Malaria diagnoses - Capstone project

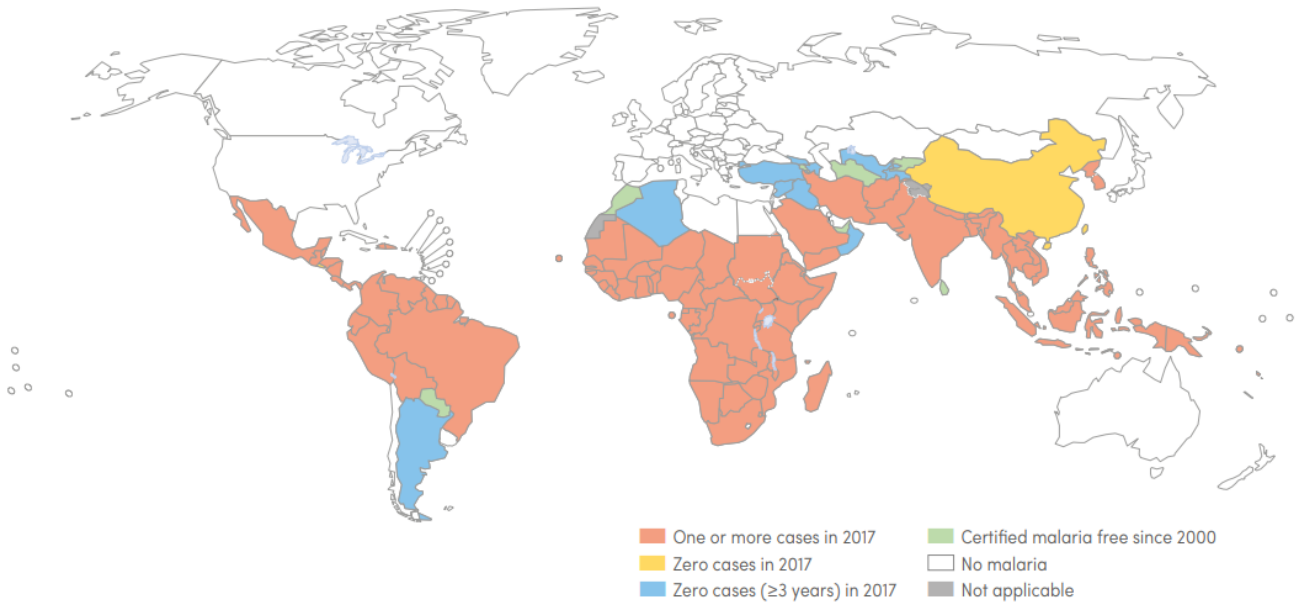
Paulo Henrique Zen Messerschmidt

The machine learning models have been applied in several areas, such as financial market, real estate market and fraud detection and also used to solve problems of the most varied fields. Among these areas, machine learning models plays an important role in the health care industry, assisting physicians in the early diagnosis of various types of diseases, such as heart disease, locomotor disorders, among others (Groves et al., 2013). An example application can be seen in the research of Esteva et al (2017), where a model for dermatological classification of skin cancer with deep neural networks was developed.

Among the many diseases, malaria is a life-threatening disease caused by the parasite *Plasmodium*. Female Anopheles mosquitoes pick up the parasite from infected people when they bite to obtain blood needed to nurture their eggs. Inside the mosquito the parasites reproduce and develop. When the mosquito bites again, the parasites contained in the salivary gland are injected and pass into the blood of the person being bitten. (World Health Organization, 2019). According to World Malaria Report in 2017 there were 219 million cases of malaria compared to 217 million cases in 2016. The estimated number of malaria deaths was 435,000 in 2017, a similar number as in the previous year.

In addition, by 2017 the African region contained 92% of malaria cases and 93% of malaria deaths. In the same year, 5 countries accounted for nearly half of all malaria cases in the world: Nigeria (25%), Democratic Republic of Congo (11%), Mozambique (5%), India (4%) and Uganda %) (World Malaria Report, 2018). Figure 1 shows the status of the disease in each country of the world in the year 2017. As can be seen, practically every African and Asian continent presented one or more cases in 2017 in addition to part of the Latin countries, such Brazil and Mexico. While the disease is uncommon in temperate climates, malaria is still common in tropical and subtropical countries.

Figura 1 - Malaria status by country in 2017.



World Malaria Report, 2018.

It can be said that, in short, malaria affects mostly underdeveloped countries, which in addition, generally have more precarious public health systems. Although the disease is curable, this precariousness of basic health care can often hamper diagnosis and even access to treatment.

