Project II - Report

Performance Analysis of Machine Learning and Deep Learning Architectures for Malaria Detection on Cell Images

Shreyas Raghunath

08.03.2020

INTRODUCTION

Plasmodium malaria is a parasitic protozoan that causes malaria in humans. In this project we adopt Convolution neural network methods(both deep and shallow) to classify if a given cell image is Parasitic or Uninfected.

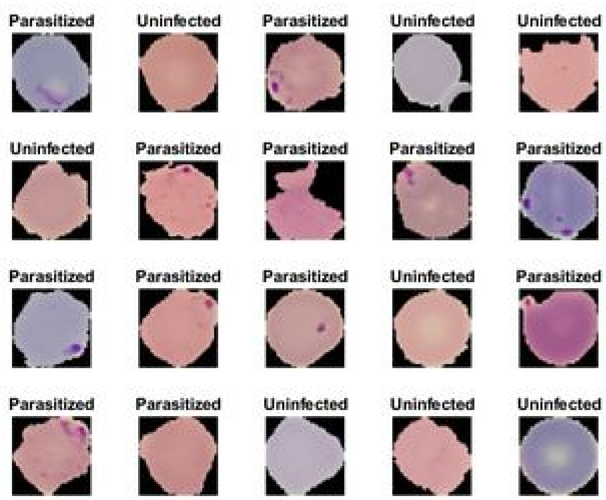
The approaches adopted in this project.

1. Fast CNN (Adapted and changed from the proposed model)
2. Transfer learning Model(VGG-16 Modified)

The Data

The data consists of 13778 Parasitic and 13779 uninfected images, which is fairly balanced.

A sample of the data is visualized below



The above visualized data represent the only 2 classes that are required to be classified.

However these images are in varying size and have differences in attributes such as intensity contrast etc.

So to use the data on our CNNs we need standardize the data so that it has the actual size of the first convolution layer and alter its characteristics uniformly.

Data Preprocessing

The main idea of preprocessing the data is to make the data adaptable to CNN and make the individual features more visible.

For instance if a minor area of infection is responsible for classifying a particular cell as parasitic then that feature has to stand out for the algorithm to learn.

In this case we have 2 sorts of preprocessing:

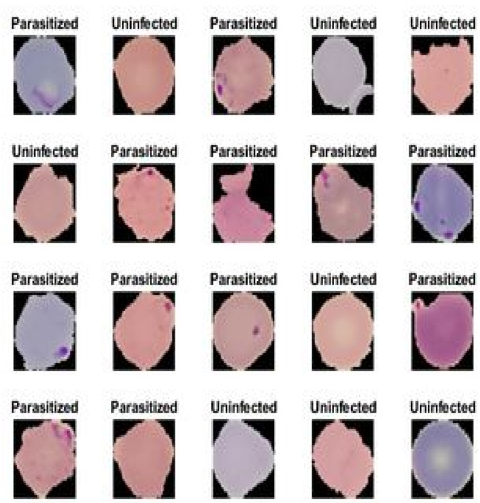
1. Resizing the image
2. Color Constancy

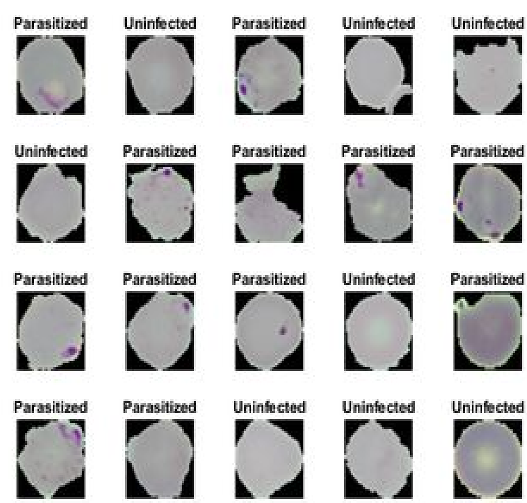
We have arrived at the color constancy method after checking the accuracy with other methods such as Color thresholding and Histogram equalization

As discussed earlier we resize the image to fit the first Convolution layer.

