

Detecting Plasmodium Parasites in Cell Images using CNN



Name: Yash Dugar
Andrew Id: ydugar



CONTENTS

| | |
|----------------|-----------------------|
| 1. Section I | Abstract |
| 2. Section II | Introduction |
| 3. Section III | Exploring the Dataset |
| 4. Section IV | Background |
| 5. Section V | Model (ResNet50) |
| 6. Section VI | Results & Conclusion |
| 7. Section VII | References |

Section I: Abstract

In recent years, Deep Learning has emerged as a popular branch of Machine Learning, revolutionizing the way we interact with technology. With the ability to carry out complex tasks such as image recognition, speech recognition, biometric authentication and more, Deep Learning has garnered attention from a multitude of industries. Of late, the medical industry has been at the forefront of utilizing technological advancements to enhance patient care. The incorporation of Artificial Intelligence and Deep Learning has led to significant progress in medical research, paving the way for improved healthcare services for individuals worldwide. PyTorch, a powerful open-source Machine Learning framework, has played a pivotal role in enabling data scientists to explore complex solutions related to computer vision and natural language processing. PyTorch offers an intuitive approach to building neural networks, including the versatile Convolutional Neural Networks (CNN), which allows data scientists to develop intricate and sophisticated models with ease. With the ability to seamlessly implement neural networks, PyTorch offers a dynamic infrastructure for developers to explore novel concepts and apply them to real-world problems, fueling the ongoing evolution of Deep Learning and its integration into a broad spectrum of industries.

Section II: Introduction

Malaria, caused by protozoan parasites of the genus *Plasmodium*, poses a significant threat to both human life and the healthcare system worldwide. Traditional diagnostic methods like staining thin and thick peripheral blood smears and rapid test methods like OptiMAL are expensive, time-consuming, and labor-intensive, limiting their efficiency.

In recent years, researchers have developed automated screening methods using machine learning techniques to detect malaria based on the photo of blood cells. CNN-based models have shown superior performance in malaria detection and can be easily trained using a pre-existing model.

For this homework, I am required to implement a convolution neural network-based machine learning model to predict whether a cell in the given picture is parasitized by the genus *Plasmodium* or not. All images in the training set are labeled with either 1 (parasitized) or 0 (uninfected). I have the option to build a model from scratch or use a pre-trained model with modifications. Once I have trained my model, I will use it to make predictions on the test set and submit my results.

Section III: Exploring the Dataset

The dataset under consideration comprises train and test images along with hidden_test images, the latter intended for making predictions. It is noteworthy that each image portrays only one cell, thereby simplifying the classification task. The training set is composed of 22,067 images, which have been duly labeled as either 1 or 0, corresponding to parasitized and uninfected cells, respectively. On similar lines, the testing set entails 2,757 images, while the hidden_test consists of an equivalent number of images. We present below example images from the training, test, and hidden_test sets to aid in comprehending the dataset.



Figure I: From Train Dataset



Figure II: From Test Dataset



*Figure III: From
Hidden_Test Dataset*

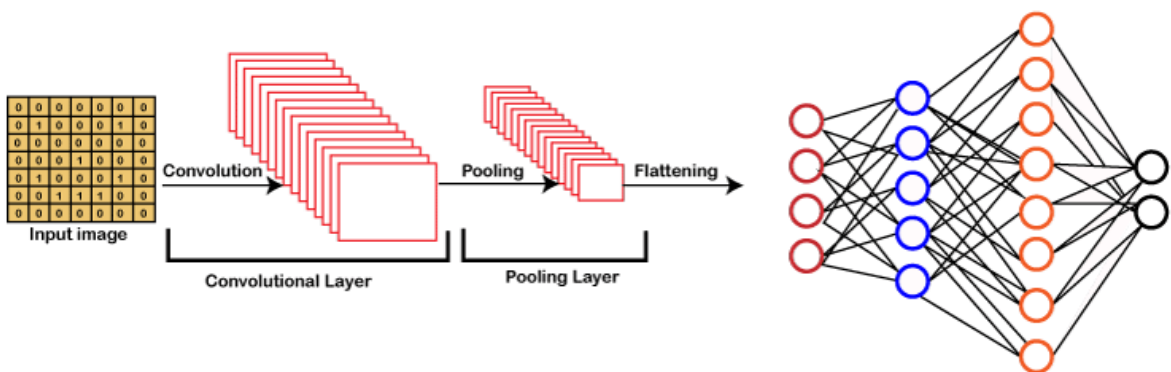
As observed in above figures, the images exhibit bright purple circles that may potentially serve as indicators of the genus *Plasmodium*, responsible for the onset of malaria. The aforementioned serves as a key insight and establishes the foundation for implementing convolutional neural network-based machine learning models for accurate prediction and diagnosis of malaria.

Section IV: Background

Convolutional Neural Networks (CNNs) have become the state-of-the-art approach for image categorization tasks due to their ability to learn hierarchical representations of input images through convolutional layers. These layers apply a set of filters to the input image, generating a set of feature maps that capture various aspects of the image, such as edges, forms, and textures. The pooling layers in CNNs help to reduce the spatial dimensions of the feature maps, which not only reduces the computational expenses but also helps to reduce overfitting by limiting the number of parameters in the model. Batch normalization, another commonly used technique, normalizes the inputs to each layer, improving the generalization of the model and increasing training stability by reducing the internal covariate shift.

