Capstone Project: Malaria Detection

By: Srikant Kumar Kalaputapu

# Executive Summary

This project proposes the Convolutional Neural Network model for the detection of Malaria plasmodium parasite in images of red blood cells. This model is an automated process with high degree of accuracy and low number of false positives and false negatives. Unfortunately, massive amounts of high-quality data and exponentially increasing computational power are required to train such artificial neural networks. The “black box” nature of neural networks makes their evaluations hard to interpret – it is unknown why or how certain decisions were made. This lack of reason leads to a lack of trust in the performance of the models. It is recommended that stakeholders consider improving ease of access, computational resources via cloud computing and media coverage to physicians and lab technicians

# Problem Summary

Malaria is one of the world’s most infectious diseases. Traditional diagnosis procedures are tedious and time consuming, requiring experienced professionals. Inaccuracies caused by human error adversely impact diagnosis. An **automated system to help with early and accurate detection of malaria** can save more lives with better accuracy than manual diagnosis.

The objective of this project is to build an efficient computer vision model to detect malaria. The model will identify an image of a red blood cell and determine where it is infected with the malaria parasite or not. The red blood cell images are classified as parasitized or uninfected and split into two groups: training images and test images. The convolutional neural network model presented here allows for the evaluation of red blood cell images as parasitized or uninfected. The use of confusion matrix, classification report, F1 score, accuracy, recall and precision help measure the success of the final model.

This model will help us cut down the cost of time when it comes to diagnosis of malaria, resulting in more lives saved.

# Solution Design

Several neural network models were explored as part of the solution design, including basic convolutional neural networks (CNN), data augmentation and a pretrained VGG16 model. The final proposed solution is a Convolutional Neural Network with multiple Convolutional, Maximum Pooling, and Dropout layers using the provided red blood cell images.

**Figure 1** shows the best model’s confusion matrix and classification report. This model had very high accuracy, F1 score, recall and precision indicating a very accurate model for malaria detection. Compared to the base CNN model which had fewer Max Pooling, Convolutional and Dropout layers, the accuracy is slightly less (see Appendix 1). This model is extremely like the final model, however the final model had fewer false negatives with a similar F1 score and accuracy.

precision recall f1-score support

0 0.99 0.97 0.98 1300

1 0.97 0.99 0.98 1300

accuracy 0.98 2600

macro avg 0.98 0.98 0.98 2600

weighted avg 0.98 0.98 0.98 2600

A picture containing graphical user interface

Description automatically generated

Figure 1: Best overall model Convolutional Neural Network using red blood cell image set

False negatives are cell images evaluated to be uninfected by the model but are actually parasitized. We want these to be minimized as we want patients to get the proper treatment and survive if infected with plasm. These false negatives can lead to death as the patient will not receive treatment if the decision is made solely on the model.

**Figure 2** shows the validation accuracy vs training accuracy graph of the final model. Both accuracies follow a similar trend with validation accuracy higher than training accuracy. If the training accuracy was higher than the validation accuracy it would imply the final model is overfitting the training data but that does not seem to be the case.

A picture containing shape

Description automatically generated

Figure 2 Validation vs Training Accuracy

# Limitations and Recommendations for Further Analysis

There are 3 major limitations in using the CNN for malaria detection. Neural networks take a considerable amount of computational processing power. Each pixel of the red blood cell images is considered a node in our CNN, and with normalized sizes of pictures to 64 x 64 or 4096 pixels, the number of nodes grows extremely quickly with each hidden layer. **Figure 3** shows the number of nodes and layers of the final model.

Figure 3: Nodes in the Final CNN Model

Model: "sequential"

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Param #

=================================================================

conv2d (Conv2D) (None, 64, 64, 32) 416

max\_pooling2d (MaxPooling2D (None, 32, 32, 32) 0

)

dropout (Dropout) (None, 32, 32, 32) 0

conv2d\_1 (Conv2D) (None, 32, 32, 32) 4128

max\_pooling2d\_1 (MaxPooling (None, 16, 16, 32) 0

2D)

dropout\_1 (Dropout) (None, 16, 16, 32) 0

conv2d\_2 (Conv2D) (None, 16, 16, 32) 4128

max\_pooling2d\_2 (MaxPooling (None, 8, 8, 32) 0

2D)

dropout\_2 (Dropout) (None, 8, 8, 32) 0

conv2d\_3 (Conv2D) (None, 8, 8, 32) 4128

max\_pooling2d\_3 (MaxPooling (None, 4, 4, 32) 0

2D)

dropout\_3 (Dropout) (None, 4, 4, 32) 0

flatten (Flatten) (None, 512) 0

dense (Dense) (None, 512) 262656

dropout\_4 (Dropout) (None, 512) 0

dense\_1 (Dense) (None, 2) 1026

=================================================================

Total params: 276,482

Trainable params: 276,482

Non-trainable params: 0

276,482 total parameters must be seen through with both forward and backward propagation to train this model. Considering there are only 24,958 cell images in the training data set, and training this set took around an hour using the extra computational power of Google Colab, applying this model to a larger data set will take a great deal of time and power.

