REFERENCES

- [1] M. Poostchi, K. Silamut, R. J. Maude, S. Jaeger, G. Thoma. (2018). Image analysis and machine learning for detecting malaria [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5840030/
- [2] WHO. (2019, Dec. 4). The "World malaria report 2019" at a glance [Online]. Available: https://www.who.int/news-room/feature-stories/detail/world-malaria-report-2019
- [3] S. Rajaraman, S. K. Antani, M. Poostchi, K. Silamut, Md. A. Hossain, R. J. Maude, S. Jaeger, G. R. Thoma1. (2018, Apr. 16). Pre-trained convolutional neural networks as feature extractors toward improved malaria parasite detection in thin blood smear images [Online]. Available: https://peerj.com/articles/4568/
- [4] Expert System Team. (2020, May. 6). What is Machine Learning? A definition [Online]. Available: https://expertsystem.com/machine-learning-definition/
- [5] A. Deshpande. (2016, Jul. 20). A Beginner's Guide to Understanding Convolutional Neural Networks [Online]. Available: https://adeshpande3.github.io/A-Beginner%27s-Guide-To-Understanding-Convolutional-Neural-Networks/
- [6] H. Pokharna. (2016, Jul. 28). The best explanation of Convolutional Neural Networks on the Internet [Online]. Available: https://medium.com/technologymadeeasy/the-best-explanation-of-convolutional-neural-networks-on-the-internet-fbb8b1ad5df8
- [7] Stanford University (n.d.). CS231n Convolutional Neural Networks for Visual Recognition [Online]. Available: https://cs231n.github.io/convolutional-networks/
- [8] Worldometer (2017). GDP per Capita [Online]. Available: https://www.worldometers.info/gdp/gdp-per-capita/

MAL.ARIA
IDENTIFICATION MADE SIMPLE

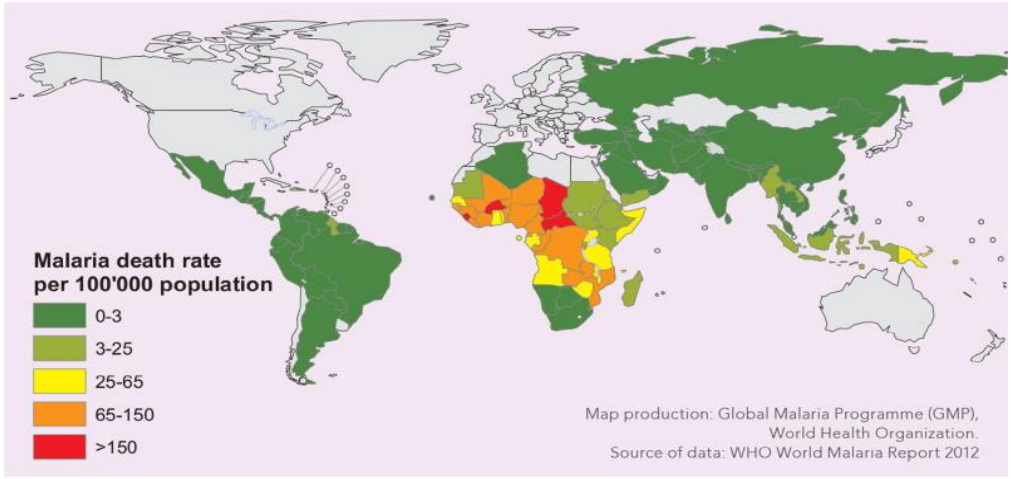
ASSESSING MALARIA BLOOD SAMPLES USING DEEP LEARNING

MAI.ARIA

IDENTIFICATION MADE SIMPLE

MALARIA

- Major impact on global health
- 2018: estimated 228 million cases worldwide & 405,000 deaths [2]
- Only 6 countries account for more than 50% of all cases [2]
- Nigeria accounts for 25% of all cases [2] and has 34% of the world average GDP (per Capita 2017) [8]



[1]

MAL.ARIA
IDENTIFICATION MADE SIMPLE

WHY USE AI?