Color constancy however help us in isolating the infected area easily

The below visualization explains how color constancy works on cell images





What the color constancy function basically does is:

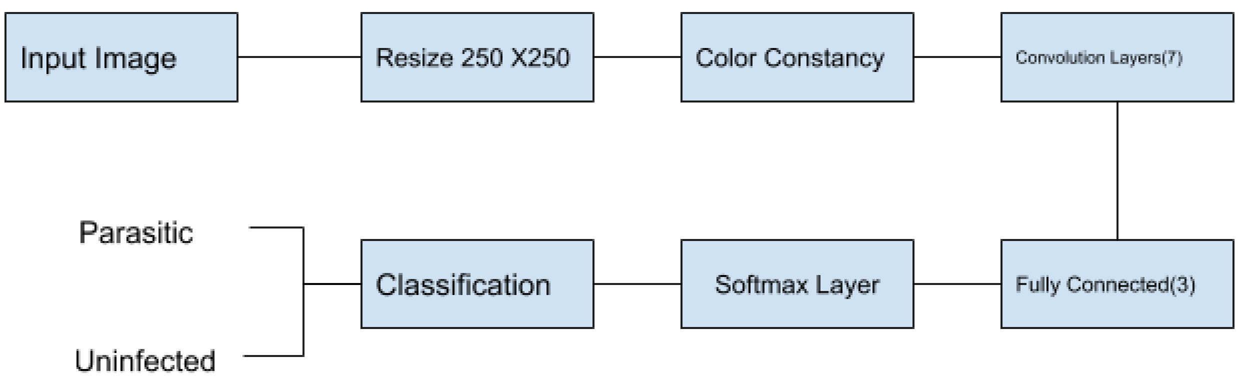
1. Separates the given images into 3 channels
2. Calculates average of all pixel values of the image

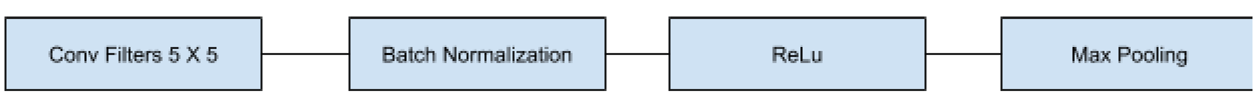
3. Calculates the mean for these values across all the 3 channels
4. Calculate the product of (3) and the matrix of each channel
5. Concat them back into a single image

The advantage of this method is that, you can clearly see from the visual that all infected cells have a purple colored pigment which stands out unique this makes it very easy for the system to learn

Proposed Fast CNN Architecture

The architecture adopted for this project has been built upon the architecture provided in the reference paper



