

**Project Report**

Cell Images for Detecting Malaria

* by AMOL SINHA

A report submitted in part fulfilment of internship at

**DIGINIQUE TECHLABS**

ON

**DATA ANALYTICS, ML and AI ( using PYTHON)**

# ACKNOWLEDGEMENT

I would like to make my deepest appreciation and gratitude to Mr Sudhir Chaudhary for his invaluable guidance, constructive criticism and encouragement during the course of this project.

Thanks to Mr. Bipul Shahi for being uniformly excellent advisor. He was always open, helpful and provided strong broad ideas .

I am also grateful to Diginique TechLabs for providing the necessary opportunities for the completion of this project.

AMOL SINHA

DEPT OF COMPUTER SCIENCE

MANIPAL UNIVERSITY JAIPUR

# CONTENTS

1. INTRODUCTION
2. PURPOSE
3. DATASET

4. SCIKIT-LEARN

5. OPENCV

6. BUILDING AND EXPLORING IMAGE DATASET

8. PLOTTING CELL IMAGE DATA AND PREDICTIONS

9. FURTHER SCOPE

10.CONCLUSION

11. REFERENCES

Abstract

In this project our aim is to identify whether a cell is malaria infected or not. Working with infected and uninfected datasets to save humans by detecting and deploying Image Cells that contain Malaria USING scikit learn and openCV.



# Introduction

Although the malaria virus doesn’t take the form of a mutant mosquito, it sure feels like a mutant problem. The deadly disease has reached epidemic, even **endemic** proportions in different parts of the world — killing around **400,000** people annually [1]. In other areas of the world, it’s virtually nonexistent. Some areas are just particularly prone to a disease outbreak — there are certain factors that make an area more likely to be infected by malaria [2].

* High poverty levels
* Lack of access to proper healthcare
* Political instability
* Presence of disease transmission vectors (ex. mosquitos) [6]

With this mixture of these problems, we must keep some things in mind when building our model:

* There may be a lack of a **reliable power source**
* Battery-powered devices have less**computational power**
* There may be a lack of **Internet connection** (so training/storing on the cloud may be hard!)

While we want to obtain the highest level of accuracy as possible, we must also consider making the model as small and computationally efficient as possible — and also able to be deployed to edge and Internet of Things devices.

Current diagnosing methods of this disease are tedious and time-consuming.



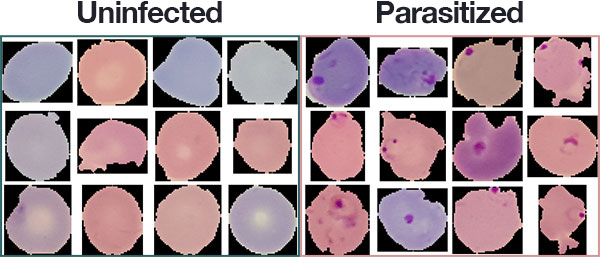
Thin film of red blood cells. Stained parasites are deep purple. [(Source)](http://www.southsudanmedicaljournal.com/archive/february-2011/malaria-in-south-sudan-3.-laboratory-diagnosis.html)

The most widely used method (so far) is examining thin blood smears under a microscope, and visually searching for infected cells. The patients’ blood is smeared on a glass slide and stained with contrasting agents to better identify infected parasites in their red blood cells.

Then, a clinician manually counts the number of parasitic red blood cells — **sometimes up to 5,000 cells**(according to [WHO protocol](http://www.wpro.who.int/mvp/lab_quality/2096_oms_gmp_sop_09_rev1.pdf))

Purpose

Our primary focus is to classify the malaria cell images as 'Parasitized' or 'Uninfected'. However, with a little experimentation, visualization and analysis, we can get a deeper meaning of the provided images, which in turn might help us in creating better features for a more efficient model.



### Dataset

Thankfully, we have a great dataset of labeled and preprocessed images to train and evaluate our model. NIH has a malaria dataset of 27,558 cell images with an equal number of parasitized and uninfected cells. A level-set based algorithm was applied to detect and segment the red blood cells. The images were collected and annotated by medical professionals; more information can be found [here](https://ceb.nlm.nih.gov/repositories/malaria-datasets/). Also, download the data from the page: the file is named cell\_images.zip

# Scikit-Learn: A Python Machine Learning Library

Scikit-learn was initially developed by David Cournapeau as a Google summer of code project in 2007.

Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python.

It is licensed under a permissive simplified BSD license and is distributed under many Linux distributions, encouraging academic and commercial use.

The library is built upon the SciPy (Scientific Python) that must be installed before you can use scikit-learn. This stack that includes:

* **NumPy**: Base n-dimensional array package
* **SciPy**: Fundamental library for scientific computing
* **Matplotlib**: Comprehensive 2D/3D plotting
* **IPython**: Enhanced interactive console
* **Sympy**: Symbolic mathematics
* **Pandas**: Data structures and analysis

Extensions or modules for SciPy care conventionally named [SciKits](http://scikits.appspot.com/scikits). As such, the module provides learning algorithms and is named scikit-learn.

