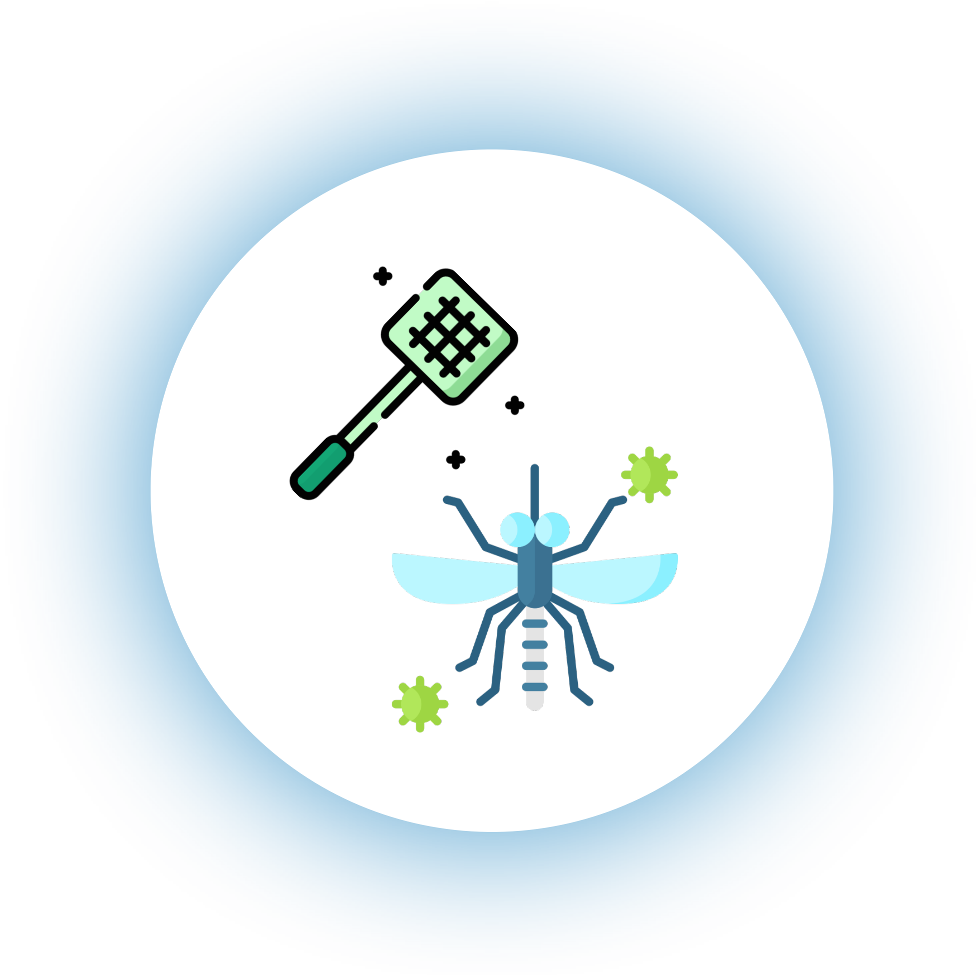
Convolutional Neural Networks for Malaria Diagnosis

An additional tool in my BAMM framework, here we train a CNN to accurately diagnose malaria - a potentially life saving tool in a world of such scarce medical resources.



Zach Wolpe 2020 WLPZAC001

Convolutional Neural Nets for Malaria Diagnosis

Machine & statistical learning techniques can aid the fight against malaria in a multitude of ways - that extend far beyond parameterising mathematical compartmental models. To illustrate an example of other possible applications of statistical learning, consider the problem of accurate, complete diagnosis.

Malaria diagnosis

As aforementioned, in attempts to analyse any disease, we good, accurate data is imperative. Unfortunately it is almost always problematic. In the case of malaria, testing individuals in deeply impoverished communities is arduous. Assuming we can get access to members of those communities, our two testing options are:

