# Computer Vision for Malaria Cell Detection

Gabriel Mennesson (gm1936), Christopher Osufsen (cro257), Youchen Liu (yl3558), and Benedict Teoh (bwt223)

## 1. Introduction

### 1.1 The Disease

Malaria, a disease carried by Anopheles mosquitoes, is one of the world's most prolific, with 219 million estimated cases in 2017 resulting in 435,000 deaths. Nearly half the world's population is at risk. Although curable, there is no vaccine for the disease, and prevention is predominantly done by taking antimalarial pills, which reduce risk of infection by 90%.[1]

Unfortunately, a high proportion of those at risk live in some of the world's poorest areas. Specifically, the WHO African region (including countries like the Central African Republic, Congo and Ethiopia) carried 92% of worldwide malaria cases and 93% of deaths. These nations lack resources and medical professionals to properly diagnose, track and treat the disease. Early diagnosis and treatment of malaria reduces symptoms, prevents death, and prevents the further spread of the disease.[2]

### 1.2 Current Detection Methods

**Clinical Diagnosis** is the most common detection method of malaria. It is a diagnosis from a doctor based off symptoms and physical findings at an examination. This method is limited as the first symptoms of malaria are not specific to the disease, and are found in many other diseases such as flu viruses. Additionally, doctors in areas where malaria is uncommon

[1] https://www.nhs.uk/conditions/malaria/prevention/
[2] https://www.cdc.gov/malaria/diagnosis_treatment/diagnosis.html

may not be familiar with the disease at all, and as such may be unlikely or unwilling to make a positive malaria diagnosis.

Another detection method is **Microscopic Diagnosis**, which is a diagnosis based on the examination of a giemsa-stained blood smear under a microscope. The giemsa differentially stains human and pathogenic cells, allowing the examiner to visually identify the presence of the malaria-causing bacteria in a person's blood. While microscopic diagnosis is considered the "gold standard" for the diagnosis of malaria, there are a number of issues.[3] It requires a lab and significant other medical infrastructure, as well as trained personnel such as lab technicians who are familiar with the procedure. This makes it financially infeasible for low-income areas, which typically suffer from high malaria rates. Additionally, results are only available in days to weeks time, which is serious given the mortality rate of malaria in high-risk regions.

**Antigen Detection** in the form of RDT (Rapid Diagnostic Test) kits is also used in the diagnosis of malaria. These kits are available for purchase and can detect antigens derived from malaria parasites. The kits provide very quick results, usually in 2-15 minutes. However, RDT kits are approved in the US only for hospital and commercial labs, and not by individual clinicians or patients. They also are not replacements for microscopy, as they may not be able to detect infections with lower numbers of malaria parasites circulating in the bloodstream. In addition, two less common species of malaria, *P. ovale* and *P. malariae* aren't conclusively detected by the kits, so negative RDT results need to be followed by microscopy regardless. RDTs are also not cheap, with a 12-pack costing $330. Considering this is nearly a year's salary in the Democratic Republic of the Congo, this detection method is not feasible in high-risk areas.

[3] https://www.cdc.gov/malaria/diagnosis_treatment/diagnostic_tools.html#tabs-1-1

**Molecular Diagnosis (PCR)**, whereby parasitic acids are detected using polymerase chain reactions (PCR), is another method of malaria detection.This method provides accurate results, however diagnosis is slow, with results often not available quickly enough to be of value in diagnosis. Molecular diagnosis is more useful in confirming the specific species of malaria after the existence of the parasite has already been confirmed through another method.

**Serology** is also used in malaria detection, and uses an indirect fluorescent antibody test to identify past malaria exposure. It is very useful for determining possible sources of malaria from blood transfusions or chronically infected patients. It is limited however in the sense that it does not detect current malarial infection, and as such it cannot be used as a real-time method for diagnosing malaria.

