MALARIA CELL IMAGE DETECTION

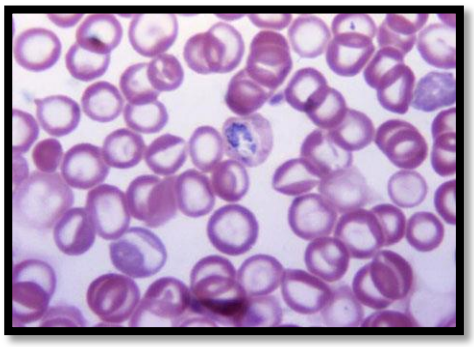
Domain Background

Malaria is a life-threatening disease caused by parasites that are transmitted to people through the bites of infected female Anopheles mosquitoes. It is preventable and curable.

- In 2017, there were an estimated 219 million cases of malaria in 90 countries.
- Malaria deaths reached 435 000 in 2017.
- The WHO African Region carries a disproportionately high share of the global malaria burden. In 2017, the region was home to 92% of malaria cases and 93% of malaria deaths.

Malaria is caused by Plasmodium parasites. The parasites are spread to people through the bites of infected female Anopheles mosquitoes, called "malaria vectors." There are 5 parasite species that cause malaria in humans, and 2 of these species – P. falciparum and P. vivax – pose the greatest threat.

Diagnosis of malaria can be difficult



- Where malaria is not endemic any more (such as in the United States), health-care providers may not be familiar with the disease. Clinicians seeing a malaria patient may forget to consider malaria among the potential diagnoses and not order the needed diagnostic tests. Laboratories may lack experience with malaria and fail to detect parasites when examining blood smears under the microscope.
- 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. If not treated within 24 hours, P. falciparum malaria can progress to severe illness, often leading to death.

Microscopic Diagnosis

Malaria parasites can be identified by examining under the microscope a drop of the patient's blood, spread out as a "blood smear" on a microscope slide. Prior to examination, the specimen is stained to give the parasites a distinctive appearance. This technique remains the gold standard for laboratory confirmation of malaria. However, it depends on the quality of the reagents, of the microscope, and on the experience of the laboratories.

Identifying the Malaria detection by using computer Vision architecture is not much accuracy while finding the malaria cells in the human body. So to overcome this problem then Deep learning came into the picture by using it we can identify the uninfected image cell.

Convolutional neural networks have the ability to automatically extract features and learn filters. In previous machine learning solutions, features had to be *manually* programmed in — for example, size, color, the morphology of the cells. Utilizing Convolutional neural networks (CNN) will greatly speed up prediction time while mirroring (or even exceeding) the accuracy of clinicians.

Reference Link:

https://towardsdatascience.com/detecting-malaria-using-deep-learning-fd4fdcee1f5a

Inspiration

Save humans by detecting and deploying Image Cells that contain Malaria or not!

Reference Links:

1. https://www.cdc.gov/malaria/diagnosis_treatment/diagnosis.html
2. https://www.who.int/news-room/fact-sheets/detail/malaria

Personal motivation

- In this project I can learn how to handle the image data and pre-processes datasets for the training the data.
- In this project I am going to learn how to use the kaggle kernals.
- In this project I am going to learn how to detect cell image data by using PyTorch.

Problem Statement

The Aim of this project is to Detect the Malaria by using the Cell Image Dataset. In this project I am going to use PyTorch by employing with the network called Model ResNet 50 (Transfer Learning) for improving the Accuracy for detecting the cell Image.

Datasets and Inputs

Dataset was Downloaded form

https://www.kaggle.com/iarunava/cell-images-for-detecting-malaria

Content

The dataset contains 2 folders:

1. Infected
2. Uninfected

And a total of **27,558** image

- **Dimension 148 x 148**
- PyTorch allow us to train on many variations of the original images that are cropped in different ways or rotated in different ways.
- I want to set up the data transformations for each set of data. In general, we want to have the same types of transformations on the validation and test sets of data. However, with the training data, we can create a more robust model by training it on rotated, flipped, and cropped images.