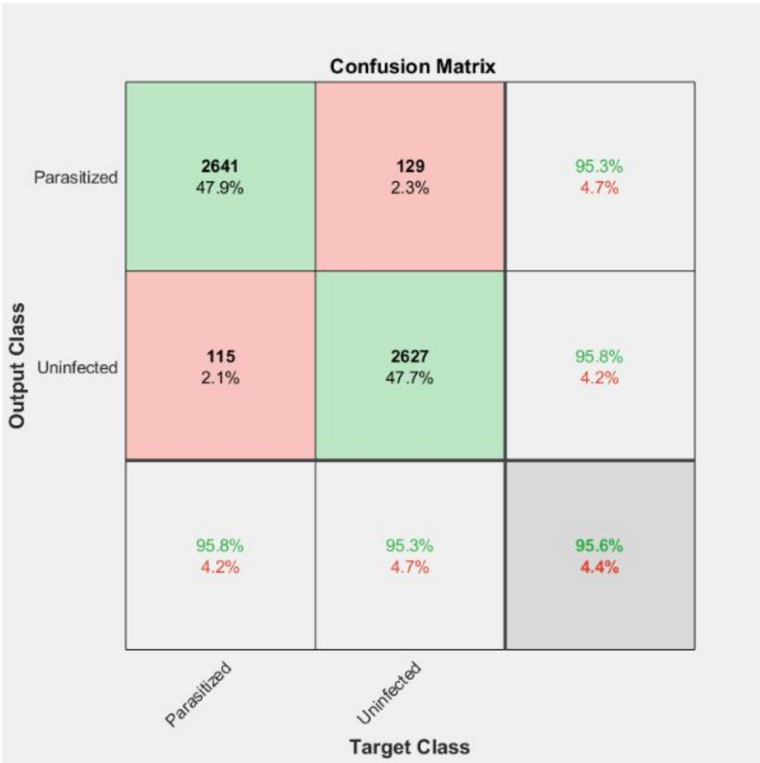


All filter sizes are 5 X 5

Convolution Layer	Number of filters
1	8
2	16
3	32
4	64
5	128
6	256
7	512

The training Accuracy of this model is 95.25

The confusion Matrix of the above model is as below



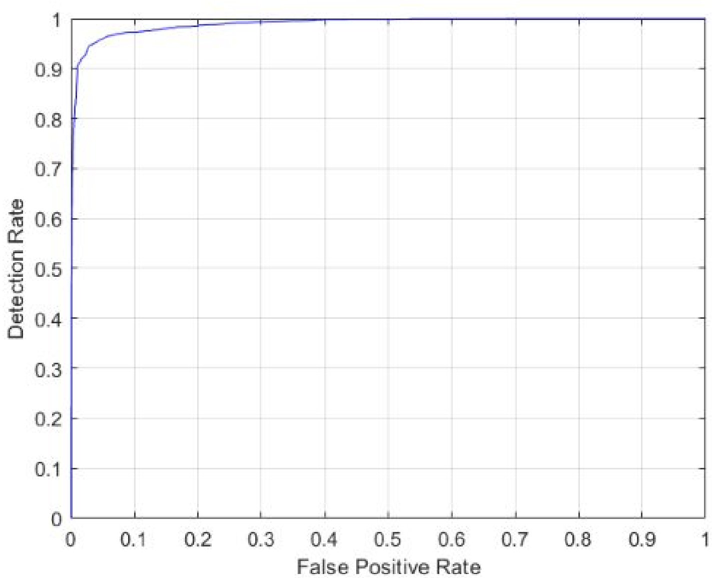
Number of Epochs 4 and Mini batch Size 64

The results with the same model were also tested without **Data augmentation and color constancy** and the confusion matrix is as below

The accuracy When Image preprocessing was not involved is 91%

The confusion matrix was lost during execution , sorry for the inconvenience

The ROC curve is as follows



AUC

```
AUC = 0.9896
```

Steps that can be taken to improve accuracy:

1. Come up with a better pre processing function
2. Improve training with more rigorous data augmentation

Transfer Learning Approach:

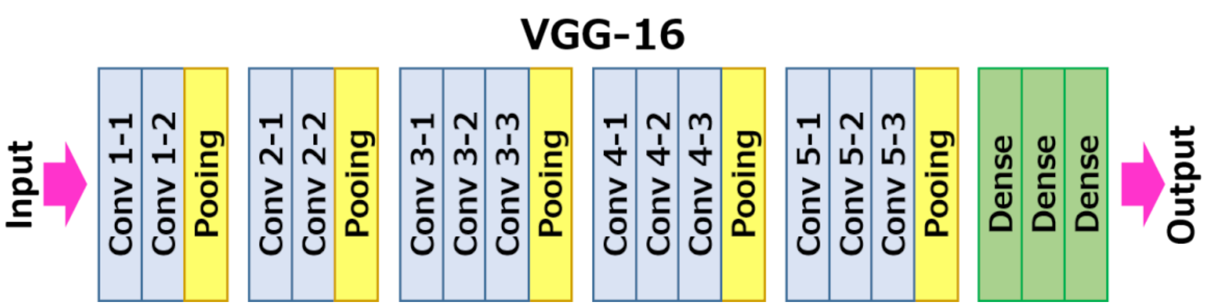
Transfer learning is a research problem in machine learning that focuses on storing knowledge gained while solving one problem and applying it to a different but related problem.

In this project we have used VGG-16, a CNN used to classify 1000 classes of images.

Why was this choses over other models:

1. Proven method for image classification
2. Lesser layers compared to other networks
3. Easier to computer with less computation power

VGG 16 Architecture



The last 3 dense layers are constructed to accommodate 1000 classes, however in our case we only need to classify 2.