### 1.3 Takeaways from Current Methods

Although there are many methods to detect malaria, only one is consistently reliable, and can produce results quickly enough to be of use: microscopic diagnosis. However, microscopic diagnosis laboratories are obviously resource and labor intensive, which is especially problematic in poor regions with few trained staff. The status quo could be improved by either reducing cost of testing, lab technician hours spent, or increasing the speed of diagnosis (typically 2-15 days).

### 1.4 Emergence of Smartphone Microscopes

Smartphones have become more available across the world, and even low-income countries such as Ethiopia have 12 million smartphone users, with 11.2% adoption in 2018.[4] With the ubiquity of the smartphone comes the increased availability of microscopes, as

[4] https://en.wikipedia.org/wiki/List_of_countries_by_smartphone_penetration

smartphone microscopes have been developed that are able to simply plug into the devices, and save photos directly to the phone. These microscopes are available cheaper than conventional microscopes, and can be purchased for under $200. There are also (allegedly) high quality smartphone microscopes available for anybody to 3D print, developed by Australian researchers. [5] Due to the availability and relative affordability of smartphone microscopes, we expect them to be utilized more, especially in countries who lack medical resources

### 1.5 Data Mining: How & Why?

In a situation in which thousands of photos of potentially malaria-infected cells might be taken in low-income countries every day, we think data mining is an effective solution to process them, and help quickly, cheaply and correctly diagnose more patients infected with malaria. Our solution is one of binary classification; is the photo of an uninfected or an infected cell? If we successfully implement this classification model, we expect to be able to help over-encumbered lab technicians save time and resources by determining very quickly, at scale, if cells are infected with malaria. We do this using blob detection, and convolutional neural networks.

## 2. Data Set

We accessed a World Health Organization (WHO) dataset from their website, which contained 27,558 images of treated blood cells split into two folders of normal and infected cells. These images are photos of thin blood smear slides taken using Android smartphones with attached light microscopes. *P. falciparum* is the deadliest pathogenic species that causes malaria in humans, and its presence in blood cells can be visually identified using a giemsa stain at the

[5] https://www.engadget.com/2018/02/20/3d-printed-smartphone-microscope-is-good-enough-for-scientists/

microscopic level, which differentially stains pathogenic and human cells. To generate the dataset, thin blood smear slides from 150 infected and 50 healthy patients were collected, stained, and photographed (using the smartphone's built-in camera transfixed with a microscopic lens) at Chittagong Medical College Hospital in Bangladesh. These images were then manually annotated by an expert slide reader at the Mahidol-Oxford Tropical Medicine Research Unit in Bangkok. We will be using this image data as inputs to the system.[6]

This is a binary image classification of stained blood cells with a boolean target variable representing the existence of malaria in the cell image. The patient cell is either infected / parisitzed (1) or uninfected (0). It is a supervised learning task given the presence of a target variable.

Example instances from the dataset:

