Capstone Project: Malaria Detection

Executive Summary

This project proposes the Convolutional Neural Network (CNN) model for Malaria disease detection. This model was suggested for its robustness and better performance at larger amounts of information, which in our case consists of a total of 27558 photographs of infected and uninfected cells. Although the performance of CNNs at larger amounts of information is usually acceptable, in this case because it is a public health problem, it is suggested to improve the performance of the CNN model with the implementation of Gaussian Blurring, this tool can help the model to be more resilient to images that could have bad focus or defects making our model even more robust. The variables to take into account are the false negative rate (included in recall variable for parasitized predictions) and F1-score for being a public health problem.

Problem Definition

Malaria is an acute febrile illness caused by Plasmodium parasites, which are transmitted to people through the bites of infected female Anopheles mosquitoes. The first symptoms - fever, headache and chills - usually appear 10 to 15 days after the infectious mosquito bite and can be mild and difficult to recognize as malaria. If left untreated, P. falciparum malaria can progress to severe illness and death within 24 hours.By 2020, nearly half of the world's population was at risk of malaria. Some population groups are at significantly higher risk of contracting malaria and developing severe disease: infants, children under 5 years of age, pregnant women, and HIV/AIDS patients[1]. Traditional diagnosis of malaria in the laboratory requires careful inspection by an experienced professional to discriminate between healthy and infected red blood cells. It is a tedious, time-consuming process, and the diagnostic accuracy (which heavily depends on human expertise) can be adversely impacted by inter-observer variability[2].

Hypothesis

Applications of automated classification techniques using machine learning (ML) and artificial intelligence (AI) have consistently demonstrated higher accuracy than manual classification[2]. Based on the above, an automated system can aid in the early and accurate detection of malaria.

Objective

Build an effective computer vision model to detect malaria. The model should identify whether the blood cell in question is that of a malaria-infected cell or not, and classify it as parasitized or uninfected, respectively.

Data exploration

The data set to be used consists of 27558 colored images of red blood cells (24958 for training set and 2600 for testing set) containing parasitized and uninfected cells, where: parasitized cells contain Plasmodium parasite and uninfected cells are free of Plasmodium parasites but could contain other impurities.

Of the total images corresponding to the training set (24958) 12582 belong to the parasitized group and 12376 to the uninfected group, this 50/50 % represents a good balance to develop a CNN model with good performance. A similar behavior is observed in the test set which consists of a total of 2600 images, of which 1300 are parasitized and 1300 are uninfected. The test set corresponds to approximately 10 % of the total images in the training set, which is a good ratio for developing a reliable model.

In order to facilitate the management of the graphic information, it is recommended in both sets (train and test) to homogenize the image size, identify the pixel range and normalize the values in the range 0-1.

Proposed approach

One of the first results obtained can be seen in Figure 1, the average color of a parasitized cell is slightly stronger than the average color of an uninfected cell that is lighter, this may be because the parasite in the cell has stronger color contrasts that affect the average color. This feature could be important for the implementation of Machine Learning (ML) techniques such as logistic regression or even Decision Trees. However, CNN models tend to have a better performance at higher amounts of information as in our case, this is important to take into account since it is a public health problem and what is pursued is to minimize the false negative rate by increasing the recall and precision (F1-score).

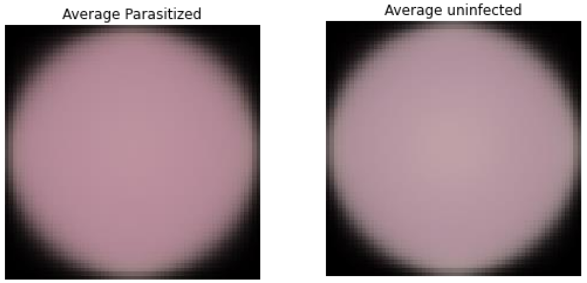


Figure 1. Color-averaged images of parasitized and uninfected cells

The original format of the images was changed to HSV format in order to have a better contrast between the cell and the parasite and thus facilitate the identification of the CNN model, the result can be seen in Figure 2.

