different objects in the image, by taking the values below the threshold as cells and the values above as the background, it fills the majority of the hollow cells and eliminates background noise in the image. As a result, we obtain a binary mask which indicates a large majority of the pixels of the background.
- Canny method was used for contours calculation based on mathematical formulas. This response is then optimized to obtain fine contours. Thus, filtering by hysteresis is added to that.

- Then, the circular Hough transformation was used iteratively to locate point alignments in an image and was adapted for circle detection, such that each contour point votes for the circles to which it belongs, then keep the circles with a high number of votes, i.e it's certainly a cell.
- Circular markers are then placed in the detected cells and labeled (with different numbers). Then the background is eliminated to avoid false detections, and watershed is used to fill the detected cells which are not circular.

Regularly, groups of cells appear. As soon as two cells are a bit contiguous, the Hough transformation will mark one, and the watershed will fill both. To deal with this situation:

- We detect if in the labeled objects some are clusters of cells based on (Area,...)
- We calculate on the bounding box of the cluster the contours by the method of Canny.
- Then, we recalculate the circular Hough transformations with a condition on (circles number as proportional to the area of the cell cluster) then set the markers as 0.2 or 0.4 or 0.8 circle radius detected by the Hough transform, then use these markers to make a watershed on the initial cluster. If cells are properly separated, the cutting is kept. otherwise, we remove the cut and start again with markers of radius 0.4 or 0.8. If none of these works, the cluster of cells is cleared and we continue the algorithm.

This method, therefore, gives cell detection performance of approximately 98% for infected patients. The calculation of this performance is done in relation to the error proportion. The error proportion is calculated based on the sum of all missed cells, badly cut cells, unseparated cell clusters, and a slight tolerance for edge clusters to include a maximum of cells.

2.4 Image Classification

The image classification accepts the given input images and produces output classification for identifying whether the disease is present or not. The classification process provides a pre-diagnosis on the contamination of the patient. Image classification is a supervised learning problem: defining a set of target classes (healthy or infected) is usually done by doctors manually, and then the model is trained with deep neural networks to recognize them using labeled example images. If the algorithm has found an infected cell, then the practitioner can visualize it and validate the diagnosis by machine. If the algorithm does not detect any parasites, the result should be taken with caution: the patient will be either healthy or not very infected.

To classify cells, we can create a new deep learning network or use transfer learning. In this work, the pre-trained Resnet 152 neural network was chosen. Res-Net was chosen because it is a low time complexity compared to other networks and has shown good classification results for more than 1000 classes. The network, therefore, has the ability to generalize and extract characteristics of each class. In addition, the network chosen is already pre-trained on the ImageNet database. To use Resnet the input image must be of dimensions (84,84,3) with three channels. In this works only the green channel of the images provided was used, in order to have a better performance in terms of false-negative rate. The phase and intensity of images are as input of the network and the output is the label healthy or infected.

The performance of the algorithm is also analysed: the image segmentation stage that detects possible infected blood cells, the first classifier stage that confirms infections and the second classifier stage that differentiates species. Two measures of algorithm performance and accuracy are used: sensitivity, the ability of the algorithm to detect a parasite present; and positive predictive value (PPV), the success of the algorithm at excluding non-infected cells . These values are expressed in terms of true positives (TP), false positives (FP) and false negatives (FN):

$$sensitivity = TP / (TP + FN) \quad (1)$$

$$PPV = TP / (TP + FP) \quad (2)$$

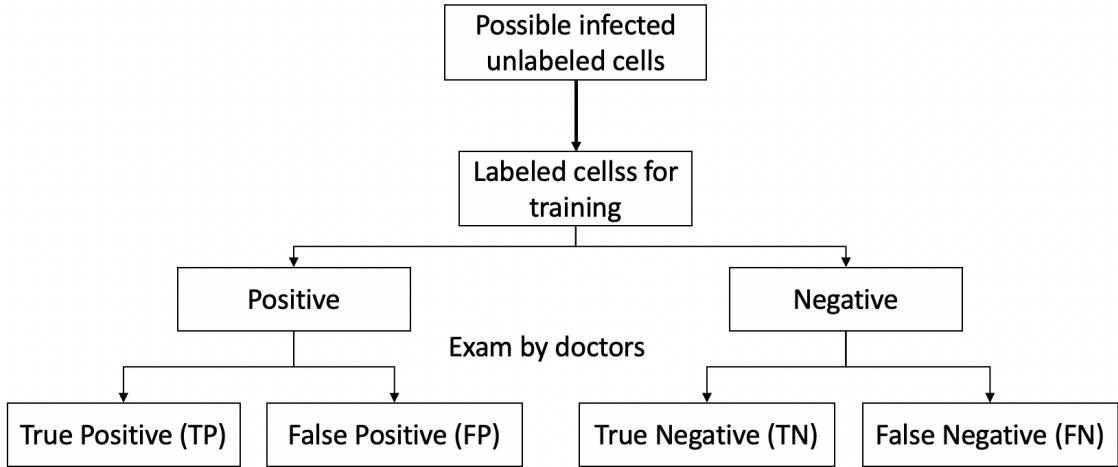


Figure 6: Supervised learning for classification

3 Integrated interactive interface

The aim of designing a Graphical User Interface is to enhance the efficiency and ease of use for the underlying logical design of a stored program, and allow users to interact with information by manipulating visual widgets that allow for interactions appropriate to the kind of data they hold.

Our GUI is named **CellAnnotator**, which is a plugin for Napari viewer. It will help users to annotate cells in biomedical images, taking advantage of napari's capability to view large n-dim images interactively and python's availability to use powerful machine learning algorithms from open-science library, such as sklearn, opencv, scikit-image, or even deep learning libraries such as TensorFlow and PyTorch and keras.

Machine and Deep Learning show remarkable accuracy for segmentation task in papers [12, 13, 14, 15]. However, these models would not work very well on your own data, simply because they are not optimized for your data.

As a result, their predictions are not going to be perfect and almost always need human efforts in the end.

CellAnnotator will provide ML/DL algorithms to help users to segment cells, to classify cells, to correct annotations.

We came up with a few scenarios for different use cases, inspired from existing tools such as Ilastik and AnnotatorJ, and each time we imagine what features the user might need, then we try to outperform the existing tools.

This user interface consists of several plugins which are described below: Image preprocessing 3.1, Image segmentation 3.2, Image reconstruction 3.5, cell image extraction 3.4, cell classification ??, Image saving & loading 3.7 and Contour assist 3.8.

