Computer Vision for Malaria Detection - Milestone 2

Sulian Thual, MIT ADSP October 2021

We build a computer vision model for malaria detection, as part of a capstone project for the MIT Applied Data Science Program (ADSP) from August-November 2021. In this second report (Milestone 2), we focus on refined insights, comparison of methods, proposal of a final method and key recommendations.

1 Refined Insights

A main takeaway from the present report is that most methods tested reach around 95% accuracy. Nevertheless, all methods show some limitations for classifiying a few types of red cells that exhibit complex features. Figure 1 shows examples of misclassified red cell images images. In previous report (Milestone 1), we had already discussed the main characteristics of red cell images: Uninfected cells are uniform with sometimes a lighter center, while Parazited cells contain one or several malaria parasites of varying shapes. In Figure 1, misclassified red cells tend to have a deformed envelope/border. In fact, the shape of the envelope is irrelevant for malaria detection and is only an artifact of the experimental method (framing and cropping). Another source of misclassification is very small parasites.

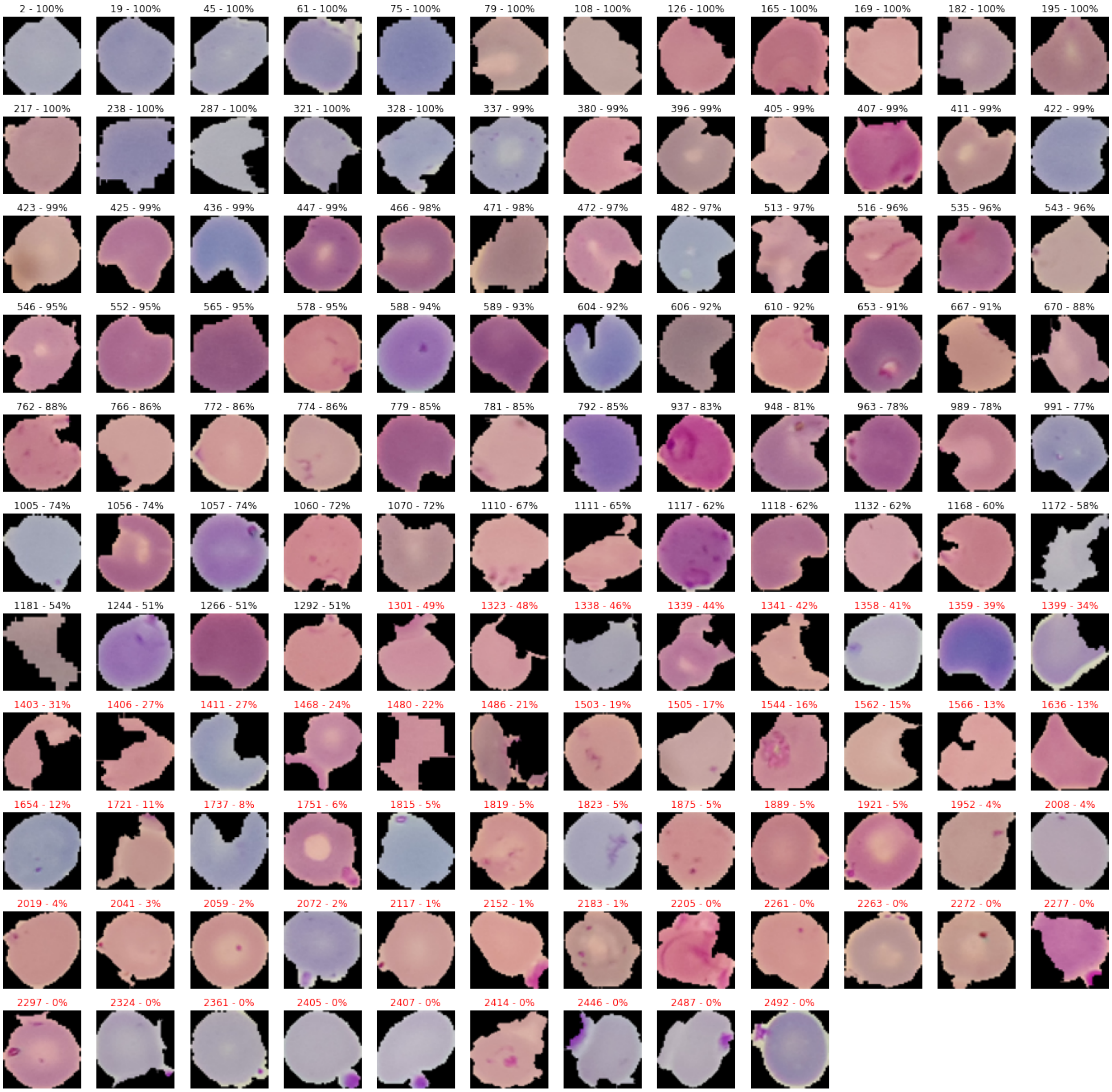

Figure 1: Database of misclassified images (for the experiment with CNN0). Subtitles indicates the image index, as well as the predicted probability p of being parasitized (%). Images with p>50% (black) are predicted as parasitized but are uninfected, while images with p<50% (red) are predicted as uninfected but are parasitized.