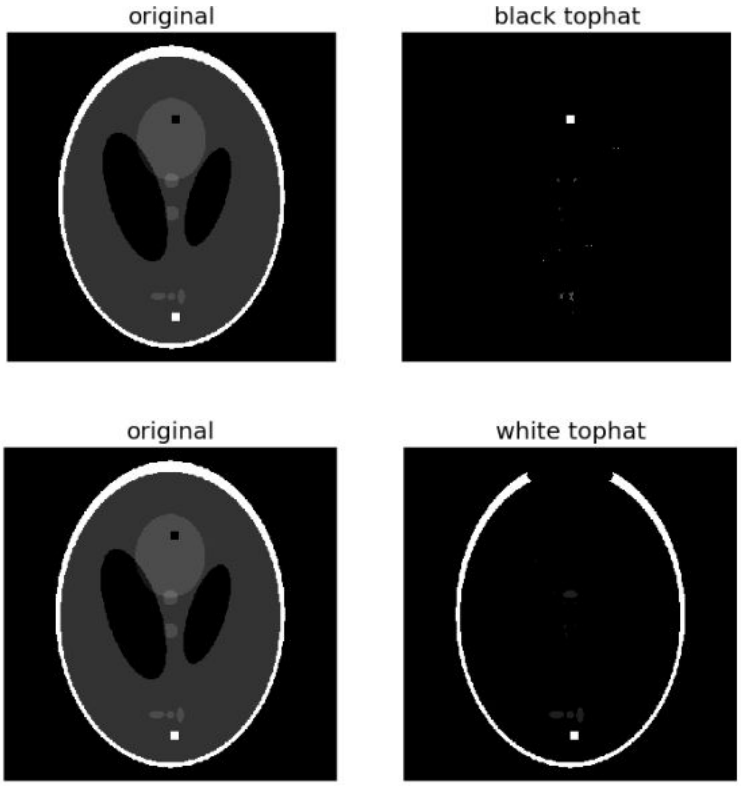

## 3. Data Preparation

[6] https://ceb.nlm.nih.gov/repositories/malaria-datasets/

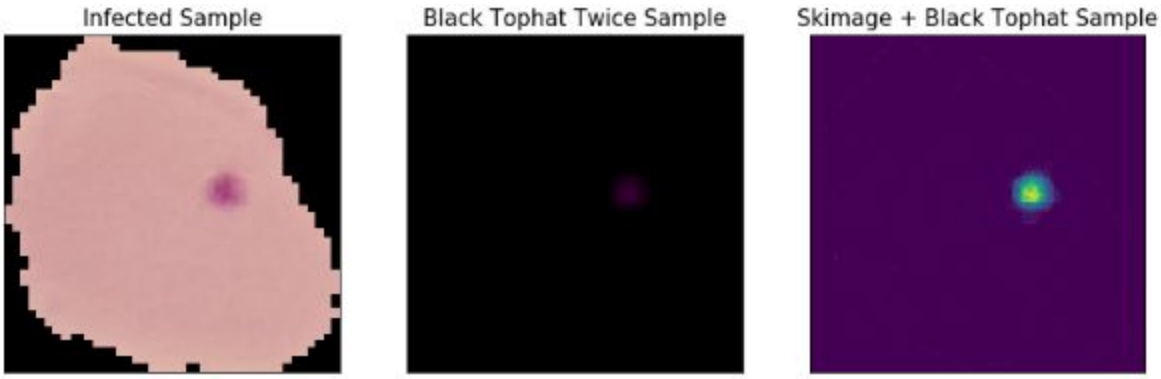
Data preparation for raw image data was crucial to our task. We used PIL, OS, Pandas, numpy, sklearn, and skimage in the data preparation process. First we imported images one at a time into an empty np.zeros array of dimensions (# images, 125, 125, 3). This allowed for us to store each image as a 125x125 resolution RGB image in an array. We used the PIL image library for this process. Since infected and uninfected images were in two seperate folders we initially imported them as two seperate arrays, respectively denoted. We then created arrays of labels, the infected labels were an array of all 1's and uninfected an array of 0's sized respectively. Next we applied sklearn train_test_split method to split uninfected and infected data/labels into 75% training and 25% testing subsets. We combined infected and uninfected training data and randomly shuffled indices to evenly combine instances. The same process was repeated on our testing data.

With the data loaded, labeled, randomized and split we looked to manipulate the images to make classification easier. The goal was to "excite" the dark purple areas of our images, which represented the parasite marking, and reducing the intensity of the rest of the image. We utilized various methods in the skimage.morphology library, specifically we settled on testing variations of black_tophat and white_tophat applications. At a basic level applying black_tophat to an image returns only the dark spots in an image that are smaller than the "structural elements."[7] We can think of structural elements in our case as the cell outline. Applying white_top returns the opposite, only smaller light spots. A comparison of the image manipulation effects are seen below:

[7] Skimage.morphology.black_top_hat documentation, https://scikit-image.org/docs/0.12.x/auto_examples/xx_applications/plot_morphology.html




We tested multiple applications of the two morphology methods, such as applying each once, apply each twice, and applying the two in succession. Since the infected spots tend to be a darker purple marking we found more success in applying black_tophat. Our final approach for data used in our models entailed applying black_tophat to the image twice and then skimage.color function to make the image much brighter. A summary of the process on an infected sample is seen below:



With these results we felt strongly about applying our classification models to this stripped down data. The important dark markers on infected cells were vibrant and obvious for most samples and it would be the task of our models to discern their class.