3.1 Image preprocessing plugin

The image preprocessing allows the user to realise simple operations on the input image and to prepare the data for the segmentation part.

This plugin allows for the following operations:

- **Choose a channel:** The user can choose a channel in case of multi channel images, or choose "gray" to get the grayscale of the input image (RGB images), or "None" to pick the raw image.
- **Image smoothing:** The user can perform a gaussian filtering of the image by selecting first the image layer from the list of layers, then selecting a standard deviation value using the SpinBox, then press the button "OK".

- **Difference of Gaussians:** The user can perform a pass band filtering of the image by selecting first the image layer from the list of layers, then selecting a minimum and maximum values of the standard deviation using the 1st and 2nd SpinBoxes, then press the button "OK".
- **Invert the image:** The user can invert the image in case of white background and black foreground by selecting first the image layer from the list of layers, then pressing the "Invert" button.
- **Crop the image:** The user can draw a/many shape(s) using the shape layer then extract the pixels from those shapes(patches) as a new image layer (this could be useful for training a model or making fast processing of the images when dealing with big images), by adding first a shape layer and drawing the shapes, then selecting the source image layer from which the patches will be extracted, then pressing the "Crop" button, this results in rectangular cropped images.

Figure 7, illustrates the different buttons and options that Image processing plugin contains. While, the Figure 8 illustrates the results obtained when applying the crop and invert functions, where, a shape layer is used to draw a patch on the image region to be cropped. Also, since the image has a black foreground, it should therefore be inverted. Finally, we produce an image which will be used as input for the segmentation.

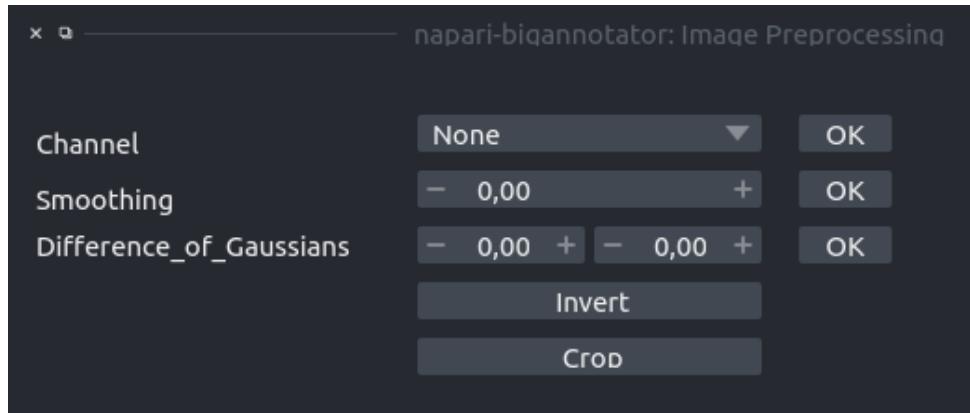


Figure 7: Image preprocessing plugin

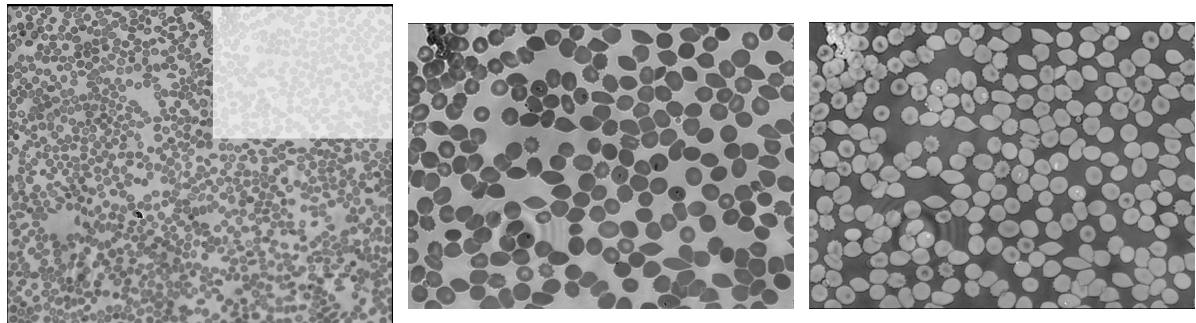


Figure 8: **Image preprocessing results:**

Left: original image+drawn shape, **Center:** cropped image, **Right:** inverted image

3.2 Image segmentation

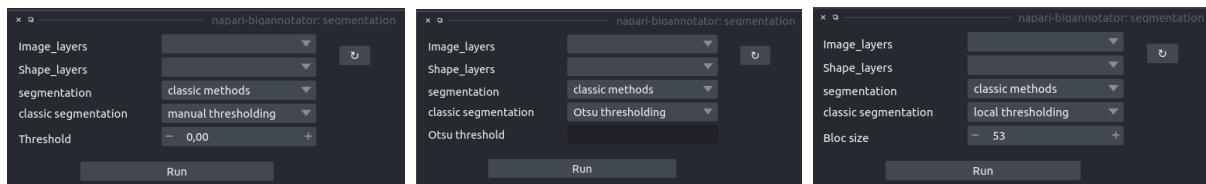
3.3 Image segmentation plugin

The image segmentation plugin allows the user to realise classic and advanced image segmentation on the preprocessed image and to output a mask and label images.

This plugin allows for the following operations:

- **Choice of an image and shape layers:** By clicking on the reload button the list of image/shape layers will be updated. The shape layer is chosen only if the user wanted to perform segmentation only on specific regions of the image, an alternative of this is to crop the regions of interest on the preprocessing step, then perform segmentation on the cropped images.
- **Choice of the segmentation method:** After selecting the image and shape layers (if needed) the user will have to choose between using classic or advanced segmentation algorithms.
- **Classic methods:** If the classic method is chosen, then one of the 3 thresholding based algorithms are suggested.
 - **Manual thresholding:** By selecting this option the user will have to choose a threshold then to press "Run" button to perform segmentation, as illustrated in figure9 top left.
 - **Otsu thresholding:**¹ By selecting this option the user will have to press "Run" button to perform segmentation, then the Otsu threshold is displayed in LineEdit box, as illustrated in figure9 top center.
 - **Local thresholding:** By selecting this option the user will have to choose a bloc size (the characteristic size surrounding each pixel) then to press "Run" button to perform segmentation, as illustrated in figure9 top right.
- **Hough transform :** (sec 2.3) By selecting this option the user will have to set the parameters by pressing the "set parameters" button, then a new window will appear. The parameter's window contain a list of parameters:
 - * **Verbose:** Whether or not to plot all figures while processing.
 - * **Cell mean:** To choose a cells diameter, by default its set to 60.
 - * **Output directory:** To choose a directory where the produced mask and labels will be saved as 'tiff' images.