The vision for the library is a level of robustness and support required for use in production systems. This means a deep focus on concerns such as easy of use, code quality, collaboration, documentation and performance.

OpenCV

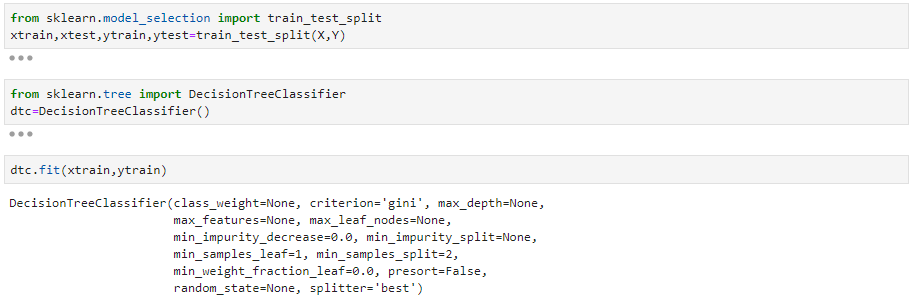
(Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code.

The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality, etc. OpenCV has more than 47 thousand people of user community and estimated number of downloads exceeding [18 million](https://sourceforge.net/projects/opencvlibrary/files/stats/timeline?dates=2001-09-20+to+2019-01-30). The library is used extensively in companies, research groups and by governmental bodies.

## Building and exploring image dataset

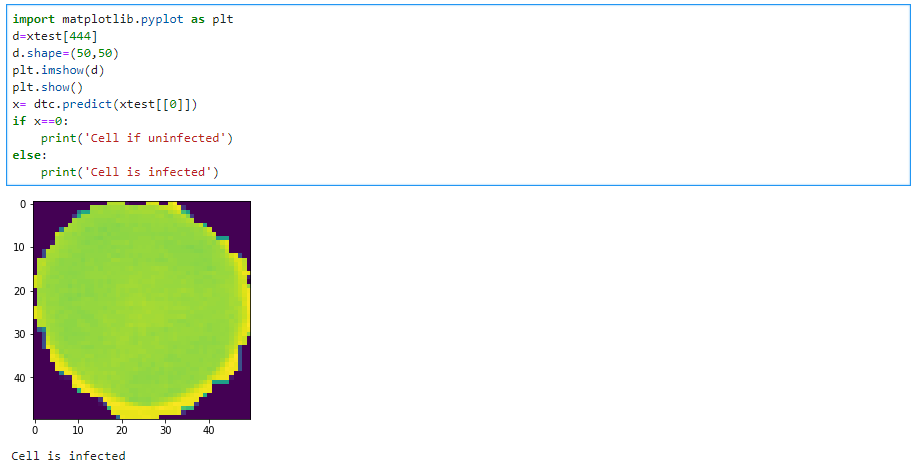
To build machine learning models, we need training data, but we also need to test the model's performance on unseen data. We will use a 60:10:30 split for train, validation, and test datasets, respectively. We will leverage the train and validation datasets during training and check the performance of the model on the test dataset.

The images will not be of equal dimensions because blood smears and cell images vary based on the human, the test method, and the orientation of the photo. Let's get some summary statistics of our training dataset to determine the optimal image dimensions



Plotting cell image data and predictions

We use matplotlib library along with resizing the image data for proper prediction whether the input cell is infected or not.



### Further Scope:

* Using blob\_dog or blob\_log detection techniques, hoping for better accuracy at the expense of time
* Using the fact that the parasitic part of the cell is usually of a different colour than the cell colour.
* We can obviously use Keras Neural Networks efficiently to get a much better accuracy score. However, we often use Keras as a black-box, not knowing exactly how we determine whether a cell is parasitized or not.

Conclusion

We loaded the data, created our train & test datasets and analyzed the images to obtain a good idea regarding many aspects of malaria parasites. Through this method, by using the openCV lib, we were able to create a hight accuracy system.

Moreover, if we used keras or deep learning algorithms, a lot of different things could have been done.

All of the deep learning articles published so far have concentrated on thin blood smears, but it is very likely that we will see articles for thick films very soon. Given the wide acceptance of deep learning, the importance of large annotated data image repositories for training is now widely understood, leading to a great support of data acquisition efforts. This will likely lead to larger test suites on patient level, allowing for more standardized evaluations and extensive field testing. Given these developments, automated microscopy is very much in the race toward a cheap, simple, and reliable method for diagnosing malaria.

References

[1] Michal Drozdzal et al. “The importance of skip connections in biomedical image segmentation”. In: Deep Learning and Data Labeling for Medical Applications. Springer, 2016, pp. 179–187.

[2] Marc Macenko et al. “A method for normalizing histology slides for quantitative analysis”. In: 2009 IEEE International Symposium on Biomedical Imaging: From Nano to Macro. IEEE. 2009, pp. 1107–1110.

[3] <https://www.kaggle.com/sakshat/malaria-cell-images-exploratory-data-analysis>

[4] <https://opensource.com/article/19/4/detecting-malaria-deep-learning>