# 4. Methods

## 4.1 Blob Detection

In exploring our data we noted that infected cells were differentiated from uninfected cells by bright purple blobs. The bright blobs are the result of the Giemsa-staining process the blood smears have been exposed to. The process is meant to make the cells visible discernable. After our significant data preparation we had processed image data with vibrant and obvious "blob" markings. With this in mind we explored various blob detection methods. We chose to move forward with the skimage blob library due to its completeness and ease in controlling statistically significant thresholds. Blobs are defined as either bright on dark or dark on bright regions on an image. The three methods we compared were: Laplacian of Gaussian (LoG), Difference of Gaussian (DoG), and Determinant of Hessian (DoH). Laplacian of Gaussian is the slowest, but most accurate model. It computes the Laplacian of Gaussian images with successively increasing standard deviation and stacks them up in a cube. Blobs are local maximas in this cube.[8] Difference of Gaussian is a faster approximation of the LoG approach. In this case the image is blurred with increasing standard deviations and the difference between two successively blurred images are stacked up in a cube.[9] For both LoG and DoG approaches only bright blobs on a dark background are recognized. Determinant of Hessian is the fastest approach. It detects blobs by finding maximas in the matrix of the Determinant of Hessian of the

[8] Definition of Skimage Blob_LoG
[9] Definition of Skimage Blob_DoG

image. The detection speed is independent of the size of blobs as internally the implementation uses box filters instead of convolutions. Bright on dark as well as dark on bright blobs are detected. The downside is that small blobs (<3px) are not detected accurately.[10] For each method detected blobs are returned as coordinates with a radius and a corresponding standard deviation.

To turn the LoG, DoG, and DoH blob methods into classification models was simple. In addition to using these methods we utilized sklearn, PIL, skimage, pandas and numpy for training the model. The model approach followed this logic: if an image input into a blob method returned any number of blobs with standard deviations over a certain threshold we would count this as an infected cell (1) and those that produced no blobs were classified as uninfected (0). Each model was trained by adjusting three thresholds: min_sigma, max_sigma, and threshold. Min_sigma is intuitively the lower bounds of the standard deviation of identified blobs. Lowering this minimum would allow us to identify smaller blobs. Similarly max_sigma was the upper bounds of standard deviation and allow us to filter out very large blobs, such as accidentally classifying the whole cell as a blob. The threshold parameter acts as a lower bound to for scale space maxima, any local space maxima smaller than this threshold are ignored. By reducing this threshold the methods are able to identify less intense blobs. During training, predictions were saved into an array and the compared to the training labels to generate accuracy, confusion matrices, and other evaluation metrics.

This model is highly effective in providing instant classification results, an important aspect for the problem at hand. Since the intensity of classification is focused in finding optimal weights in the training phase, the processing of any new image is instantaneous. Even classifying

[10] Defintion of Skimage Blob_DoH
https://scikit-image.org/docs/dev/auto_examples/features_detection/plot_blob.html

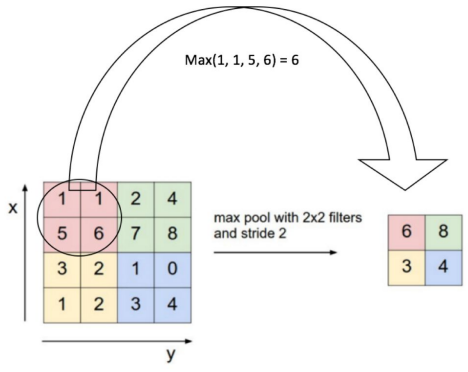
a few thousand images at once takes under a minute. Given the high accuracy of our results, discussed later, and the speed of results we believe this is a promising approach to malaria detection.