This is illustrated in figure9 bottom left and right.



- **Advanced methods:** If the user chooses to use advanced methods, then one of the 3 advanced algorithms are suggested.
 - **StarDist:**² By selecting this option the user will have to choose one of the image types from the list of image types, then to press "Run" button to perform segmentation using a pretrained StarDist, as illustrated in figure10 left.
 - **CellPose:**³ By selecting this option the user will have to set the parameters by pressing the "set parameters" button, then a new window will appear. The parameter's window contain two lists of parameters:

¹**Otsu thresholding** Otsu is an automatic global thresholding algorithm, and usually have the following steps: 1)- Process the input image. 2)- Obtain image histogram (distribution of pixels). 3)- Compute the threshold value T (The core idea is separating the image histogram into two clusters with a threshold defined as a result of: minimizing the within-class variance or maximizing the between-class variance).

²StarDist is a cell detection method that predicts a shape representation that is flexible enough. Furthermore, it performs well on images with very crowded nuclei and does not suffer from merging bordering cell instances. [16]

³Cellpose is a generalist model that can segment a wide range of images of cells, without requiring parameter adjustments, new training data or further model retraining. Cellpose uses two major innovations: a reversible transformation from training set masks to vector flows that can be predicted by a neural network, and a large segmented dataset of varied images of cells [17].

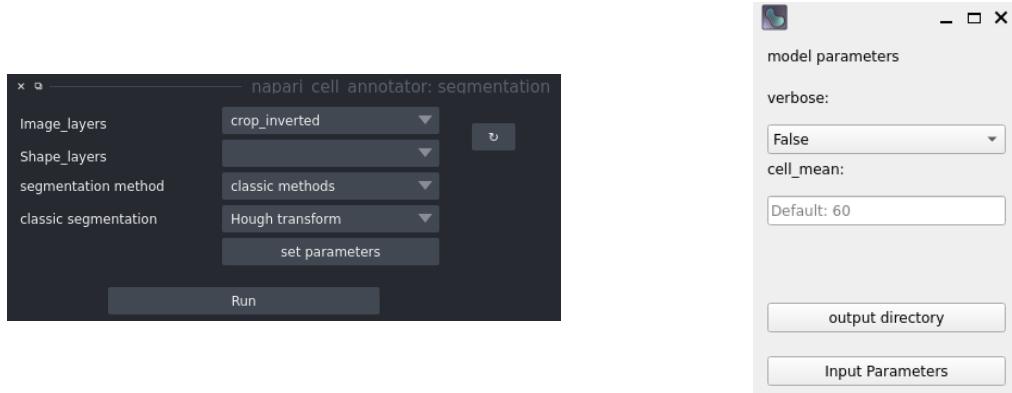


Figure 9: **Image segmentation plugin (classic methods):**

Top left: manual thresholding, **Top center:** Otsu thresholding, **Top right:** local thresholding,
Bottom left: Hough transform, **Bottom right:** Hough parameters

- **Model parameters:** This parameters are used to import the model.

The user will have:

- Whether or not to save model to GPU, will check if GPU available (default: False).
- To choose the model type: ‘cyto’=cytoplasm model; ‘nuclei’=nucleus model; if None, pretrained_model used.
- To set a mean ‘diameter’ (default: 27).
- To set the path to pretrained cellpose model(s).

- **Model evaluation parameters:** This parameters are used for the model evaluation.

The user will have:

- To choose a list of channels, either of length 2 or of length number of images by 2. First element of list is the channel to segment (0=grayscale, 1=red, 2=green, 3=blue). Second element of list is the optional nuclear channel (0=none, 1=red, 2=green, 3=blue), (default is [0,0]).
- To choose a flow error threshold (all cells with errors below threshold are kept), (default: 0.4).
- To choose cell probability threshold (all pixels with prob above threshold kept for masks), (default: 0.0).
- To choose minimum number of pixels per mask, can turn off with -1, (default: 15).

Finally, the user will have to press "Input parameters" then close the parameters window, then press "Run" button to perform segmentation, as illustrated in center and right figures in figure 10.

3.3.1 Image segmentation results

The figures 11 and 12 illustrate a comparison of the results obtained using classic and advanced segmentation algorithms, respectively.

For the classic algorithms, we observe that, Otsu thresholding poorly detects the blood cells, as illustrated in figure 11 left, a green box is drawn to show how is Otsu algorithm segmenting a cluster of cells as when cell (false positive), in the other hand, Hough transform perform very well but have some drawbacks like: false positives (box in green) where two cells are labeled as one, or some cells get an extra mask that contain parts from neighboring cells (in red).

For the deep learning algorithms, we observe that, Cellpose outperform StarDist and results on the best segmentation results. In figure 12 left, we observe that StarDist detect well the cells, but also detect many false negatives (box in blue), whereas, cellpose detects all cells and have no False pos/neg, and both algorithms detect well the overlapping cells.

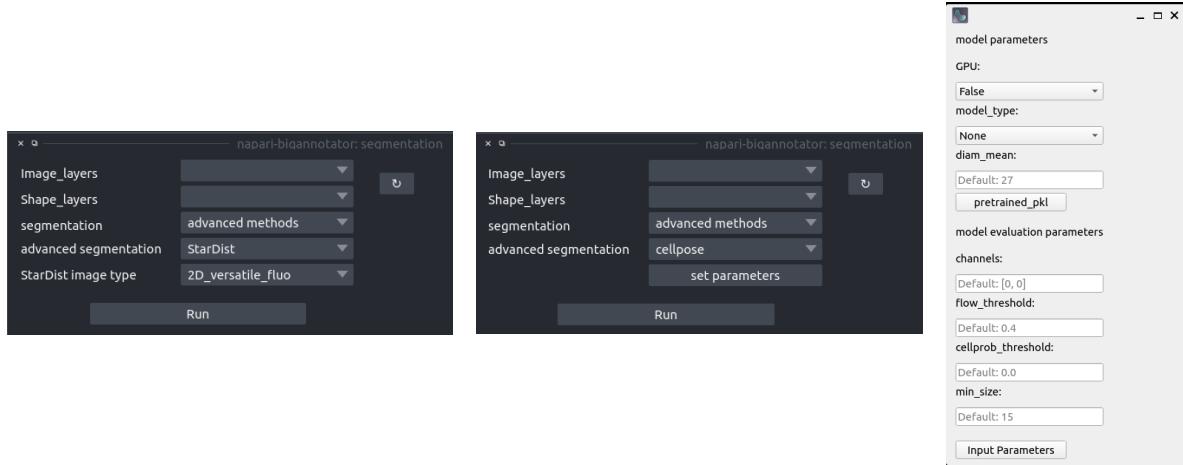


Figure 10: **Image segmentation Plugin (advanced method):**
Left: StarDist, **Center:** CellPose, **Right:** CellPose parameters