The other reason why these layers were removed and made as one class is because of lack of computation power.

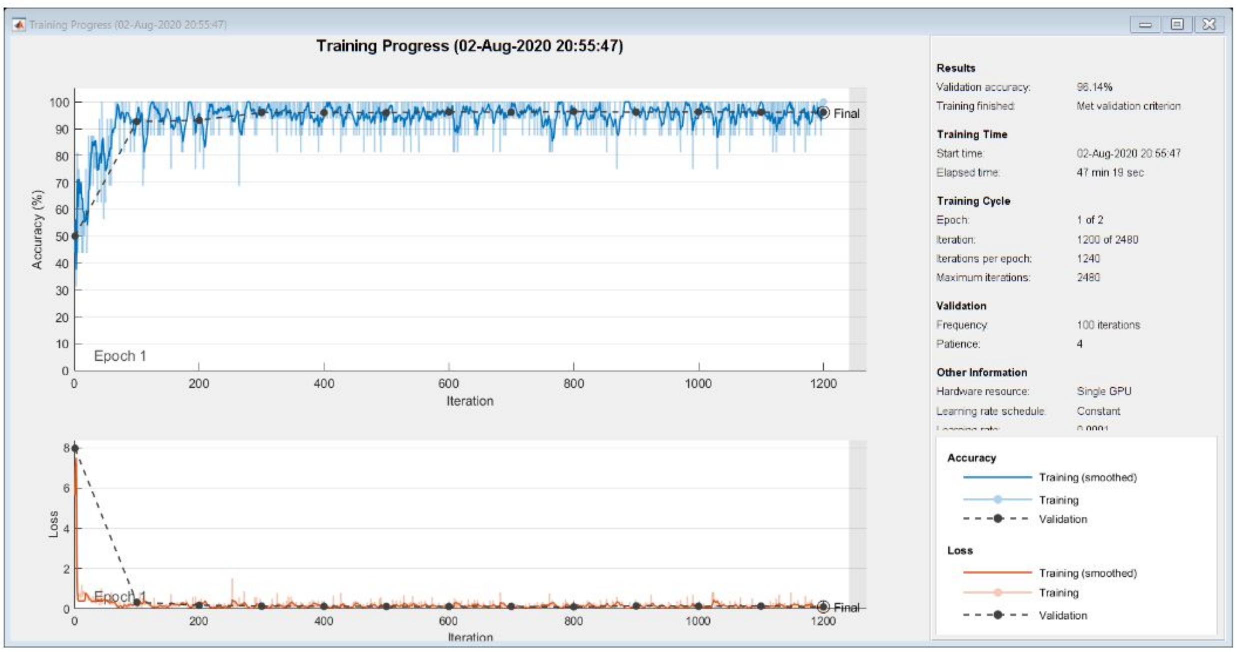
Pre Processing and Data Augmentation

The preprocessing and data augmentation is the same as the one used for the fast CNN method.

The data augmentation methodology used is as follows:

1. Use a random generator to generate a number between 1 and 4
2. Based on the result the image is rotated to 90, 180 or 270 degrees clockwise