- Yearly hundreds of millions of blood films examined by a trained microscopist
- This involves manual counting of parasites in red blood cells, which is timely
- Microscopists in poorer areas may have low quality training
  - Low-resource areas and poor-quality control setting
- Lead to incorrect diagnosis
  - False positives and False negatives
- Standardise using AI
- Faster testing
- Reduced workload
- More reliable (with an accurate model)

MaI.ARIA
IDENTIFICATION MADE SIMPLE

MACHINE LEARNING

- Allow a system to automatically learn without being explicitly programmed
- Computer sees images differently to us, so we need a way to aid its ‘learning’
- Computers see images as arrays of numbers
  - Each pixel has 3 values to create an RGB colour e.g. 180, 50, 200 creates pink
- Convolutional Neural Networks…

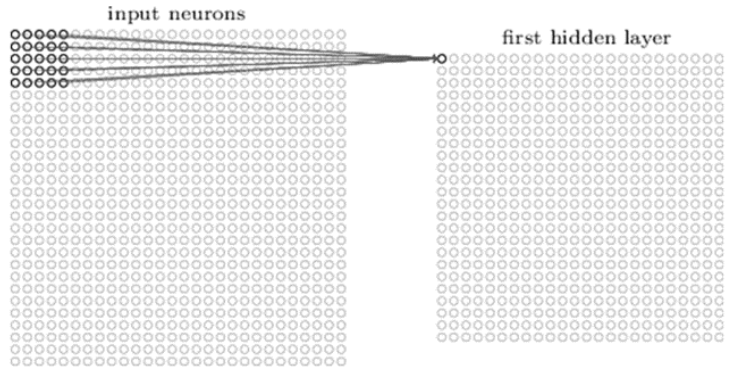
[5]

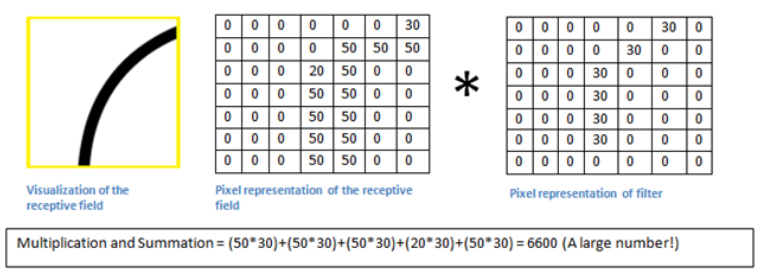
MaI.ARIA
IDENTIFICATION MADE SIMPLE

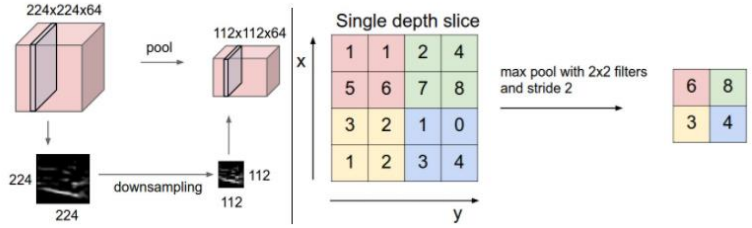
CNNs AS IMAGE CLASSIFIERS

[5]

- CNNs are based on the visual cortex (1962 D. Hubel & T. Wiesel)
- CNNs essentially take an image and pass it through a multiple layer network and identify features that overtime become more abstract e.g. edges
- Pre-processing
- Convolutional layers
  - Filters, feature maps, convolving
- Rectified Linear Activation Function or Rectified Linear Unit (ReLU)
- Pooling layers
- Fully connected layers







MALARIA
IDENTIFICATION MADE SIMPLE

CNNs – TRAINING AND TESTING

- Good rule of thumb for splitting the dataset – 70/30 or 80/20
- What makes the algorithm actually ‘work’
- Training process – backpropagation
  - Gradient descent - technique for minimising loss
  - ½ (desired output – real output) ^{2}
  - Caches values on forward pass
  - Backwards pass determines which inputs contributed to the overall loss
  - Adjusts weights accordingly (weight update) via a parameter called learning rate
- Test by comparing outputs to ground truth

MAL.ARIA
IDENTIFICATION MADE SIMPLE

USEFUL LIBRARIES

- TensorFlow
- Keras
- Pandas
- NumPy

MAL.ARIA
IDENTIFICATION MADE SIMPLE

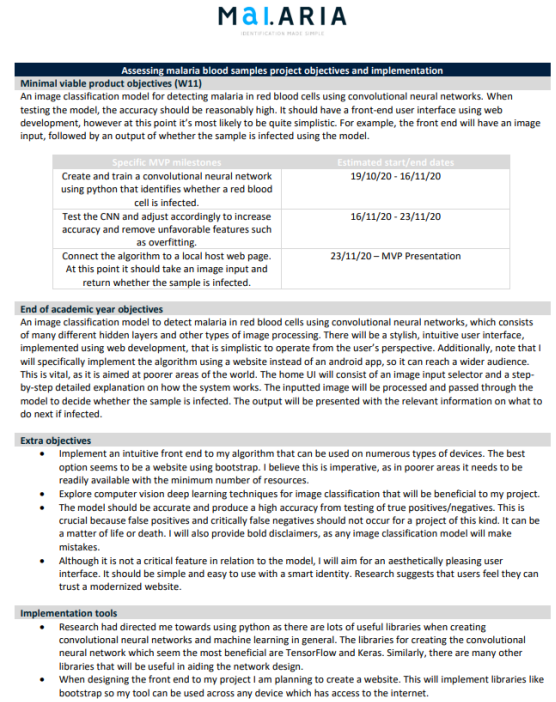
FRONT END UI

- Simple
- Very accessible
- Requires smallest number of resources possible
  - Internet capable device
  - Internet connection
- Bootstrap for device compatibility
  - PC and mobile
- Brand

