Capstone Proposal

Carlos Mertens

March 23rd, 2019

Malaria Cells Classifier

Domain Background

The Health-care field of research is one of the fastest growing domain for Machine Learning and Artificial Intelligence owing to the vast collection of data in the sector. Only in the US, the Health-care system generates approximately one trillion gigabytes of data a year[1]. According to Panner[2], data is the key to the future medical revolution. He states that, the most data we collect and learn from it, the closer we could come to discover new treatments and cures.

We could also use Machine Learning to prevent or to detect early stages of diseases. For example, researchers from Stanford University have successfully created a Machine Learning Classifier to detect cancer skin using data images. Using Deep Convolutional Neural Network, they have reached the accuracy of certified dermatologist[3]. By taking advantage of the knowledge and technique used by the researches from Stanford University, we could also create a classifier that would detect cells infected with Malaria in their early stage.

According to the World Health Organization[4], there were 435000 deaths around the world in 2017 and more than 80% of all deaths occurred in developing nations. A Malaria Cell Classifier could potentially help doctors from these developing nations by instantly detecting Malaria in the blood cells tested from patients. Therefore by early detection, many affected people can start having the proper treatment avoiding death or possible organ damages due to the infection.

By growing up in the Amazons area of Bolivia, I was constantly exposed to the bites of the mosquito infected with Malaria. Although I have not experience any death cause by Malaria, I have many friends that were treated late and they have been left with liver or kidneys complications.

Problem Statement

The Malaria could take from 7 to 18 days to develop and to start showing the symptoms and even in some cases it has taken a year. The initial symptoms could easily be confused by a flu and therefore it could be difficult for doctors to identify it as a Malaria without a blood test and lab tests. The disease is not fatal instantly, instead most complications are due to the lack of proper treatment on time. For example, in my experience with friends having the disease, the blood test had to be sent to another city in order to be tested in the lab for Malaria. By the time the Malaria tests came back from the other city, my friends were exposed to the disease for too long.

A Machine Learning Malaria Cell Classifier would be very helpful in developing countries in order to diagnose the patient fast enough to avoid further organs complications and/or death. Our problem at hand is quantifiable, it means that images of blood cells (data) could be expressed or classified if the malaria parasite is detected or not. By having and collecting a decent amount of images of cell infected or not, the problem could be easily measured by percentage and accuracy. And of course, the problem is replicable because is an ongoing global disease

Datasets and Inputs

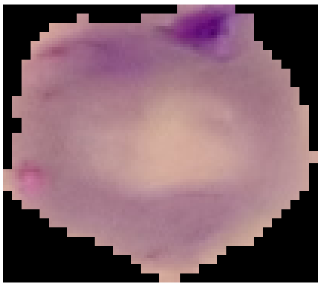
The dataset acquired for this project contains 27558 images of cells infected with the malaria parasite and cells that are not infected. They are divided equally in two folders named Parasitized and Uninfected and the data is ready to be used to train our classifier.

The data was originally obtained from patients that were diagnosed with malaria and patients that were not diagnosed in Bangladesh[5]. The data is perfectly adequate for our problem since we are intending to create a image classifier that can detect the patterns in cells infected with the malaria parasite.

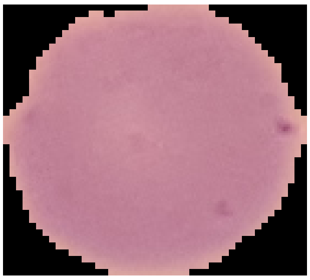
We have collected the data from the US National Library of Medicine (NIH) but it could also be obtain from Kaggle[6] collection of datasets. The images were capture by a built-in camera of an Android phone attached to a conventional light microscope and carefully annotated by experts[5].

As an input, we will require the user end to provide the path of the image that they would like to pass though our classifier in order to detect if the image contains an infected cell or not.

Cell with Malaria



Cell without Malaria



Solution Statement

We would create an application to classify the input image of a blood cell if it is infected with the malaria parasite or if it is not. The application would be able to measure the accuracy of the blood cell tested in order to help a doctor to make a decision on applying the proper treatment or not. Therefore we would be helping hospitals and care centers in developing countries in avoiding the delay in the decision to start a treatment for malaria in patients.

Furthermore, the malaria parasite is adaptable to the environment[7], therefore we would built an application that would be portable and the training can be improved or retrained with additional data in order to classify different parasites in the future.

Benchmark Model

Initially, we thought about using as a benchmark the project from the Stanford University on the skin cancer classifier. They have obtained with their classifier an overall accuracy of 72% ± while professional dermatologists had an accuracy of 65% ± by looking at the same infected blood cell images. However, we have to consider that the project from Stanford University is classifying several type of skin cancer and not just one type of classification that we are going to be needing.

Nevertheless, we have found some Kaggke kernels- “a cloud computational environment that enables reproducible and collaborative analysis” [8] – that can reach as high as 97% accuracy on their models. We will be choosing as our benchmark a simple but well structured kernel create by Rohit Pawar[9] where he had reach an accuracy of around 95%.

Evaluation Metrics

Our benchmark project used only accuracy as a evaluation test which for a simple application it could be consider a reliable evaluation and validation metric. However, we would like to have a more complex evaluation metrics and one that it is more focus on our problem intended to solve. We will be using F-beta score in parallel with accuracy.

Using the concept of Confusion Matrix where we store four values (True Positives, True Negatives, False Positives, False Negatives), we could say that we have a problem of a Type Error 1 (Error of the first kind, or False Positive) which In the medical terms, this would be when we misdiagnose a healthy patient as sick. It means that for our problem, it will be okay to classify a healthy cell as infected but not okay to classify an infected cell as healthy.