Besides that Malaria is an acute febrile illness. In a non-immune individual, symptoms usually appear 10–15 days after the infective mosquito bite. The first symptoms (fever, headache and chills) may be mild and difficult to recognize as malaria because they are common to many types of diseases. If not treated within 24 hours, *P. falciparum* malaria can progress to severe illness, often leading to death. (World Healt Organization, 2019).

Problem Statement

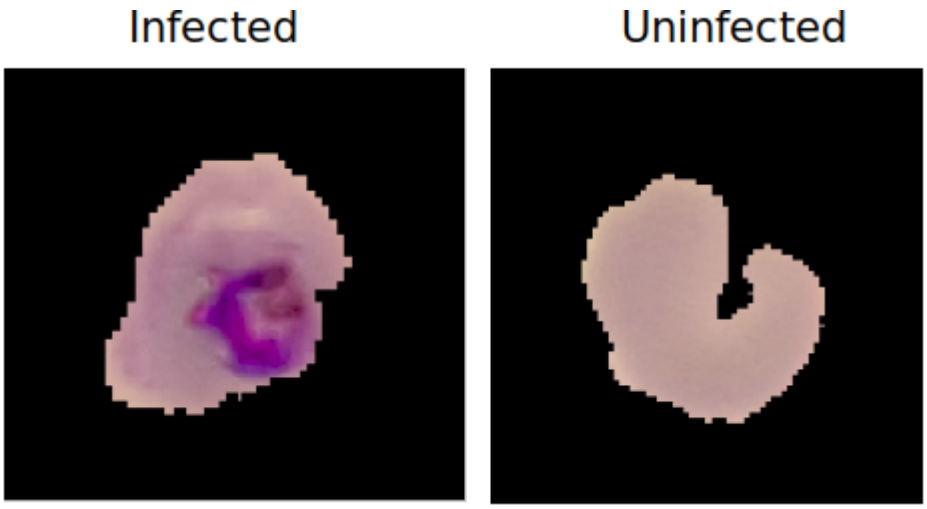
Based on this information, it is exposed the motivation for the development of this work, which is summarized in ***how to create a machine learning model to detect the presence of malaria based on cell images***. This model aims to aid in the early diagnosis of the presence of malaria, allowing the patient to have medical treatment in anticipation. In general terms, the purpose of this work is to contribute to save lives.

Datasets e inputs

The data used in this project was obtained from kaggle, specifically in the Malaria cell Images dataset.

This dataset is originally provided by the U.S. National Library of Medicine (NIH), and can be found in this link. This dataset has a total of 27,558 images labeled "Infected" and "Not Infected", as shown in Figure 2.

Figure 2 - Example of an infected cell and an uninfected cell.



Solution Statement

In order to solve the proposed problem, a deep neural network model, more specifically Convolutional Neural Networks (CNNs), will be implemented, which is composed of convolutional layers and pooling layers. For the construction of the network, two approaches will be considered:

- First approach: create a CNNs from scratch, which will be composed of a certain amount of convolutional layers interspersed by pooling layers. The final layer will consist of a densely connected 2-node layer whose nodes will have a softmax activation function to return the probability of each label (infected and uninfected).
- Second approach: create a model using the **transfer learning** technique. This technique consists of adapting neural networks already consolidated and trained in ImageNet database and adapt to the problem that we are trying to solve, in this case the prediction of the presence of malaria.

As presented in udacity's lessons on transfer learning, there are basically four different approaches to implement trasnfer learning:

- Small dataset, similar data;
- Small dataset, different data;
- Large dataset, similar data;
- Large dataset, different data;

Remembering that the algorithms were pre-trained using the ImageNet, which has about 150,000 images labeled with 1000 different categories, including animals and everyday objects. In this case, we will consider the second approach (small Dataset, different data), since we have a set of only 27,558 images, which present cell images. In a first moment, the algorithm used will be the VGG-16, but others algorithms already available in Keras can be tested in for performance comparison. The implementation steps based on the chosen approach will be:

- Cut the final layers of the network, keeping only the first layers, responsible for identifying common patterns, such as borders, common shapes such as stripes, circles, rectangles.
- Add to the pre-trained initial layers densely connected layers corresponding to the number of classes of the data set under analysis, in this case "Infected" and "Not infected".
- Freeze the weights of the pre-trained network and train only the part of the network added.

The last step will be to compare the performance of the network model created from scratch and the model using transfer learning. The transfer learning model is expected to perform better. The implementation steps are detailed in the ***Project Design section.***

Benchmark model

The benchmark model used for comparison is presented in Ross et al (2006). In this model, the automated image processing algorithm is designed to diagnose malaria in the same way as a human operator performing microscopy. To do this, the algorithm finds and identifies erythrocytes and malaria parasites present in a microscopic field of blood (thin layer of blood). Based on the parasites and erythrocytes found, the program makes a diagnosis about whether or not malaria is present. Using this algorithm, the authors obtained an accuracy of 73%, recall score of 85% and precision score of 81%.