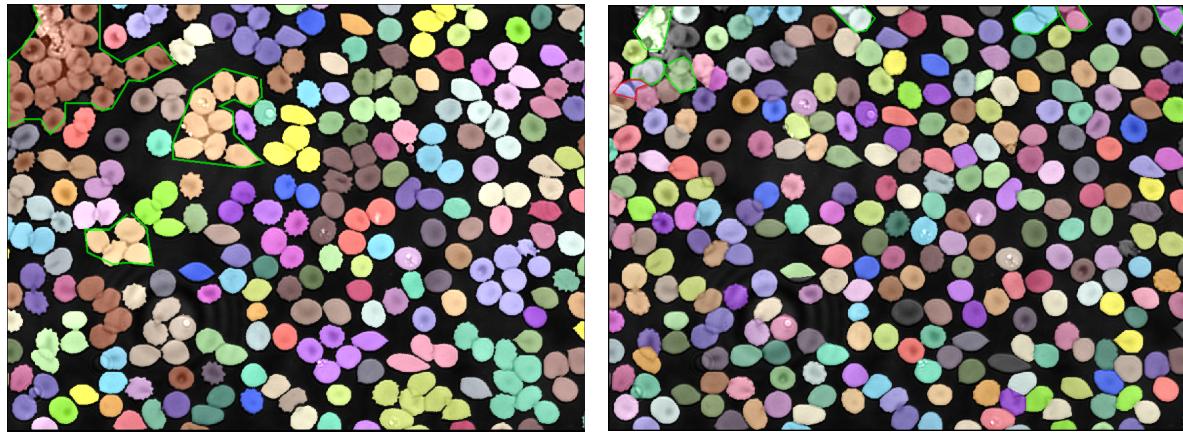


Figure 11: **Image segmentation results (classic methods):**
Left: Otsu thresholding, **Right:** Hough transform

3.4 Cells extraction

After image segmentation, a mask and label images are generated, which indicates cells positions in the image, those images are used for cell extraction and to obtain individual cells images, which will be used later for classification to train/test the classification algorithm. This plugin allows for the following operations:

- **Choice of an image and label layers:** By clicking on the reload button the list of image/label layers will be updated. The label layer image, is the best one obtained from segmentation, in our case its Cellpose, and the image layer is the input preprocessed image.
- **Setting the parameters:** "set parameters" button allows for parameters setting. The parameter's window contain a list of parameters:
 - **Mask:** If 'True', the mask will be used to extract only cells, and outputs a rectangle of fixed size. If 'False', the outputted rectangle will contain the cell with some neighboring cells.
 - **Cell mean:** The cell diameter used for cell segmentation.
 - **Box size:** The size of the rectangle that will contain the extracted cells.
 - **Output directory:** The directory where the extracted cells will be saved as 'png' images.

This is illustrated in left and right figures of figure13.

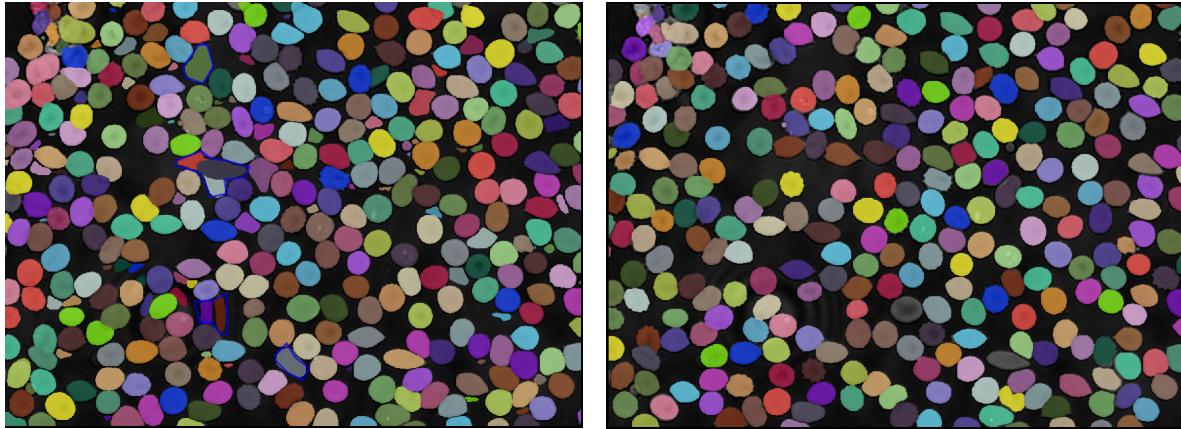


Figure 12: Image segmentation results (Advanced methods):**Left:** StarDist, **Right:** Cellpose

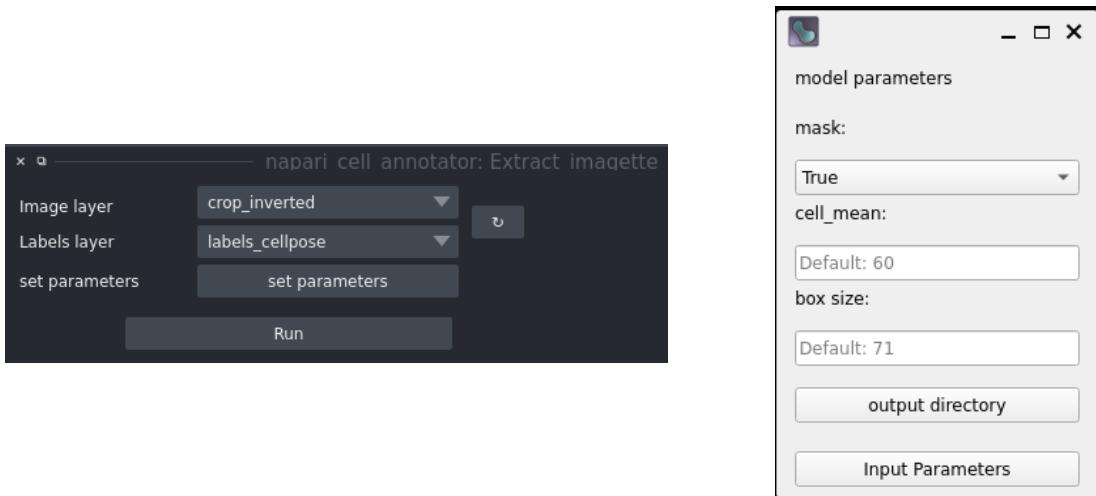


Figure 13: Cells extraction:**Left:** Cells extraction plugin, **Right:** Cells extraction parameters

3.5 Image reconstruction using (FPM)

This plugin could be used even before or after segmentation to obtain a high resolution spatial image and a phase image using Fourier Ptychography algorithm, the recovered spatial image can be used as input to the segmentation algorithm in order to enhance the cells detection by the segmentation algorithms. Also, will be used for cells extraction, which produces the cells that will be used for cell classification. Whereas, phase images will be used after a cell extraction, for cells classification as an additional information about the the cells.