Confusion Matrix

	Classified Infected	**Classified Healthy**
Infected Cell	True Positives (Correctly Classified)	False Negatives (Type Error2)
Healthy Cell	False Negatives (Type Error 1)	True Negatives (Correctly Classified)

Therefore, we will be using F-beta score and we will be setting the value of beta (β) closer to 1 in order to get a high recall value which in medical terms would be high sensitivity value. Furthermore, we will also keep track of the accuracy in order to keep up with our benchmark model.

F-beta Score Formula

$$F_{\beta} = (1 + \beta^{2}) * \frac{Precision * Recall}{\beta * Precision + Recall}$$

Project Design

We would build an image classifier that would recognize patterns in an image of a blood cell and classify it if it is infected with malaria parasites or not. In general, we would like to create an application that can be exported in the future in order to be used as a smart phone app or a website app. When the project is completed, we would have a Neural Network application that would be able to be trained with any set of labeled images and be used out of the box as a command line application.

In order to accomplish our goal and to create an application that can be easily updated, we have thought about breaking down the project in three parts with each part having different modules in order to make easier for future updates and implementations. Most of our work would be executed on a Anaconda Notebook in order to make it easier for us on the testing and training process. However, the application would consist of several Python files containing all our modulated codes with two of them being the main files: *train.py* and *predict.py*. The architecture of the Neural Network would have its own Python file coded as an object oriented program.

1. Prepare Datasets

In this part of the project, we would load the data. Although, the dataset we have is ready to be used, we would have to resize the images because they are not all the same size and then we would have to convert them into tensors or array of numbers to be used in the Neural Network. We would separate the data in training, validation and testing subsets which is a common practice in Machine Learning. After that we would have to create the labels and the classes we would be using which are infected or healthy.

2. Train a Neural Network Classifier

Here we would create a model architecture first. We would be using Keras which is a library written in Python and it is highly used in this days due to its user-friendly and modular approach. In this stretch, we would be using the training subset to feet the model for training and the validation subset to validate the training. We also would be tracking the accuracy and the F-beta score evaluation metrics we have proposed. After that, we would be testing the model with our testing subset and track the accuracy and the F-beta score again. Finally, when we would be satisfied, we would be saving the model in checkpoints to be exported a later times.

3. Predict an image with the trained classifier

In this session, we would first require a path of a blood cell image and then load it. We would apply the same resizing and converting into a tensor that we would used previously. Then we would have to load the model saved on the checkpoints in order to pass the image from the user. We would be printing the the accuracy of the image to be classified as infected with malaria.

We would also be testing some other techniques such as one-hot encoding in the data labels, transfer learning pre-trained models and some transformation and regularization on the data. Although we may not included on the final project, but as we mentioned before we would like to keep an open mind that in the future the application can be extended to test other variation of parasites and be trained on more modern images.

Reference

1. David Champagne, Sastry Chilukuri, Martha Imprialou, Saif Rathore, and Jordan VanLare (2018), “Machine Learning and Therapeutics 2.0: Avoiding Hype, Realizing Potential”. *McKinsey & Company* [Online], Available at: https://www.mckinsey.com/industries/pharmaceuticals-and-medical-products/our-insights/machine-learning-and-therapeutics-2-0-avoiding-hype-realizing-potential [Accessed on 1 April, 2019]

2. Morris Panner (2018), “Health Data Meets Artificial Intelligence And Machine Learning”. Forbes [Online], Available at: https://www.forbes.com/sites/forbestechcouncil/2018/11/21/health-data-meets-artificial-intelligence-and-machine-learning/ [Accessed on 1 April 2019]

3. Andre Esteva, Brett Kuprel, Roberto A. Novoa, Justin Ko, Susan M. Swetter, Helen M. Blau, Sebastian Thrun, “Skin Cancer Classification with Deep Learning”. nature [Online], Available at: https://cs.stanford.edu/people/esteva/nature/ [Accessed on 2 April 2019]

4. World Health Organization [Online] “Global Health Observatory (GHO) Data”, Available at: https://www.who.int/gho/malaria/epidemic/deaths/en/ [Accessed on 2 April 2019]

5. U.S. National Library of Medicine [Online], Available at: https://ceb.nlm.nih.gov/repositories/malaria-datasets/ [Accessed on 3 April 2019]

6. User: Arunava (2019), “Malaria Cell Images Dataset”. Kaggle [Online], Available at: https://www.kaggle.com/iarunava/cell-images-for-detecting-malaria [Accessed on 3 April 2019]

7. Martin Rono (2017), “How the malaria parasites adapts when faced with different opportunities for transmission in its natural enviroment”. Kemri – Welcome Trust [Online], Available at: http://kemri-wellcome.org/blog-post/how-the-malaria-parasites-adapts-when-faced-with-different-opportunities-for-transmission-in-its-natural-environment/ [Accessed 3 April 2019]

8. Documentation, “How to Use Kaggle”. Kaggle [Online], Available at: https://www.kaggle.com/docs/kernels [Accessed 4 April 2019]

9. Rohit Pawar, “Malaria Cells Classification Through Keras”. Kaggle [Online], Available at: https://www.kaggle.com/sharp1/malaria-cells-classification-through-keras [Accessed on 4 April 2019]

Additional Reference

http://web.orionhealth.com/rs/981-HEV-035/images/Introduction_to_Machine_Learning.pdf

https://towardsdatascience.com/a-review-of-recent-reinforcment-learning-applications-to-healthcare-1f8357600407

https://www.mayoclinic.org/diseases-conditions/malaria/symptoms-causes/syc-20351184

https://www.nhs.uk/conditions/malaria/symptoms/