Data augmentation techniques such as random rotations, translations, flips, and adjustments to brightness and contrast are often used to increase the size of the training dataset. This enhances the generalization of the model by introducing more variation in the data and helps to avoid overfitting by reducing the model's sensitivity to the training data.

For the malaria classification problem, ResNet50 model was used. ResNet50 uses residual learning to resolve the vanishing gradient problem that can occur in deep networks, making deep network training easier. It is 50 layers deep, that's why the name ResNet50. We also utilized different augmentation techniques like normalization, resizing, center crop, etc.



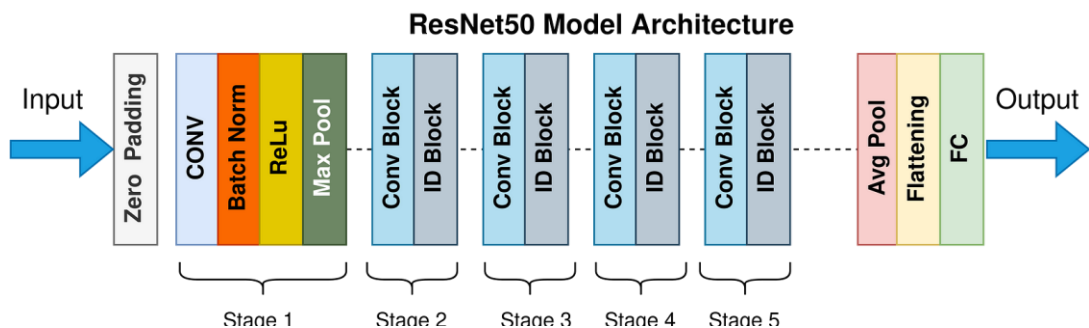
Section V: Model (ResNet50)

ResNet50

ResNet-34 was the original ResNet architecture with 34 weighted layers, which addressed the vanishing gradient problem by introducing shortcut connections. This allowed more convolutional layers to be added to a CNN. ResNets have fewer filters and are less complex than VGGNets, and a 34-layer ResNet can achieve a performance of 3.6 billion FLOPs while an 18-layer ResNet can achieve 1.8 billion FLOPs, which is much faster than VGG-19 with 19.6 billion FLOPs. The ResNet architecture follows two design principles: the number of filters in each layer is the same based on the output feature map size, and if the feature map size is halved, the number of filters is doubled to maintain layer time complexity.

ResNet-50 is an extension of the ResNet architecture that builds upon the model's design principles while incorporating a bottleneck design for the building block. The bottleneck design uses 1×1 convolutions that act as a "bottleneck" to reduce the number of parameters and matrix multiplications. This results in much faster training of each layer. The bottleneck residual block consists of a stack of three layers instead of two layers.

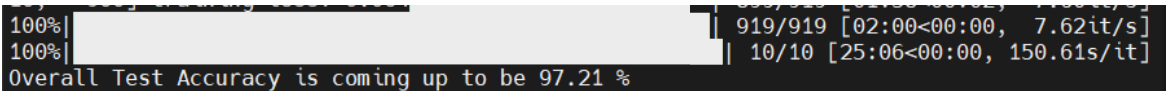
The 50-layer ResNet architecture includes a 7×7 kernel convolution with 64 other kernels that have a 2-sized stride, followed by a max pooling layer with a 2-sized stride. It then has 9 more layers, consisting of a 3×3 , 64 kernel convolution, a 1×1 , 64 kernel convolution, and a 1×1 , 256 kernel convolution, which are repeated three times. The architecture also includes 12 more layers with 1×1 , 128 kernels, 3×3 , 128 kernels, and 1×1 , 512 kernels, which are iterated four times. Additionally, there are 18 more layers with 1×1 , 256 kernels, and 2 cores of 3×3 , 256 kernels and 1×1 , 1024 kernels, which are iterated six times. Finally, the architecture has 9 more layers with 1×1 , 512 kernels, 3×3 , 512 kernels, and 1×1 , 2048 kernels, which are iterated three times. The network comprises a total of 50 layers.



Section VI: Results & Conclusion

Results:

| ResNet50 | Accuracy | | Loss | |
|---|-----------|-------------|-----------|-------------|
| Model Architecture & Augmentation | Test Data | Hidden Data | Test Data | Hidden Data |
| <ul style="list-style-type: none">Normalization60 RotationCenter Crop 100 Size (224 x 224)Vertical & Horizontal Flip with a probability of 50% | 97.21% | 96.88% | 0.064 | - |



Conclusion:

In conclusion, the deep learning model employed in this study achieved an impressive accuracy rate of over 96% on both test data and Kaggle, showcasing its exceptional ability to accurately classify novel data. The success of this model is a testament to the critical role played by techniques such as data standardization and augmentation in enhancing the performance of deep learning models. These methods provide a robust means of improving the model's ability to generalize its learned features to novel data. By leveraging these techniques, the model was able to effectively extract meaningful patterns from the data, enabling it to perform classification with remarkable accuracy. The outcomes of this study underscore the significant impact of data preprocessing techniques on the overall efficacy of deep learning models, highlighting their crucial role in achieving optimal model performance.

Section VII: References

[1] *Class Notes and Materials*

[2] *Towards Data Science*

[3] *MathExchange*

[4] *StackOverflow*

[5] *Collaboration with other students*