This plugin allows for the following operations:

- **Testing or Training:** By choosing testing, this will allow for loading a pretrained model, and to visualize the 3 images (spatial, phase, Fourier transform), that are the reconstructed images of a low resolution image that was used for training, as well as, producing the high resolution image for the input image.

To do so, the user have to:

- Choose an image
- Set the path to the pretrained algorithm's weights
- Choose a directory where the output images will be saved.

This is illustrated in the left figure of figure15. By choosing training, user have to set the parameters for the training:

- **Index down-sample:** It presents the size ratio between the high and low resolution images.
- **Image size:** The size of the input low resolution image.

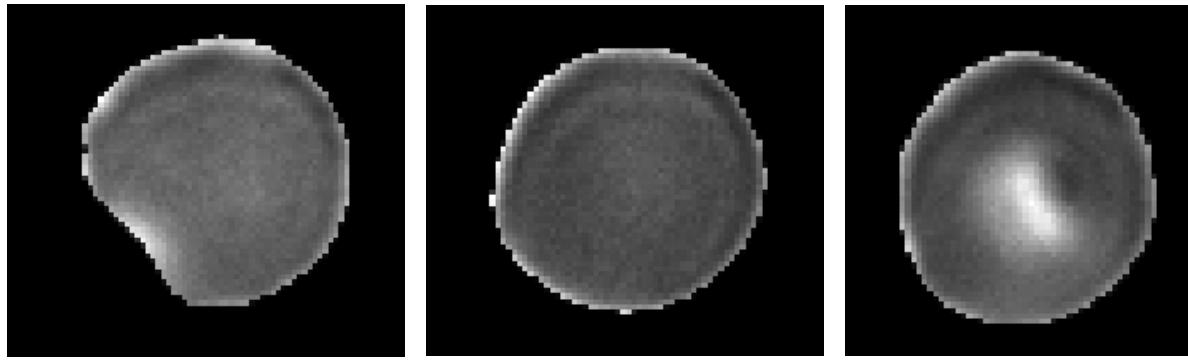


Figure 14: **Cells extraction results:** Left and Center: healthy cells, Right: infected cell

- **Array size:** Whose square represents the number of LEDs used to illuminate the sample and produce the input images.
- **Input image directory:** The directory where the input images, that will be used for the training of the model are situated.
- **Output image directory:** The directory where the output high resolution images will be saved.
- **Save weights directory:** The directory where the best model parameters will be saved.

This is illustrated in center and right figures of figure15.

The produced (phase, spatial and Fourier transform) images are illustrated respectively in figure 16.

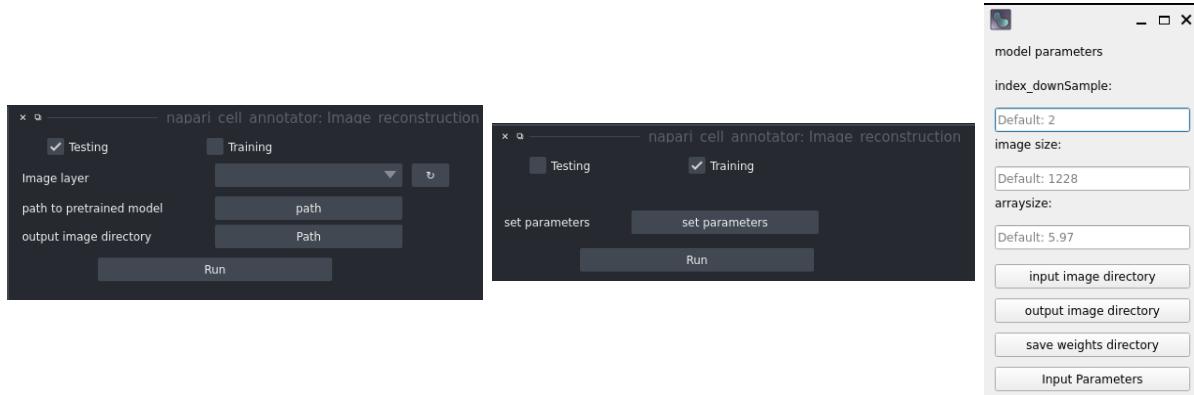


Figure 15: **Image reconstruction (FPM):**
Left: Testing, Center: Training, Right: Training parameters

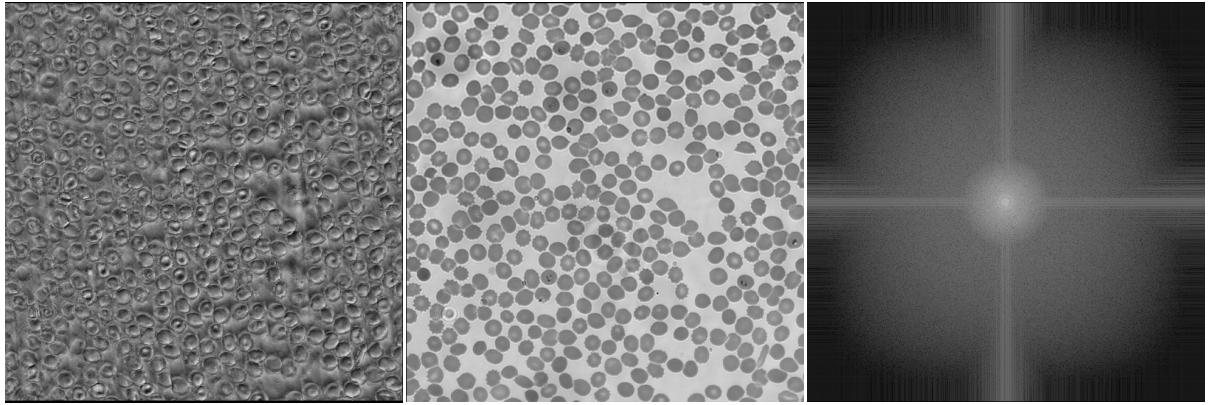


Figure 16: **Cells extraction results:** Left and Center: healthy cells, Right: infected cell

3.6 Image classification

3.6.1 Image classification plugin

This plugin is used to allow for either the training or the testing of the algorithm. In case of training, a ResNet152 (supervised deep learning model) is trained on annotated (intensity and phase) cell images composed of 2 classes (infected and healthy), which are annotated by biologist, and represent the ground truth for classification.