- Utilise a testkit
- Blood examination by an expert

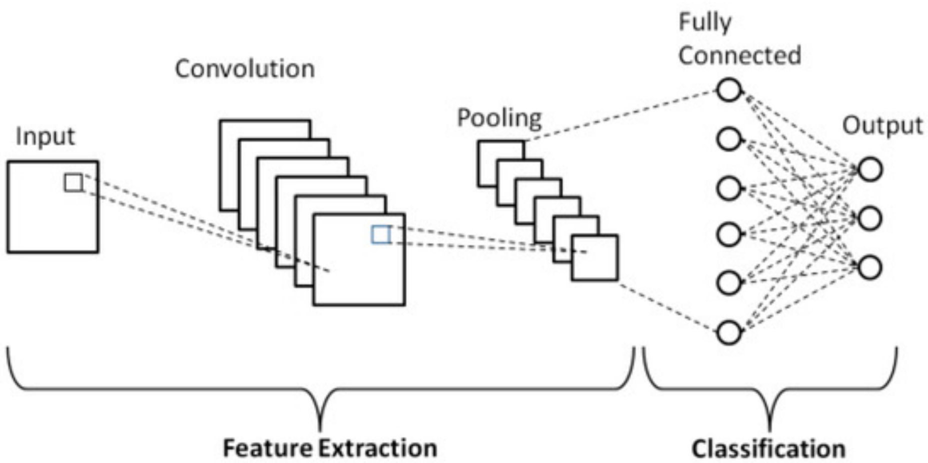
The former is the simplest & most reliable option, however resources are scarce & it may be infeasible to provide enough testing kits. The latter is only applicable if we have access to a physician.

Here I pose an alternative that elevates these resource constraints.

A Convolutional Neural Network

CNNs (convolutional neural networks) are a machine learning model architecture that using convolutional techniques to process high-resolution image data to allow for pattern discover. One can consider a colour image a (3 x n x p) dimensional sparse numerical matrix where n & p are the number of pixels & 3 represents the corresponding RGB values.

CNNs process images for pattern discover, often doing so far superior to human experts (one can think of a human expert as simpler running a more complex variant of this model architecture in their mind - we call this expertise). Here's a typical CNN architecture:



CNNs for Malaria Diagnosis

In the instance of malaria diagnosis, an expert would examine a blood sample & try to identify the parasite in the blood sample. Here we apply a CNN to detect the parasite, given an image.

Dataset

The labeled dataset utilised contains 27'558 images of blood samples - even split between infected & uninfected - and is available on the National Library of Medicine's (NLM) website [7].

Results

The data was split into training & validation sets, thereafter the model was trained to detect parasites in the blood samples. The model architecture is available in the appendix & all code is available on the GitHub.

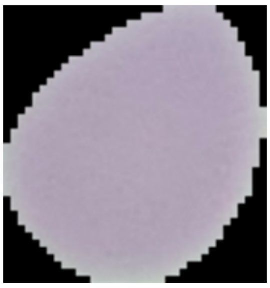
$$Accuracy = 90\%$$

90% accuracy was achieved - which may cause alarm of overfitting, thought the data was split into training-testing sets. I do not feel this result to be particularly alarming as the nature of the data is archetypical of the type of issues that CNN's perform particularly well on. Additional metrics were not examined as false-positive/negatives are roughly evenly distributed, but would follow as a natural extension.

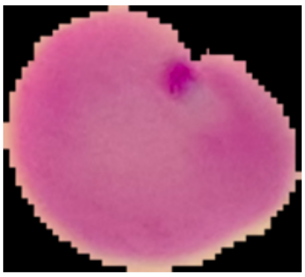
Not only can this algorithm be used anywhere where a blood sample can be taken & there is computer access; it is able to provide more accurate results than a trained professional. It's marginal cost is zero, & does not warrant medical expertise past the ability to draw blood.

The dashboard available at link allows one to run predictions on the validation set, here are 5 sample predictions. One can see the prediction ('parasitized' or 'uninfected'), the model's confidence in the prediction & whether or not the prediction is accurate. Note: 4/5 of the predictions are accurate, whilst the inaccurate classification is ambiguous to the human I (parasites are not obviously visible) & the model took it's stance with far less conviction than in the other examples.

```
[[1]]
[1] "Parasitized"

[[2]]
[1] 0.99722

[[3]]
[1] "Accurately Classified"
```



```
[[1]]
[1] "Uninfected"

[[2]]
[1] 0.9836111

[[3]]
[1] "Accurately Classified"
```



```
[[1]]
[1] "Parasitized"

[[2]]
[1] 0.9986827

[[3]]
[1] "Accurately Classified"
```