The Results

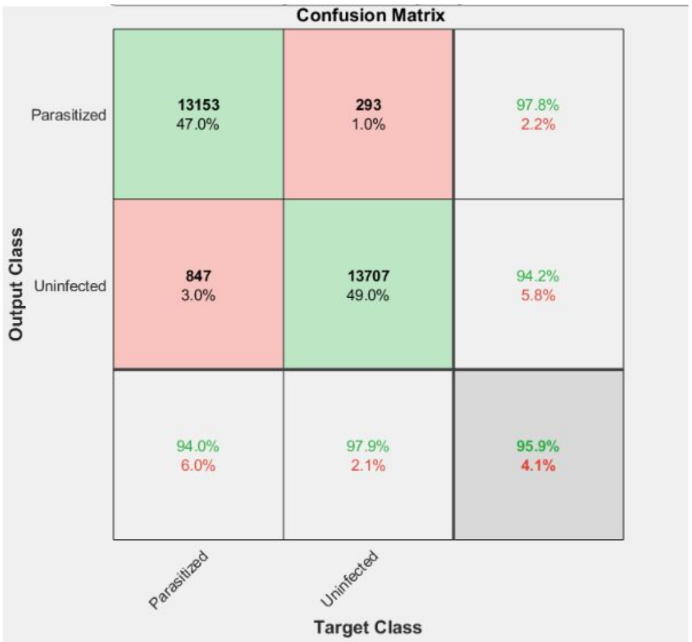


The Training Accuracy is 96.14%

Number of Epochs : 2

Min Batch Size : 16

The Confusion Matrix

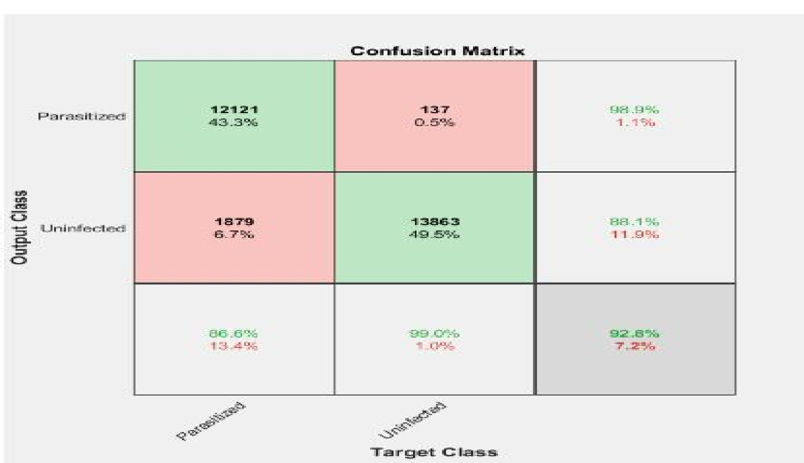


Due to poor computation power testing with the entire test data set was not possible.

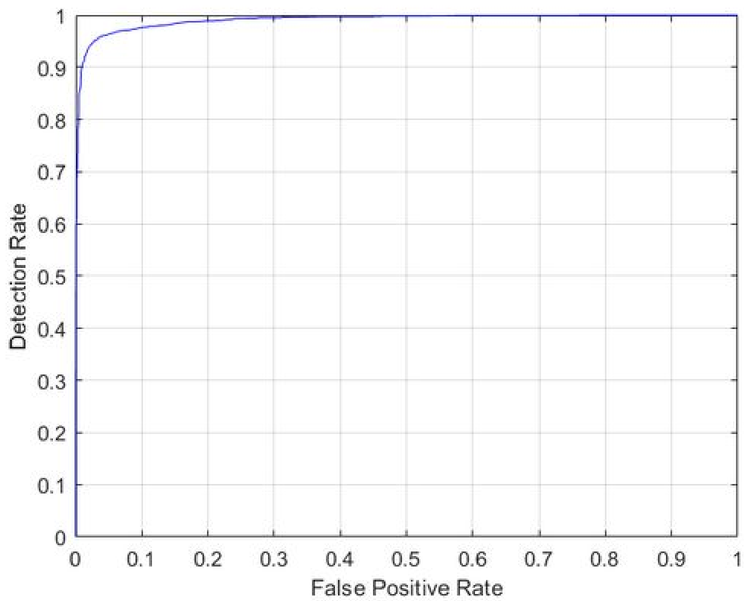
To overcome this issue we've reduces the test dataset to a smaller dataset size and used a loop to generate multiple results and combine them.

Since the splitting function is a random function we can say that the results are accurate.

Below is the confusion matrix when there is no preprocessing or data augmentation is involved



AUC Curve




```
AUC
```

```
AUC = 0.9910
```

Summary of Results

Name of the Method	Training accuracy	Testing accuracy	AUC
Fast CNN	95.25	95.6	0.9896
VGG-16 (Modified)	96.14	95.9	0.9910

Suggestions:

1. Arrive at a better preprocessing function to increase the accuracy
2. The component of paramount importance is computational power so before starting such a project it is important to make sure one has higher computational power
3. The number of layers can surely be increased in the Transfer learning approach
4. Provided the computational power one can experiment with other possible models as well