MAL.ARIA
IDENTIFICATION MADE SIMPLE

MVP OBJECTIVES/IMPLEMENTATION

- Image classification model for detecting malaria in red blood cells using convolutional neural networks
- Relatively high accuracy model
- Link algorithm to front end user interface
- Simple front end UI that takes an image input and outputs whether sample is infected



MAL.ARIA

Assessing malaria blood samples project objectives and implementation

Minimal viable product objectives (W11)

An image classification model for detecting malaria in red blood cells using convolutional neural networks. When testing the model, the accuracy should be reasonably high. It should have a front-end user interface using web development, however at this point it's most likely to be quite simplistic. For example, the front end will have an image input, followed by an output of whether the sample is infected using the model.

Specific MVP milestones	Estimated start/end dates
Create and train a convolutional neural network using python that identifies whether a red blood cell is infected.	19/10/20 - 16/11/20
Test the CNN and adjust accordingly to increase accuracy and remove unfavorable features such as overfitting.	16/11/20 - 23/11/20
Connect the algorithm to a local host web page. At this point it should take an image input and return whether the sample is infected.	23/11/20 – MVP Presentation

End of academic year objectives

An image classification model to detect malaria in red blood cells using convolutional neural networks, which consists of many different hidden layers and other types of image processing. There will be a stylish, intuitive user interface, implemented using web development, that is simplistic to operate from the user's perspective. Additionally, note that I will specifically implement the algorithm using a website instead of an android app, so it can reach a wider audience. This is vital, as it is aimed at poorer areas of the world. The home UI will consist of an image input selector and a step-by-step detailed explanation on how the system works. The inputted image will be processed and passed through the model to decide whether the sample is infected. The output will be presented with the relevant information on what to do next if infected.

Extra objectives

- Implement an intuitive front end to my algorithm that can be used on numerous types of devices. The best option seems to be a website using bootstrap. I believe this is imperative, as in poorer areas it needs to be readily available with the minimum number of resources.
- Explore computer vision deep learning techniques for image classification that will be beneficial to my project.
- The model should be accurate and produce a high accuracy from testing of true positives/negatives. This is crucial because false positives and critically false negatives should not occur for a project of this kind. It can be a matter of life or death. I will also provide bold disclaimers, as any image classification model will make mistakes.
- Although it is not a critical feature in relation to the model, I will aim for an aesthetically pleasing user interface. It should be simple and easy to use with a smart identity. Research suggests that users feel they can trust a modernized website.

Implementation tools

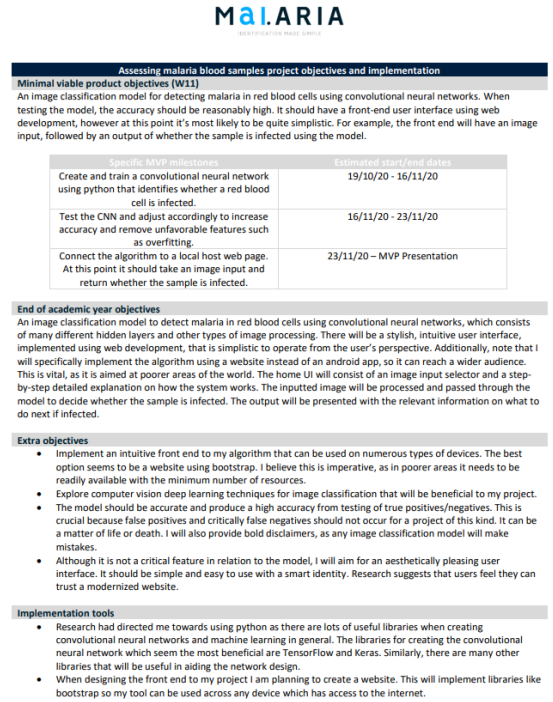
- Research had directed me towards using python as there are lots of useful libraries when creating convolutional neural networks and machine learning in general. The libraries for creating the convolutional neural network which seem the most beneficial are TensorFlow and Keras. Similarly, there are many other libraries that will be useful in aiding the network design.
- When designing the front end to my project I am planning to create a website. This will implement libraries like bootstrap so my tool can be used across any device which has access to the internet.