**4.2 Convolutional Neural Network**

We chose to use a convolutional neural network as another method of classification This was achieved using keras with a tensorflow backend, opencv, and sklearn for various parts of our data manipulation, shuffling, and array splitting. Our theory is that the convolution will be a good approach to search for areas excitement in the raw pixel values of the image, once we manipulate the data to highlight the infected areas. The build of the model is as follows:

- Two layers of convolution, both using 32 3x3 filters that we use to create our feature maps from the images. These feature maps are the result of taking the dot product between our filter images and the inputted image along various parts of the image. The results are new images, or feature maps, which are used as features.
- For each round of convolution, we included a max pooling step to reduce the number of dimensions in the image. This process involves taking a 2x2 matrix, sliding across the image and only keeping the largest element out of what it covered.
    - This is intended to include only the most important features from the input image, the infected areas on the cell, which usually look like small blobs.[11]

[11] "An Intuitive Explanation of Convolutional Neural Networks." The Data Science Blog

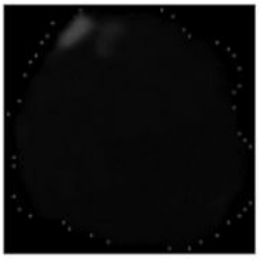



-

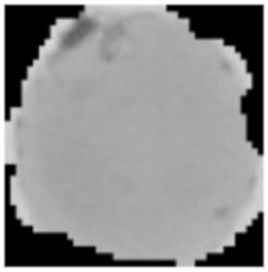
- We used dropout to avoid overfitting. This process ignores the use of different nodes in the fully connected layer of the neural network during training, so that they can limit the codependency on each other and help to reduce the likelihood of overfitting.
- Once convolution is finished, it is fed into the neural network to train and create our weights, utilizing a final softmax activation function.

We adjusted some of these parameters a bit, and even tried incorporating a third layer of convolution, but the preliminary results proved to be the best with this less complex, two-convolution process. It was run with a batch size of 16 for just 20 epochs to help limit our need for computational power as we tested and compared on different data inputs. Because we settled on this model, the main variation in performance (discussed in our results section) was stemming from our data manipulation and inputs for the model. We tried six different tactics of manipulation for this. Below are the methods and what a parasitized image looked after their application:

No Manipulation

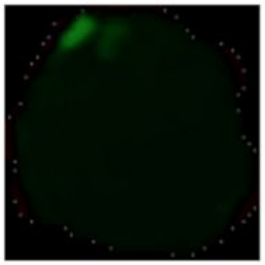


black_tophat Grayscale



Grayscale Conversion

2 application of
black_tophat Grayscale



black_tophat Color

black_tophat Binary

## 5. Result Evaluation

### 5.1 Blob Detection Evaluation

The three blob detection models based on LoG, DoG, and DoH approaches were trained on 75% split of our data set, which equates to 20,669 images. Initial evaluation on a smaller subset (~10,000 images) was done to select appropriate thresholds for each approach. We tested thresholds by iterating over five inputs for each threshold and comparing results. The optimal

weights for LoG blob detection were threshold = 2.5*1e-5, min_sigma = 4, max_sigma = 8. We quickly found that the LoG model was not optimal due to its sensitivity in finding very small blobs. This resulted in a high level of false positives and a maximum accuracy of 85.90%. The optimal weights for blob DoG were threshold = 2.5*1e-4, min_sigma = 2, max_sigma = 8. The accuracy was higher at 94.73% with the following confusion matrix:

| | (True) p | (True) n |
|---|---|---|
| [Predicted] Y | 3263 | 181 |
| [Predicted] N | 182 | 3264 |

The optimal weight for DoH were threshold = 2.5*1e-9, min_sigma = 2, max_sigma = 25. This resulted in an accuracy of 92.83% with the following confusion matrix:

| | (True) p | (True) n |
|---|---|---|
| [Predicted] Y | 3278 | 327 |
| [Predicted] N | 167 | 3118 |

From these results we felt that the DoG approach was optimal, as it was able to more accurately classify data and is more sensitive to small blobs than the DoH approach. In the DoH threshold we needed to increase the maximum standard deviation to very high levels to avoid a large amount of false negatives, but this also increased our false positive count significantly. The tradeoff between accuracy and speed of DoG makes us consider this the optimal model.