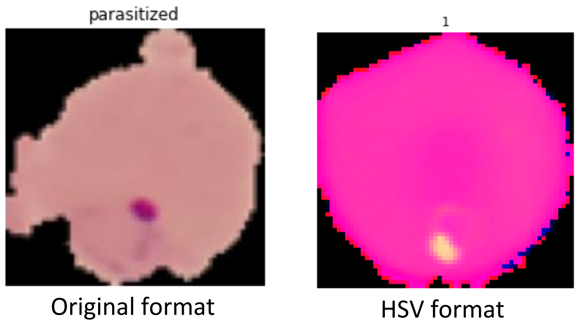


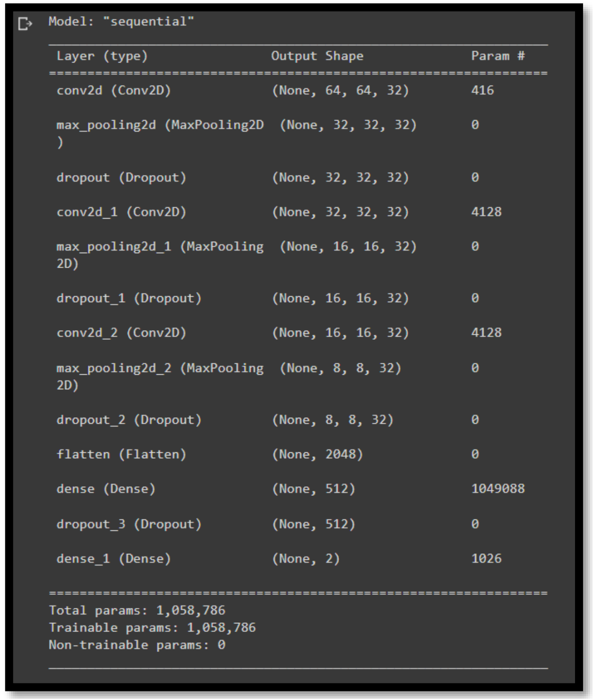
Figure 2. Color-averaged images of parasitized and uninfected cells

When we train computer vision models, we often take ideal photos of our subjects. We line up our subject just right and curate datasets of best case lighting. But our deep learning models in production aren't so lucky. Deliberately introducing imperfections into our datasets is essential to making our machine learning models more resilient to the harsh realities they'll encounter in real world situations[3]. For this reason we decided to implement the Gaussian blurring tool, which allows us to degrade our images in order to make our model more resilient to possible imperfections in real field images. Box blurring method and a differential privacy-based pixilation method can also be used[4].

Refined Insights

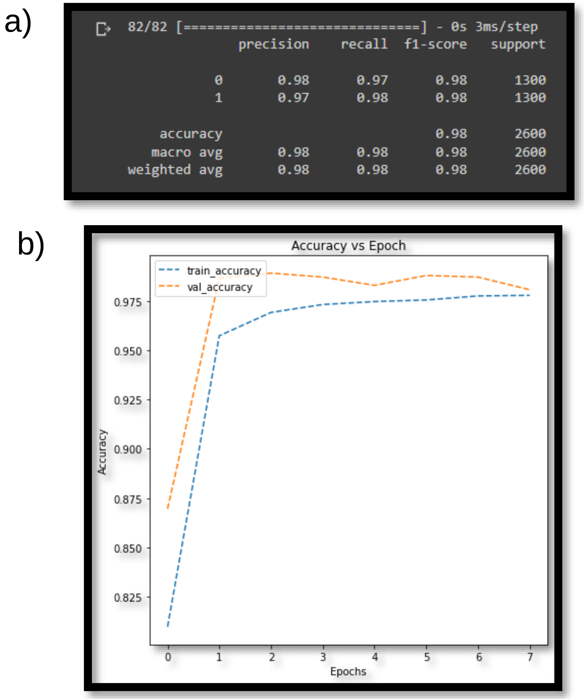
This project proposes the Convolutional Neural Network (CNN) model for Malaria disease detection. This model was suggested for its robustness and better performance at larger amounts of information, which in our case consists of a total of 27558 photographs of infected and uninfected cells. Although the performance of CNNs at larger amounts of information is usually acceptable, in this case because it is a public health problem, it is suggested to improve the performance of the CNN model with the implementation of Gaussian Blurring, this tool can help the model to be more resilient to images that could have bad focus or defects making our model even more robust. The variables to take into account are the false negative rate (included in recall variable for parasitized predictions) and F1-score for being a public health problem.