In case of testing, a pretrained ResNet152 weights, are loaded and used for predicting the class if unseen cell images.

A good deep learning algorithm, is the one that can generalise to unseen datasets, which means, can predict correctly for test datasets.

This plugin allows for the following operations:

- **Testing or Training:** By choosing testing, a pretrained model's weights are loaded and used to predict the class of unseen images.

There are two types of images:

- **Internal images:** Which means, images from the list of images present on the interface layers list. Thus, the user will have to set the cell image name, then click 'run classification', this will output the class of that cell on the black empty box. As shown in figure 17 top left.
- **External images:** Which means, to load a list of cell images from an external directory, by indicating the path to that directory. This will output the class of that list of images on terminal. As illustrated in figure 17 top right.

By choosing training, the user will have to set parameters for the training of ResNet152 model:

- **Learning rate:** Which is a tuning parameter in an optimization algorithm, that determines the step size at each iteration while moving toward a minimum of a loss function.
- **Number of epochs:** How many epochs are needed to train the algorithm.
- **Batch size:** Defines the number of training examples in one forward/backward pass. The higher the batch size, the more memory space you'll need.
- **Split training/(validation + test):** If set to 0.8, this mean 80% of dataset is chosen as training set, and 10% for validation and 10% for testing.
- **Input image directory:** The directory where the cell images (phase + intensity) are situated.
- **Output directory:** The directory where the best model parameters will be saved, as well as, the training history (accuracy + loss) at each epoch.

This is illustrated in bottom two figures of figure17.

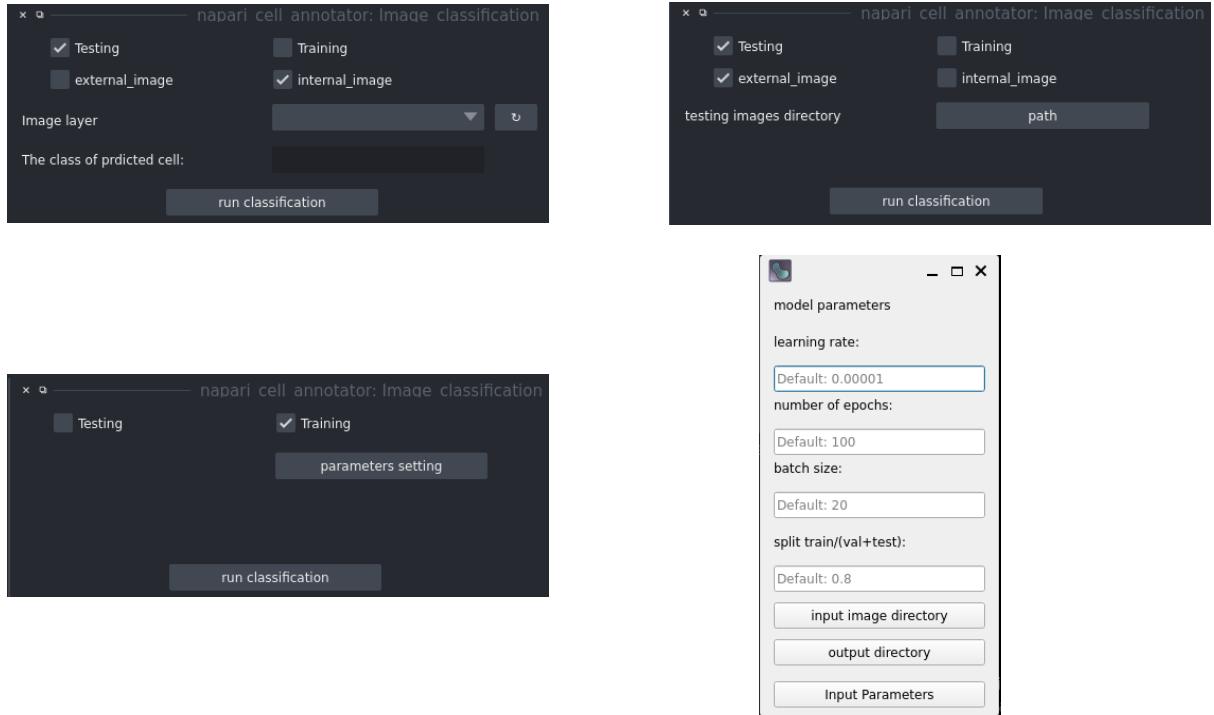


Figure 17: **Image classification plugin:**

Top Left: model testing (internal images), **Top Right:** model testing (external images),
Bottom Left: model training, **Bottom Right:** model training parameters

3.6.2 Image classification results

The ResNet152 model was trained for 300 epochs, with dropout = 0.7 and learning rate=0.0001.

The training set is composed of the reconstructed images in intensity and phase.

The figure 18 illustrate the accuracy and loss classification curves, we observe a very stable accuracy and loss curves, in addition to a faster convergence to reach 99.98% for accuracy, 0.001% for loss.

After training the classification model, we calculate the confusion matrix that illustrate how many false/true positives and false/true negatives the trained model can identify.

In figure 19, we illustrate an example of false positives (Left figure) and false negatives (right figure) identified by trained ResNet152.

In case of false negatives, the model misclassified healthy cells as infected cells, whereas, in case of false positives, the model misclassified infected cells as healthy cells.