A next step now that we have seen our test accuracy on this data set would be to revisit the DoG model, with a specific intent on finding the model with lowest false negative count (highest precision). This makes sense for the task of malaria detection as false negatives are far more harmful than false positives. It is difficult for us to initially quantify these amounts but below is our initial thought process. A false negative is associated with a large negative cost:

potential for spreading of disease, sickness, delay in care and potentially death. The cost of a false positive could also be significant, with medical expenses and aid being allocated to a healthy individual. The benefit of a true positive are the inverse of a false negative: stopping the spread of disease, immediate treatment to combat sickness and prevent death. There is no significant cost or benefit we associate with a true negative.

**5.2 CNN Evaluation**

The convolutional neural network was tested using a sample of 2000 images (1000 parasitized, 1000 uninfected), and ran for 20 epochs on each of our six inputs for it. We tried different batch sizes for training (16, 32, and 48), and found that for 20 epochs, 16 was the best. It was difficult to test on many different types of models due to our restrictions on computational power. We were using a tensorflow back-end with no GPU. However, the 20 epochs gave us enough information to choose what data manipulation tactic was best for us:

```
Epoch 20/20
1600/1600 [==============================] - 82s 51ms/step - loss: 0.3140 - acc: 0.9325 - val
_loss: 0.3216 - val_acc: 0.9425
```

The neural network achieved its best test accuracy of 94.25% when using the images with two rounds of black_tophat filtering and grayscale conversion. The following are some more metrics with these results (the confusion matrix is in the bottom left):

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| class 0(uninfected) | 0.92 | 0.97 | 0.94 | 201 |
| class 1(parasitized) | 0.96 | 0.92 | 0.94 | 199 |
| micro avg | 0.94 | 0.94 | 0.94 | 400 |
| macro avg | 0.94 | 0.94 | 0.94 | 400 |
| weighted avg | 0.94 | 0.94 | 0.94 | 400 |

```
[[194   7]
 [ 16 183]]
```

The accuracy greatly varied depending on the type of manipulation we did. The key was to isolate the infected parts of the parasitized cell and keep all other parts of the cell constant across both types of cells. The two applications of black_tophat with grayscale did it the best, because when it was an infected cell, it only showed the blob of infection, while the uninfected cells were primarily just black images. However, the rest of the image inputs gave very different results for images of the same class. For example, one round of black_tophat with no grayscale conversion would produce some uninfected cell transformations that varied greatly from the transformations of other uninfected cell images. This clearly confused the model as the uninfected cells looked much different from each other, and it became difficult for even a human to detect which type of class it was. Another example was with binary conversion (black and white). The conversion produced uninfected cells sometimes looking like an all white circle, and sometimes as an empty, black picture, despite them looking similar with no manipulation at all. This was likely because of the difficulty in choosing a color scale threshold to convert to binary with. The manipulations where this problem was present gaveus accuracy results under 70%.

The one round black_tophat with grayscale gave us an accuracy of 87.1%, while the two round gave us our best neural network accuracy of 94.25%.

We recognize the limitations with this approach. It is a relatively complex model for some very simple images, and only using the 2000-image subset reduced our need for computational power, yet forced us to use less data. However, given the small amount of images, and only running for 20 epochs, we are happy with what we were able to produce from this, and know there is much room for improvement as we continue to refine the model in the future.

