CAP 5516 Final Project

Michael Fielder
University of Central Florida
College of Engineering and Computer Science
Department of Computer Science
MichaelFielder@knights.ucf.edu

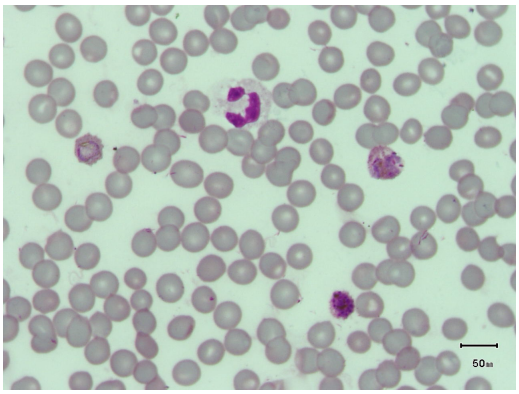

Figure 1. IML MALARIA Dataset Imag

Abstract

Edge computing has gained more attention in recent times. The prospect of being able to deploy a model on an isolated, low-power device is one that has many applications. This is also true for medical imaging, where embedded applications can preovide a cost effective and small-footprint solution. My project's main objective is to train a low-parameter model to identify if a cell is malaria infected. Through my research, I was able to train a MobileNetV2 model pre-trained on ImageNet to successfully perform the task of binary classification on whether a cell is normal or malaria infected with high accuracy.

1. Introduction

The paper *A Dataset and Benchmark for Malaria Life-Cycle Classification in Thin Blood Smear Images* serves as the foundation for my research. They provide a dataset of microscopy images containing red blood cells and a baseline model for identifying malaria infested cells. There method uses the watershed algorithm to separate cells and then uses a ResNet50 model for classification.

I decide to use a pre-trained MobileNetV2 architecture for this classification task instead. MobileNet was an architecture created to be used for Mobile application. This means that it has less parameters than larger models like ResNet which ultimately leads to a more compact and smaller architecture.

2. Dataset Creation

2.1. Overview

In order to construct the dataset, I created an altered dataset from the existing `IML_MALARIA` dataset. Instead of using the full microscopy images containing multiple red blood cells, I isolated each individual cell using its provided bounding box and label (Normal or Malaria). Each entry into this dataset contains only one cell and is used in my binary classification task. Figure 1 shows an example of an image from the original dataset and Figure 2 is an example of a deconstructed image. After each image is deconstruvted, there are 37899 *NORMAL* and 550 *INFECTED* cells in the dataset.

2.2. Project Database

It was necessary to subsample the data not only to fix the significant class imbalance present in the data, but to make the model possible to train. 38k images would be impossible to train with my current hardware. In order to fix the imbalance present, I decided to oversample the unbalanced MALARIA class so that model was trained on a similar amount of Malaria and Normal cells. Also, I provided augmentations to the dataset using the `Augmentor` library. On the sampled database, I performed random zooms and reflections to create synthetic images used for training. Figure **??** One thing to note is that I applied these augmentations to both NORMAL and MALARIA cells for consistency. After the sampling and augmentations, the final size of the training dataset is 3007 images with 1383 MALARIA and 1625 NORMAL images. For the test split, there are no augmentations with 525 total images with 25 MALARIA and 500 NORMAL.

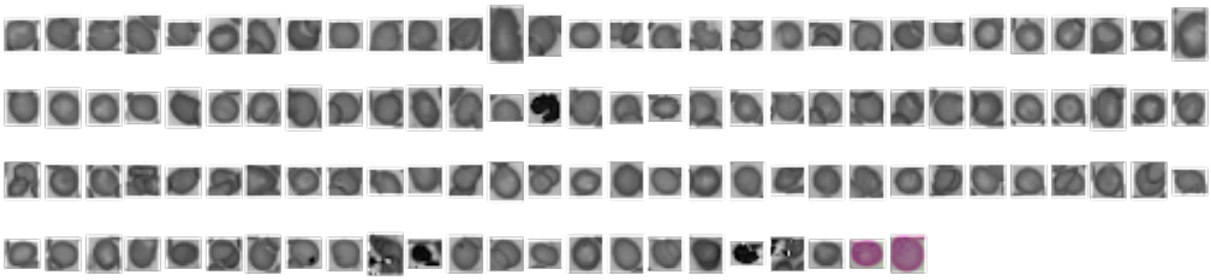

Figure 2. An example of a deconstructed image. For visualization purposes, normal red blood cells are displayed in grayscale and infected cells are in RGB. This is to highlight the unbalanced aspect of the dataset. In the actual dataset, each image is in RGB.