MAL.ARIA

IDENTIFICATION MADE SIMPLE

SUBMISSION OBJECTIVES/IMPLEMENTATION

- Already including the previous goals
- Increase accuracy so tool is reliable
- Stylish, intuitive user interface, implemented using web development
- Add brand
- Provide model disclaimers
- Guide on how to take a test
- Guide on what to do next if infected



MAL.ARIA

Assessing malaria blood samples project objectives and implementation

Minimal viable product objectives (W11)

An image classification model for detecting malaria in red blood cells using convolutional neural networks. When testing the model, the accuracy should be reasonably high. It should have a front-end user interface using web development, however at this point it's most likely to be quite simplistic. For example, the front end will have an image input, followed by an output of whether the sample is infected using the model.

Specific MVP milestones	Estimated start/end dates
Create and train a convolutional neural network using python that identifies whether a red blood cell is infected.	19/10/20 - 16/11/20
Test the CNN and adjust accordingly to increase accuracy and remove unfavorable features such as overfitting.	16/11/20 - 23/11/20
Connect the algorithm to a local host web page. At this point it should take an image input and return whether the sample is infected.	23/11/20 – MVP Presentation

End of academic year objectives

An image classification model to detect malaria in red blood cells using convolutional neural networks, which consists of many different hidden layers and other types of image processing. There will be a stylish, intuitive user interface, implemented using web development, that is simplistic to operate from the user's perspective. Additionally, note that I will specifically implement the algorithm using a website instead of an android app, so it can reach a wider audience. This is vital, as it is aimed at poorer areas of the world. The home UI will consist of an image input selector and a step-by-step detailed explanation on how the system works. The inputted image will be processed and passed through the model to decide whether the sample is infected. The output will be presented with the relevant information on what to do next if infected.

Extra objectives

- Implement an intuitive front end to my algorithm that can be used on numerous types of devices. The best option seems to be a website using bootstrap. I believe this is imperative, as in poorer areas it needs to be readily available with the minimum number of resources.
- Explore computer vision deep learning techniques for image classification that will be beneficial to my project.
- The model should be accurate and produce a high accuracy from testing of true positives/negatives. This is crucial because false positives and critically false negatives should not occur for a project of this kind. It can be a matter of life or death. I will also provide bold disclaimers, as any image classification model will make mistakes.
- Although it is not a critical feature in relation to the model, I will aim for an aesthetically pleasing user interface. It should be simple and easy to use with a smart identity. Research suggests that users feel they can trust a modernized website.

Implementation tools

- Research had directed me towards using python as there are lots of useful libraries when creating convolutional neural networks and machine learning in general. The libraries for creating the convolutional neural network which seem the most beneficial are TensorFlow and Keras. Similarly, there are many other libraries that will be useful in aiding the network design.
- When designing the front end to my project I am planning to create a website. This will implement libraries like bootstrap so my tool can be used across any device which has access to the internet.

MAL.ARIA

IDENTIFICATION MADE SIMPLE

MVP ESTIMATED DEADLINES

Specific MVP milestones	Estimated start/end dates
Create and train a convolutional neural network using python that identifies whether a red blood cell is infected.	19/10/20 - 16/11/20
Test the CNN and adjust accordingly to increase accuracy and remove unfavorable features such as overfitting.	16/11/20 - 23/11/20
Connect the algorithm to a local host web page. At this point it should take an image input and return whether the sample is infected.	23/11/20 – MVP Presentation

MAL.ARIA
IDENTIFICATION MADE SIMPLE

IMPLEMENTATION TOOLS

- Python
  - Tensorflow
  - Keras
  - Many other useful libraries that are open source
- Web development
  - Bootstrap

MAL.ARIA
IDENTIFICATION MADE SIMPLE

SUMMARY

- Researched and understood how images are classified
- Researched my problem in thorough detail
- Researched how I want to implement the user interface
- Created a detailed plan of what I want to achieve and how I want to implement it
  - Clear objectives and goals
  - Timetabled key deadlines up to MVP
  - Implementation details
  - Implementation tools
- Examined a image classifiers Google AI course on how to implement CNNs in code
- Created an example project brand to use

MaI.ARIA
IDENTIFICATION MADE SIMPLE

INITIAL NEXT STEPS

- Look at more tutorials on how to implement CNNs in code
- Implement a convolutional neural network using python libraries like TensorFlow and Keras
- Research more on CNNs while writing the code
  - Increase my knowledge
  - Improve model

MAL.ARIA
IDENTIFICATION MADE SIMPLE

THANK YOU, QUESTIONS?

MAL.ARIA
IDENTIFICATION MADE SIMPLE