2 Comparison of Methods

2.1 Model Architecture

We have experimented with various methods in order to improve the predictive power of the CNN model. The first experiments have been on the architecture of the CNN model, as listed in Table 1. The main takeaway is that all models reach similar accuracy of around 95%, but some are much faster to train. We have first focused on a standard Convolutional Neural Network CNN0 consisting of 3 convolutional layers and 1 fully-connected layers reaching 95% accuracy. CNN1 and CNN2 are small variations of the same model. VGG16 is a notorious model architecture that we use here with pretraining, but without substantial improvement of performance (as it is likely too complex to train adequately). Finally, CNN5 (with greath depth) is arguably the best model because of its fast training time and reduced complexity (i.e. number of parameters).

architecture	accuracy	train time	parameters	comments
CNN0	95%	slow	1m	base model, 3 convolutional layers, 1 fully-connected layer
CNN1	95%	slow	1.3m	CNN0 with additional fully-connected layer
CNN2	94%	medium	1 m	CNN0 with Leaky Relu and Batch Normalization
VGG16	92%	very slow	15m	13 pretrained convolutional layers, 3 fully-connected layers
CNN5	95%	**fast**	0.4m	5 convolutional layers, 4 fully-connected layers

Table 1: List of model architectures tested.

2.2 Data Augmentation

Other experiments have focused on data augmentation, i.e. generating synthetic red cell images to further train the CNN. They are summarized in Table 2. The most useful transformations have been to flip base images and rotate them in 90 degrees increment, as shown in Figure 2, resulting in a slight increase in accuracy to 96%. These are interesting transformations because they could occur naturally, e.g. when a red cell is pictured at a different angle or flipped in the blood smear. Other transformations such as zooms or shifts (that result in spurious image borders) dont bring as much improvements.

data augmentation	base model	accuracy	comments
zoom 50-150%, rotation +-30deg	CNN0	94%	spurious borders
flip, rotation +-360 deg	CNN0	94%	natural transformations
flip,rotation 90,180,270	CNN5	**96%**	natural transformations
periodic shifts	CNN5	95%	spurious borders
flip, rotation, periodic shifts	CNN5	NA	Not Enough RAM

Table 2: List of data augmentation methods tested.

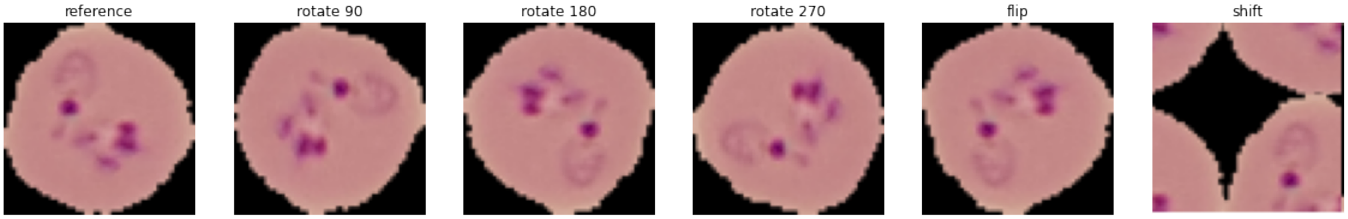

Figure 2: Examples of transformations on a red cell image.

2.3 Feature Engineering

Finally, we have also tested a few feature engineering methods that consist of applying systematic transformations to the image database prior to training. These methods were considered with the goal of removing features from the dataset that are irrelevant for malaria detection (e.g. mean red cell color or shape of the red cell enveloppe). However, these methods did not bring substantial improvements.

feature engineering	base model	accuracy	comments
convert RGB to HSV	CNN5	52%	focus on removing mean color
change borders color (black) to image mean color	CNN5	95%	focus on removing border shape

Table 3: List of feature engineering methods tested.

3 Proposal for Final Method

For the final method, we suggest to combine our best model architecture with data augmentation as well as current hyperparameters. The model architecture is shown in Figure 3: it is the architecture from CNN5, a very deep CNN with relatively few trainable parameters (0.4 million), which we prefer for its fast training. Next, the data augmentation method will consist of rotated and flipped images as described above. Finally, we suggest to use hyperparameters used in this report (e.g. Batch Gradient Descent with Adam optimizer as well as Early Stopping). The above characteristics of the final method may be subject to minor modifications if time allows. For example, we may try to augment the width of several layers resulting in potentially improved predictions but a much costlier training.