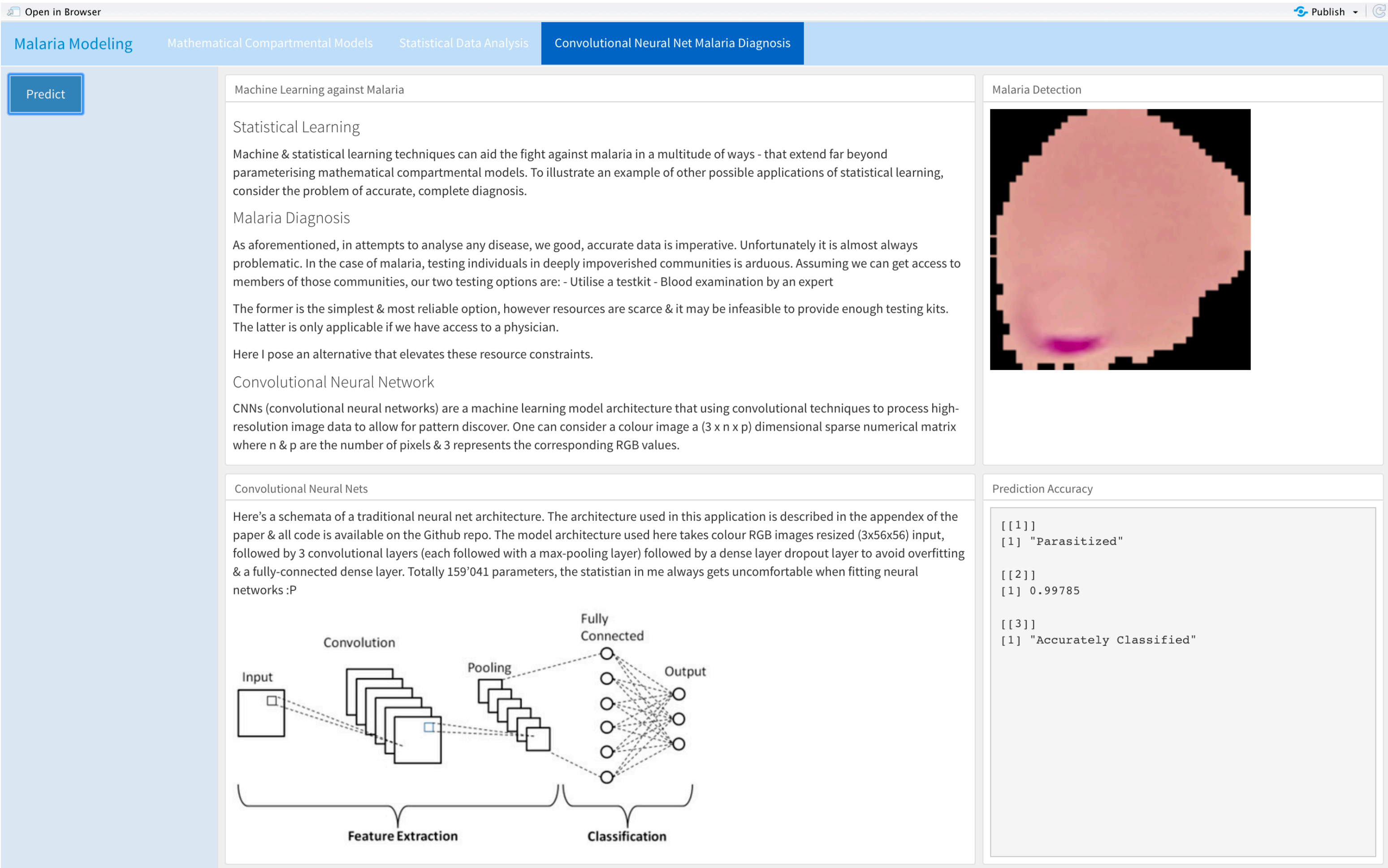
```
[[1]]
[1] "Uninfected"

[[2]]
[1] 0.9947206

[[3]]
[1] "Accurately Classified"
```

```
[[1]]
[1] "Uninfected"

[[2]]
[1] 0.8329843

[[3]]
[1] "Misclassified"
```

Dashboard Preview

The dashboard will go live with the rest of the analysis. Here is a screenshot of real-time predictions classifying a blood sample as '*parasitized*' or '*uninfected*'.

Launch date: 29 June 2020



Convolutional Neural Network Model Architecture:

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 30, 30, 32)	896
max_pooling2d (MaxPooling2D)	(None, 15, 15, 32)	0
conv2d_1 (Conv2D)	(None, 13, 13, 64)	18496
max_pooling2d_1 (MaxPooling2	(None, 6, 6, 64)	0
conv2d_2 (Conv2D)	(None, 4, 4, 128)	73856
max_pooling2d_2 (MaxPooling2	(None, 2, 2, 128)	0
flatten (Flatten)	(None, 512)	0
dropout (Dropout)	(None, 512)	0
dense (Dense)	(None, 128)	65664
dense_1 (Dense)	(None, 1)	129

Total params: 159,041
Trainable params: 159,041
Non-trainable params: 0