## 6. Deployment

The most obvious method of deployment is to use our model as an accurate, cheap, and easy to access method to diagnose infected patients. This would take the form of a smartphone application, since the images that we trained our data on were taken using a smartphone, which would be relatively cheap and easy to build. Smartphone microscopes would be distributed across areas where there is limited medical access, and the WHO could set up cheap, non-intensive stations for diagnosing at-risk individuals. This would assist in earlier diagnoses of malaria, having a marked effect on mortality rates. Data collection on documented cases of malaria would also reduce the burden on medical clinicians by codifying the diagnosis step.

Our method could also be used more broadly as a tool to track patterns of the disease over geolocation and better monitor for breakouts. Efficiency in getting aid and resources to infected regions would improve significantly with the use of a tool providing real-time tracking of malaria outbreaks. The WHO would be more informed in their allocation of important medical resources, improving the survivability of malaria and other life-threatening diseases. Early detection also leads to a reduction of transmission, which is a major risk for more

congested, less wealthy parts of the world. One of the WHO's key goals is to increase data collection in order to better respond to outbreaks of disease, and our product would certainly help them in their pursuit of that goal.

A key advantage of our product is that it is scalable. Our blob detection model was easily able to process the entire training set and proved a high accuracy solution. While we were forced to train our neural network on only 2,000 images due to limitations on computational power, our results were strong enough that the model would run with relatively high accuracy. There are also steps that could be taken in an actual deployment to ensure that accuracy improves by a greater margin. On-site personnel could send slide images to a central hub where they would be modeled against a growing dataset, and results would then be reported back to the site. Ground level infrastructure is also minimal considering diagnosis only involves the use of a smartphone. Additionally, building our model into a smartphone application serves a key purpose in making the final product portable. This is key for high-risk areas, which often already have sparse medical access at best. Our product could be quickly and cheaply adopted in these regions, and upon success, its deployment could be easily scaled up.

There are key risks involved in this project, including some of an ethical nature. Firstly, there is a risk of political instability reducing our ability to implement our platform in high-risk areas. The nations in which malaria is most prevalent are prone to revolution, and government instability. For example,  malaria is the leading cause of death in the Central African Republic, which is also currently embroiled in a civil war.[12] Maintaining stable relationships with

[12] https://www.severemalaria.org/countries/central-african-republic

healthcare providers and governments may prove to be a challenge, and worsening political conditions could prove a crutch to scaling our product in key regions.

Another risk with this project involves the potential mishandling of sensitive and private information. As with anything of a medical nature, the data that will be collected is confidential and private, and thus our solution must take this into consideration. Strict measures would need to be taken to ensure the security of the data within our platform, especially since the primary source of data is smartphones. This risk is especially pronounced in the case of an American company: foreign governments might be hesitant to allow our management of the data of their citizens if they are not confident our security is competent.

## 7. Conclusion

We believe that there is high potential for data mining in medical applications, shown in our ability to produce an accurate model and deployment strategy with our limited time and resources. A team with ample computational power, data, and expertise could greatly expand upon what we have created. We strongly believe that applications such as this will be influential in impacting accessibility to healthcare, especially in less affluent areas.

# References

[1] “An Intuitive Explanation of Convolutional Neural Networks.” *The Data Science Blog*, 29 May 2017, ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/.

[2] https://keras.io/backend/

[3] https://www.youtube.com/channel/UCWN3xxRkmTPmbKwht9FuE5A

[4] https://www.youtube.com/channel/UCjAOM-s-f2YfeLNGGZuY-Bg

[5] https://scikit-image.org/docs/dev/auto_examples/features_detection/plot_blob.html

[6] https://scikit-image.org/docs/0.12.x/auto_examples/xx_applications/plot_morphology.html

[7] https://www.cdc.gov/malaria/diagnosis_treatment/diagnostic_tools.html#tabs-1-1

[8] https://www.nhs.uk/conditions/malaria/prevention/

[9] https://www.cdc.gov/malaria/diagnosis_treatment/diagnosis.html

[10]https://www.engadget.com/2018/02/20/3d-printed-smartphone-microscope-is-good-enough-for-scientists/

[11] https://ceb.nlm.nih.gov/repositories/malaria-datasets/