Evaluation metrics

Given the context of this problem and the adopted benchmark model, the following metrics will be considered:

- Accuracy:

$$Accuracy = \frac{TruePositives + TrueNegatives}{Total} \tag{1}$$

- Precision:

$$Precision = \frac{TruePositives}{TruePositive + FalsePositive} \tag{2}$$

- Recall:

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative} \tag{3}$$

- F-Beta Score (In this case, Beta=1):

$$F1Score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{4}$$

Also, the ROC (**R**eceiver **O**peration **C**haracteristic), which basically returns the **A**rea **U**nder the **C**urve (AOC), will be used as a metric, which is plotted in a Cartesian plane where:

- The x-axis is represented by the sensitivity (equivalent to recall score):

$$Sensitivity = \frac{TruePositive}{TruePositive + FalseNegative} \tag{5}$$

- The y-axis is represented by the specificity:

$$Specificity = \frac{TrueNegative}{TrueNegative + FalsePositive} \tag{6}$$

Project Design

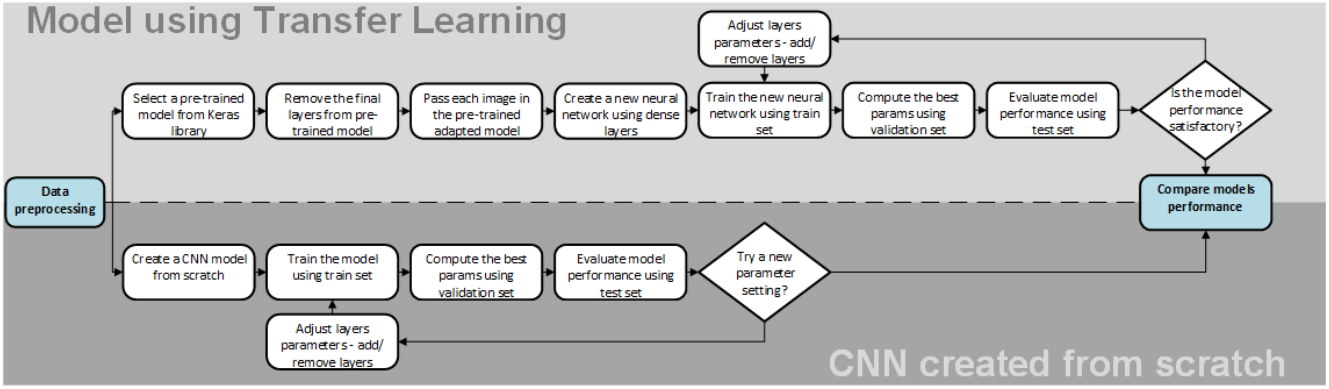
The project workflow is planned according to the flowchart shown in Figure 3. As can be seen, there are basically two sets of activities from the data preprocessing: the construction of the model using knowledge transfer; and building the CNN model from scratch.

The process starts with data preprocessing. Initially, the data will be divided into data training, validation and testing. In this step, a 4D matrix is created to serve as input to CNNs. This 4D matrix has the form (nb_samples, rows, columns, channels):

- nb_samples: total number of images;
- rows: number of rows in each image;
- columns: number of columns in each image;
- channels: number of channels of each image (RGB).

Therefore, in this stage of pre-processing each image is resize to a standard size, say (230x230 pixels). Then, each image will be converted to an array, which is then resized to a 4D tensor.

Figura 3 - Workflow



Feito isso, dividimos o workflow em duas partes.

Model Using Transfer learning

To create this model, a pre-trained network will be chosen, in this case VGG-16. Based on the approach chosen in section **Proposed Solution** the final layers of the VGG-16 model will be removed. Afterwards, each image will be passed through the reduced network in order to obtain the image features (***bottleneck features***). In the next step a new neural network will be created using dense layers fully connected, which corresponds to the final layers of the VGG-16 model. Once the network is created, it will be trained using a CheckPointer to compute the best parameters obtained from the validation data score. Finally, the model will be evaluated using the test data. If the score is satisfactory, the result will be compared with the CNN created from scratch, if not, a new adjustment of network layer parameters (add/ remove layers or remove/add nodes) should be done.

CNN created from scratch

First, the CNN network will be created from scratch, with convolutional layers, max pooling layers and dense layers. The rest of the process follows the same traditional training process, obtaining the best parameters with the validation data and performance evaluation with the test data. Once the score is obtained in the test data, the performance of this network is compared to the network using transfer learning. Both models will be compared with the benchmark model.