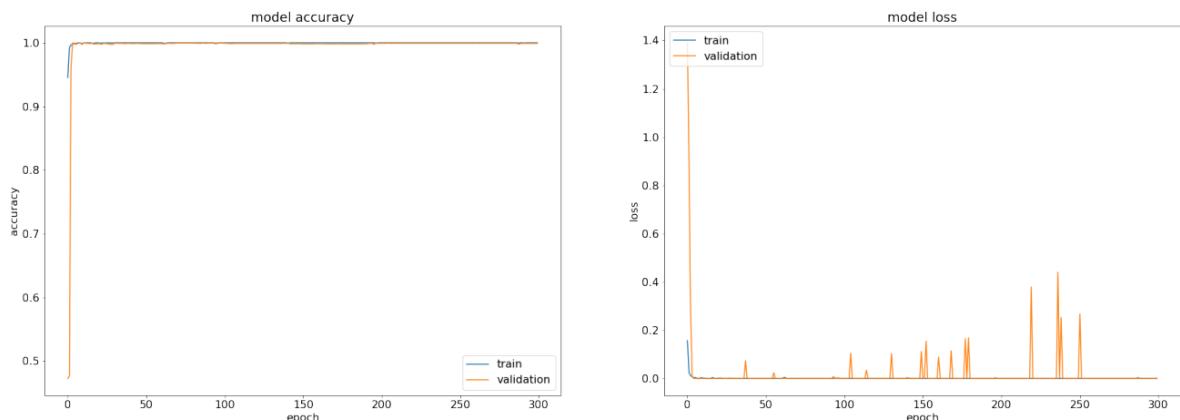


Figure 18: Accuracy and loss classification curves, using FPM reconstructed intensity and phase images.

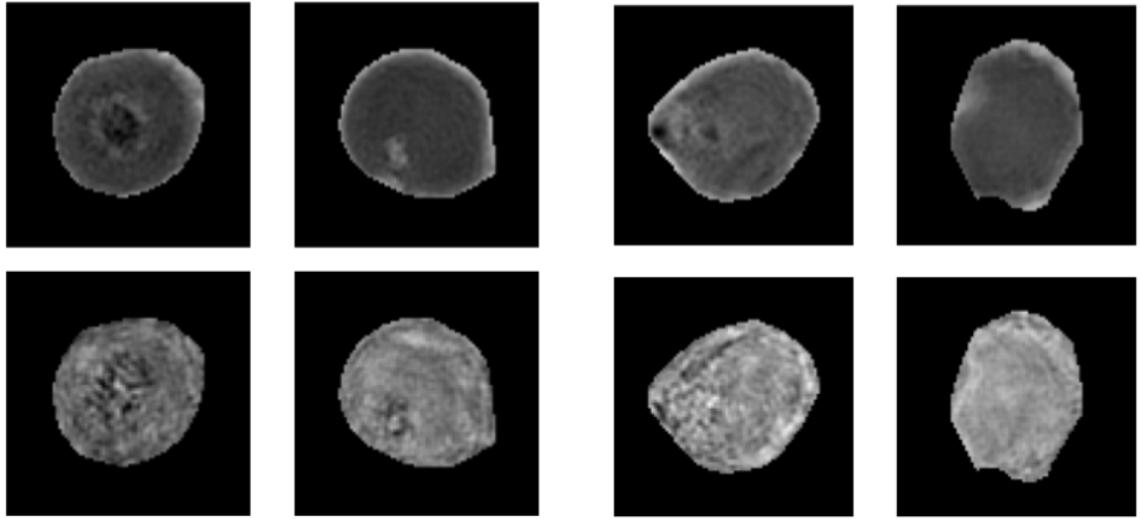


Figure 19: Example of false negatives (Left figure) and false positives (right figure) intensity and phase images, identified by trained ResNet152

3.7 Image saving and loading plugin

This plugin allows the user to save an Image or Labels layer as '.tiff' image and to load an image as Image or Labels layer.

This plugin allows for the following operations:

- **Choose an option:** The user have to first choose either to save or to load an image.
 - **Save an image:** To save images the user have to first select the Image/Labels layers from the layers' list, then to choose a directory where to save those images by clicking the button "dir", then to click "save_load" button, as illustrated in figure 20 left.
 - **Load an image:** To load images/labels the user have to first select the list of images/labels to load by clicking the button "dir", then select the layer's type (image or label), s.t to load an image as Labels it should be of type "int", then click "save_load" button, as illustrated in figure 20 right.

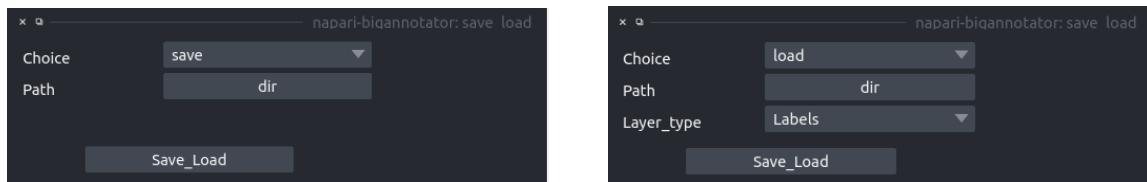


Figure 20: **save_load image plugin:** saving an image (left), loading an image (right)

3.8 Contour assist plugin

This plugin is an initiation to contour assist and consists of two options: contour suggestion, and add modifications.

This plugin allows for the following operations:

- **Option:** The user have to first choose one of the two options: contour suggestion, and add modifications.
- **Contour suggestion:** This option suggests contours and masks based on Region Growing algorithm to segment the selected regions of the image. To select regions on the image the user have to use the Points layer and to put point on the regions that needs a contour/mask suggestion, then to click the refresh button to update the list of images then select the appropriate image layer, then

the user have to set a threshold that defines the gray difference tolerated between a seed and the neighboring pixels, then click "Run". Two Labels layers (contours and masks) will be added to the layers' list, the user can use the brush tool to adjust the suggested mask/contour, as illustrated in figure 21 left.

- **Add modifications:** This option will allow the user to merge two Labels images the raw label (before modification using contour suggestion) and modified label (after modifications). To do so, the user have to click the refresh button, then to select the appropriate raw label and modified label images, then click "Run" button, this will output a final labels layer containing all the labels, as illustrated in figure 21 right.