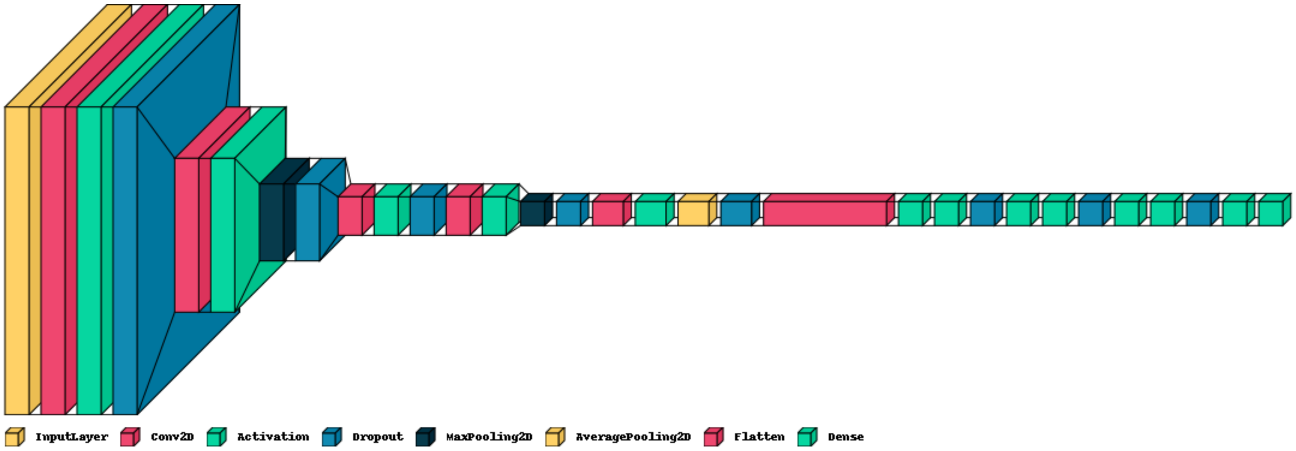

Figure 3: Architecture of the Final Method (using visualkeras).

4 Key Recommendations for Implementation

To implement our final computer vision model, we will fine-tune the final method proposed above: this will consist in making a few costly experiments where we increase the complexity of the reference architecture (e.g. layer width). It is however likely that the final model will remain near the baseline of accuracy 95%

found for most of our experiments. Recall however that in practical settings not just one but many red cells from a patient's blood smear would be analyzed: Figure 4 illustrates the effect of the 95% accuracy (i.e. 5% error) in such a setting, showing that the count of parasitized cells becomes very accurate for increased sampling of red cells. In practice, analyzing around 1000 red cells from a patient (as done from a single blood smear) seems to be sufficient to diagnose malaria from counts of parasitized cells despite the 95% accuracy of the computer vision model.

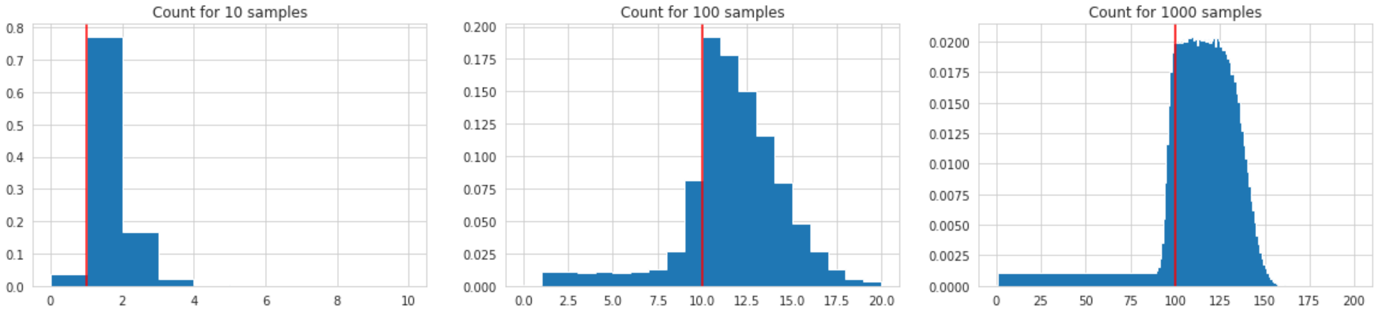

Figure 4: Distribution of predicted count of parasitized cells (blue) vs true count (red). We assume that 10% of the patient cells are infected and that is each cell is correctly classified at 95% accuracy. This is repeated for an experiment where 10 (left), 100 (middle) or 1000(right) red cells are analyzed.

References

- VisualKeras library: https://github.com/paulgavrikov/visualkeras
- VGG16: https://arxiv.org/abs/1409.1556

Appendix: Jupyter Notebook

Two jupyter notebooks (Part I and Part II) are attached to this report. My apologies for submitting two notebooks which is because of the RAM/Memory limitations I encountered with Google Collab.