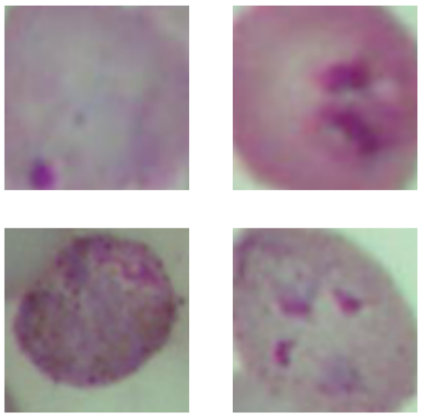

Figure 3. An example of augmented images in training set

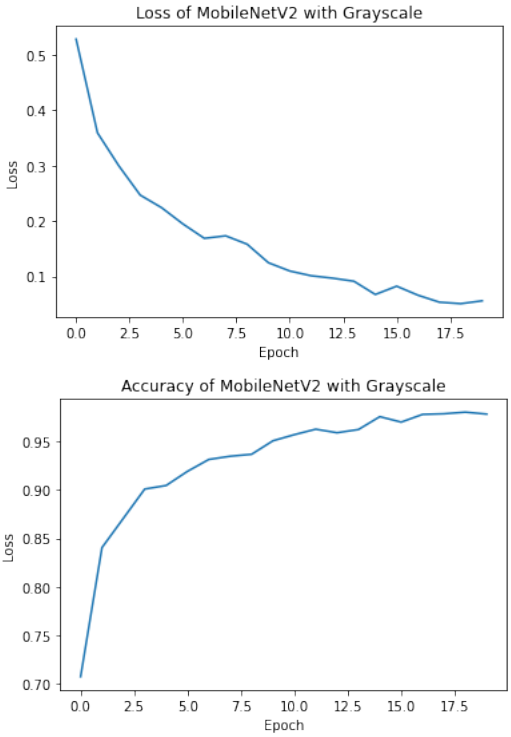

Figure 4. The loss and accuracy of the grayscale input version of MobileNetV2. Although the model's loss seems to converge, its testing accuracy is very low.

3. Failed Training

Several methodologies were attempted but could not reach any effective results. The problem was the same for each failed model, and it was the problem of converging to only predicting one class exclusively. The losses of the model would go down and the accuracy would increasing to values as high as 90%. The issue was that during training, the model would predict one class exclusively to the point that it would achieve an class accuracy of 0 for one of them. I don't believe that this was an issue of class imbalance as my final model trained very effectively. It could come down to not using a pre-trained model or it could have something to do with the output layer.

The first method that was tried was using Keras' `MobileNetV3Small` model trying it with grayscale and RGB images as input. I also tried it as a binary classification task and a multi-class task with 2 classes. None of these configurations were able to train correctly.

The more confusing result is when I tried modifications of MobileNetV2. I attempted to run the architecture using grayscale images instead of RGB. The other difference that this makes to the architecture is that it makes the model unable to be pre-trained on ImageNet since that requires input with 3 channels. This suffers the same problem as the model above and it is unclear why. I made sure that the model's parameters weren't frozen due to the lack of pre-training, but it still failed to converge. In Figure 4 i shows the the loss is converging but the test accuracy was very low.

4. Training

My model was trained for 20 epochs using a ImageNet pre-trained MobileNetV2 model.This involved freezing the base model and appending a dense layer at the end which returns the raw logits. The input to the model was an RGB image resized to be 96×96 passed into the model at a batch

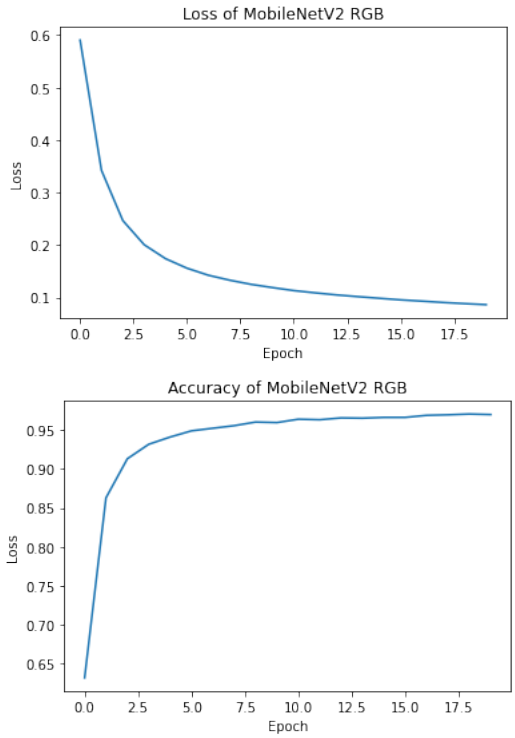

Figure 5. The loss and accuracy of the model over training

size of 16. RMSprop was used for the optimizer and used a learning rate of 0.0001. For the loss function, I used Binary Cross Entropy and set it up to receive the raw logits as inputs. This model's loss converged favorably and achieved good accuracy. Figure 5 shows the loss and accuracy of the model over the training epochs.

5. Testing

The model achieved similar results in testing and was able to achieve a total accuracy of 99.4% with specific class accuracies of 89% for MALARIA and 100% for NORMAL cells. This shows that the model is not just overfitting to the training data. THis is especially true when considering that the distributing of the test split is significantly more skewed than the train split. Overall, I would say that the model was trained correctly as shown by the results.

6. Conclusion

Through my testing, I believe that I have shown that the binary classification task of malaria infected red blood cells can be done on a smaller model whilst maintain very high accuracy.

I performed some testing with TFLite to reduce the size of the model, but it proved to be a deceivingly difficult architecture to use. With more time, I would perform some analysis of model performance using a smaller TFLite model.

7. References

1. https://arxiv.org/pdf/2102.08708.pdf
2. https://arxiv.org/pdf/1801.04381.pdf
3. https://blog.roboflow.com/how-to-train-mobilenetv2-on-a-custom-dataset/