Neural Networks have a “black box” nature. There is no context given to why or how the evaluations are made by the neural network during training and evaluation. This can result in a lack of trust by users. Adversarial and out of distribution images can fool neural networks into incorrectly identifying and evaluating the presence of classifications. In this experiment, uninfected cells may contain other impurities that contribute noise to the overall neural network, thereby resulting in incorrect evaluations.

Neural Networks have seen peaks and valleys in terms of marketing and exposure. Neural Networks have been around since the 1940s but its only due to advancements in algorithms (such as the concept of backpropagation), ease of access to better computational power (cloud computing) and the abundance of large data sets (Big Data) that has allowed it to take off over the last decade. More education and ease of access is required to bridge the gap between data scientists and end users, i.e., patients, physicians and technicians.

# Recommendations for Policy

The proposed final model demonstrates a high efficiency of neural networks for evaluation of the presence of plasmodium parasite in red blood cell images. I would recommend applying this model or other similar algorithms to more data sets for better accuracy.

I would recommend the trained model should be made easily available via cloud solutions (AWS etc.) so that technicians and physicians can access the results even in remote locations. For ease of use I recommend a user-friendly application to interact with said neural networks to reduce the need to train technicians and physicians. One such example that has seen some research is a smartphone application to acquire microscopic blood smear images for malaria detection neural networks (“Deep Learning Based Automatic Malaria Parasite Detection from Blood Smear and its Smartphone Based Application”)

Finally educating technicians and physicians on the benefits of neural networks while incorporating human feedback in future models (to reduce the effects of adversarial and out of distribution images by having a second human opinion) will help with media coverage as well as encourage usage.

Bibliography

* Kim, I., 2022. *Demystifying Batch Normalization vs Drop out*. [online] Medium. Available at: <https://medium.com/mlearning-ai/demystifying-batch-normalization-vs-drop-out-1c8310d9b516> [Accessed 10 June 2022].
* Fuhad, K M Faizullah et al. “Deep Learning Based Automatic Malaria Parasite Detection from Blood Smear and its Smartphone Based Application.” *Diagnostics (Basel, Switzerland)* vol. 10,5 329. 20 May. 2020, doi:10.3390/diagnostics10050329
* Asquero. 2022. *Advantages and Disadvantages of Artificial Neural Networks | Asquero*. [online] Available at: <https://www.asquero.com/article/advantages-and-disadvantages-of-artificial-neural-networks/> [Accessed 10 June 2022].
* Mishra, P., 2022. *Why are Convolutional Neural Networks good for image classification?*. [online] Medium. Available at: <https://medium.datadriveninvestor.com/why-are-convolutional-neural-networks-good-for-image-classification-146ec6e865e8> [Accessed 10 June 2022].
* Hutson, M., 2022. *A turtle—or a rifle? Hackers easily fool AIs into seeing the wrong thing*. [online] Science.org. Available at: <https://www.science.org/content/article/turtle-or-rifle-hackers-easily-fool-ais-seeing-wrong-thing> [Accessed 10 June 2022].
* Vincent, J., 2022. *Magic AI: these are the optical illusions that trick, fool, and flummox computers*. [online] The Verge. Available at: <https://www.theverge.com/2017/4/12/15271874/ai-adversarial-images-fooling-attacks-artificial-intelligence> [Accessed 10 June 2022].
* Marin Dujmović, Gaurav Malhotra, Jeffrey S Bowers (2020) What do adversarial images tell us about human vision? eLife 9:e55978 <https://doi.org/10.7554/eLife.55978>
* Uwimana, A. and Senanayake, R., 2022. *Out of Distribution Detection and Adversarial Attacks on Deep Neural Networks for Robust Medical Image Analysis*. [online] OpenReview. Available at: <https://openreview.net/forum?id=1iy7rdPCt\_> [Accessed 10 June 2022].
* Rumelhart, D., Hinton, G. & Williams, R. Learning representations by back-propagating errors. *Nature* **323,**533–536 (1986). <https://doi.org/10.1038/323533a0>
* <https://www.quora.com/What-are-the-advantages-of-using-Leaky-Rectified-Linear-Units-Leaky-ReLU-over-normal-ReLU-in-deep-learning>
* <https://www.quora.com/What-is-the-difference-between-dropout-and-batch-normalization>

Appendix

Appendix 1 Base CNN with fewer hidden layers than final model

A screenshot of a computer

Description automatically generated with low confidence

precision recall f1-score support

0 0.99 0.98 0.98 1300

1 0.98 0.99 0.98 1300

accuracy 0.98 2600

macro avg 0.98 0.98 0.98 2600

weighted avg 0.98 0.98 0.98 2600