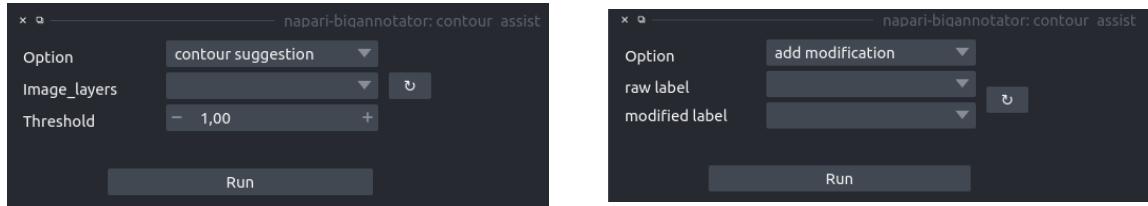


Figure 21: **contour assist plugin:** contour suggestion (left), adding modification (right)

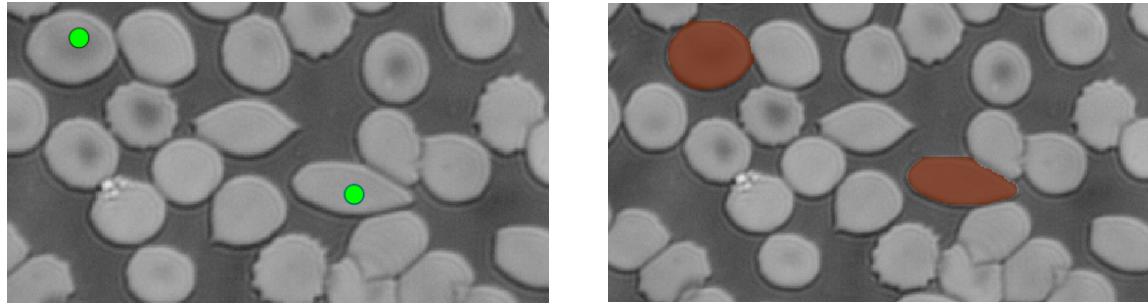


Figure 22: **Contour suggestion example:** selecting seed where non segmented cells are situated (left), masks obtained after applying region Growing algorithm (right)

4 Conclusion

In this report, we first discussed the detection methods of malaria parasites. Among them, the microscope observation equipment is simple, which can intuitively obtain the distribution, size, and number of parasites, so it is the most widely used in clinical diagnosis. However, compared with chemical and biological antibody testing, current microscope imaging technology has low accuracy and poor efficiency. The main reason is that the accuracy of microscopic imaging is not high enough, and optical imaging equipment needs to find a balance between field of view and resolution. Based on this, we proposed Fourier Ptychography Microscopy, an imaging technique that can use multiple low-resolution images to reconstruct a high-resolution image, which has positive significance for malaria detection.

In our report, we introduced the imaging principle and image reconstruction process of Fourier Ptychography Microscopy, and used FPM imaging to obtain a series of red blood cell microscopy images. We perform image reconstruction on these samples to obtain relatively high-definition red blood cell images, and perform image segmentation on the high-definition images, dividing a complete red blood cell image into sub-images containing only one cell in each picture.

Next, we use the image of the cells marked by the doctor as training data, use the convolutional neural network to classify the unlabeled cells, and divide the cells into healthy and infected. We also calculated the correct rate of the classification algorithm.

In the third part of the report, we introduced the interactive interface we designed. This interactive interface integrates image pre-processing, image reconstruction, segmentation, classification, and image post-processing and storage. The interactive interface can provide users with convenient visualization of microscopic image processing of parasites.

References

- [1] Zheng, Guoan, Roarke Horstmeyer, and Changhuei Yang. "Wide-field, high-resolution Fourier ptychographic microscopy." *Nature photonics* 7.9 (2013): 739-745
- [2] <https://www.who.int/news-room/fact-sheets/detail/malaria>
- [3] Hänscheid, Thomas, and Martin P. Grobusch. "How useful is PCR in the diagnosis of malaria?." *Trends in parasitology* 18.9 (2002): 395-398.
- [4] Bian Z, Dong S and Zheng G 2013 Adaptive system correction for robust Fourier ptychographic imaging *Opt. Exp.* 21 32400–10
- [5] Ou X, Zheng G and Yang C 2014 Embedded pupil function recovery for Fourier ptychographic microscopy *Opt. Exp.* 22 4960–72
- [6] Ou X, Zheng G and Yang C 2014 Embedded pupil function recovery for Fourier ptychographic microscopy *Opt. Exp.* 22 4960–72
- [7] S. Jiang, K. Guo, J. Liao, and G. Zheng, "Solving fourier ptychographic imaging problems via neural network modeling and tensorflow," *Biomedical optics express*, vol. 9, no. 7, pp. 3306–3319, 2018.
- [8] Zheng G 2014 Breakthroughs in Photonics 2013: Fourier Ptychographic Imaging Photonics Journal, IEEE 6 1–7
- [9] P. C. Konda, L. Loetgering, K. C. Zhou, S. Xu, A. R. Harvey, and R. Horstmeyer, "Fourier ptychography: current applications and future promises," *Optics Express*, vol. 28, no. 7, pp. 9603–9630, 2020.
- [10] R. Eckert, Z. F. Phillips, and L. Waller, "Efficient illumination angle self-calibration in fourier ptychography," *Applied optics*, vol. 57, no. 19, pp. 5434–5442, 2018.
- [11] Fuhad, K. M., et al. "Deep learning based automatic malaria parasite detection from blood smear and its smartphone based application." *Diagnostics* 10.5 (2020): 329.
- [12] Reka Hollandi, Abel Szkalisity, Timea Toth, Ervin Tasnadi, Csaba Molnar, Botond Mathe, Istvan Grexa, Jozsef Molnar, Arpad Balind, Mate Gorbe, Maria Kovacs, Ede Migh, Allen Goodman, Tamas Balassa, Krisztian Koos, Wenyu Wang, Juan Carlos Caicedo , Norbert Bara, Ferenc Kovacs, Lassi Paavolainen, Tivadar Danka, Andras Kriston, Anne Elizabeth Carpenter, Kevin Smith and Peter Horvath. "nucleAIzer: A Parameter-free Deep Learning Framework for Nucleus Segmentation Using Image Style Transfer". *Cell Systems*, 2020.
- [13] Uwe Schmidt, Martin Weigert, Coleman Broaddus and Gene Myers. "Cell Detection with Star-Convex Polygons". *Medical Image Computing and Computer Assisted Intervention*, 2018.
- [14] Martin Weigert, Uwe Schmidt, Robert Haase, Ko Sugawara and Gene Myers. "Star-convex Polyhedra for 3D Object Detection and Segmentation in Microscopy". *The IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2020.
- [15] Feixiao Long. "Microscopy cell nuclei segmentation with enhanced U-Net", *BMC Bioinformatics*, 2020
- [16] Uwe Schmidt, Martin Weigert, Coleman Broaddus and Gene Myers. "Cell Detection with Star-Convex Polygons". *Medical Image Computing and Computer Assisted Intervention*, 2018.
- [17] Carsen Stringer, Tim Wang, Michalis Michaelos, Marius Pachitariu. "Cellpose: a generalist algorithm for cellular segmentation".*Nature Methods*, 2020.