Reference Link:

https://pytorch.org/docs/stable/torchvision/transforms.html#torchvision.transforms.Compose

- This data is form the Official NIH Website: https://ceb.nlm.nih.gov/repositories/malaria-datasets/. The data is open – sourced and can be downloading for education purpose with no citation.

Solution

By using the PyTorch and transfer learning for the model training up to 20 to 25 epochs and also using Gradient descent optimizer. And finally I am going to identify the cell image.

Benchmark Model

- I plan to compare the results of the PyTorch versus the output of the Keras model I will compare the accuracy of each other to see which is more effective, as well as compare the speed of the two techniques.
- So for this problem, the benchmark model will be Keras model. I will try to beat it performance with other algorithms.

Evaluation Metrics

I've been exploring the PyTorch neural network library. When using any library to perform classification, at some point you want to know the classification accuracy. The CNTK and Keras libraries have built-in accuracy functions, but PyTorch (and TensorFlow) do not.

I set out to determine how to compute classification accuracy in PyTorch. Cutting to the chase, the very large number of details that had to be dealt with was really, really surprising to me.

PROJECT DESIGN:

STEP 1: TRANSFORMING THE DATA AND SPLITING THE DATA

- Transforming my dataset into 3 categories called "Training", "testing", "validation" using torchvision.transforms.Compose. Here itself image will be resized and normalizing had been made.
- Now with this Train data First I need to normalize the data by using torch.FloatTensor, then training indices will be obtained and that going to be used for validation.
- Combine the dataset and sampler to form data loaders.

STEP 2: APPLYING TRANSFER LEARNING (ResNet50) FOR BETTER ACCURACY

- Using these pre-trained models is known as transfer learning. PyTorch makes it easy to load pre-trained models and build upon them.
- Some of the most popular pre-trained models include **VGGNet, ResNet,** and **AlexNet**, all of which are pre-trained models from the ImageNet Challenge. These pre-trained models allow others to quickly obtain cutting edge results in computer vision without needing the large amounts of compute power, time, and patience in finding the right training technique to optimize the weights.
- I decided to use the ResNet architecture, which we can obtain from the **torchvision**

STEP 3: MODEL TRAINING AND LOADING SAVED MODEL

- Now that we have our model all set up, we will want to train the final layers. We also want to get an idea of how well it is working!
- Load the saved model.

STEP 4: MODEL TESTING

Now that I had a model that I thought was pretty good, I wanted to test out how well it would work on the test data. I simplified the function from the PyTorch documentation, so that I could quickly obtain the accuracy for each test batch.

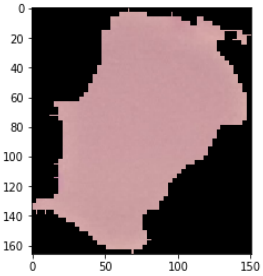
STEP 5: PROCESSING IMAGES AND ALGORITHM FOR IDENTIFYING

I am someone who needs to understand why we are doing something, so this part was a bit frustrating at first for me. The idea for this part of the project is that you want to be able to pass an individual image to your deep learning network, and for your network to predict the label for the image.

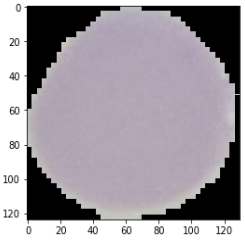
PyTorch made this easy to do for the many images we had within our folder structure. However, for a single image, it would be ideal to pass a single path without the whole folder structure set up.

Finally identifying the Image cell corresponds with class either Parataised or Uninefected.

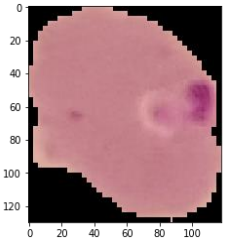
Sample Images



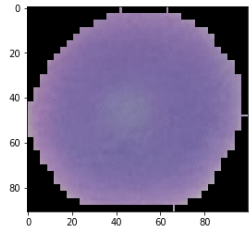
Uninfected



Uninfected



Parasitized



Uninfected

RAJA RANJITH KUMAR ASILETI,

MACHINE LEARNING NANO DEGREE,

CAPSTONE PROJECT PROPOSAL,

MALARIA CELL IMAGE DETECTION,

Sunday, February 10, 2019.