Comparison of various techniques and their relative performance



```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 64, 64, 32)        416

 max_pooling2d (MaxPooling2D  (None, 32, 32, 32)       0
 )

 dropout (Dropout)           (None, 32, 32, 32)        0

 conv2d_1 (Conv2D)           (None, 32, 32, 32)        4128

 max_pooling2d_1 (MaxPooling  (None, 16, 16, 32)       0
 2D)

 dropout_1 (Dropout)         (None, 16, 16, 32)        0

 conv2d_2 (Conv2D)           (None, 16, 16, 32)        4128

 max_pooling2d_2 (MaxPooling  (None, 8, 8, 32)         0
 2D)

 dropout_2 (Dropout)         (None, 8, 8, 32)          0

 flatten (Flatten)           (None, 2048)              0

 dense (Dense)               (None, 512)               1049088

 dropout_3 (Dropout)         (None, 512)               0

 dense_1 (Dense)             (None, 2)                 1026

=================================================================
Total params: 1,058,786
Trainable params: 1,058,786
Non-trainable params: 0
_________________________________________________________________
```

Figure 3. Structure of the base Model (Layers)



a)

```
82/82 [==============================] - 0s 3ms/step
              precision    recall  f1-score   support

           0       0.98      0.97      0.98      1300
           1       0.97      0.98      0.98      1300

    accuracy                           0.98      2600
   macro avg       0.98      0.98      0.98      2600
weighted avg       0.98      0.98      0.98      2600
```

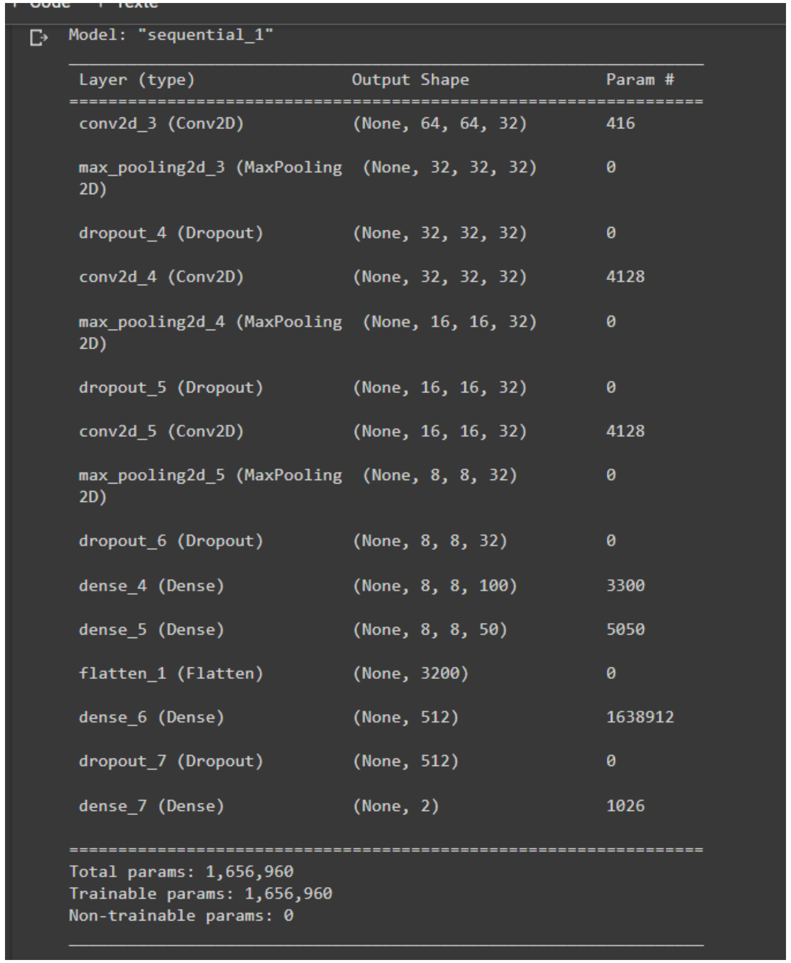
b)

Figure 4. General report of base model performance (a)) and evolution of accuracy as a function of epochs for validation and train data (b))

Figure 3 shows the base model, whose main characteristics are the use of the activation function "relu" and the Dropout function as a method of regularization of the model. This initial model has 1058786 trainable parameters.

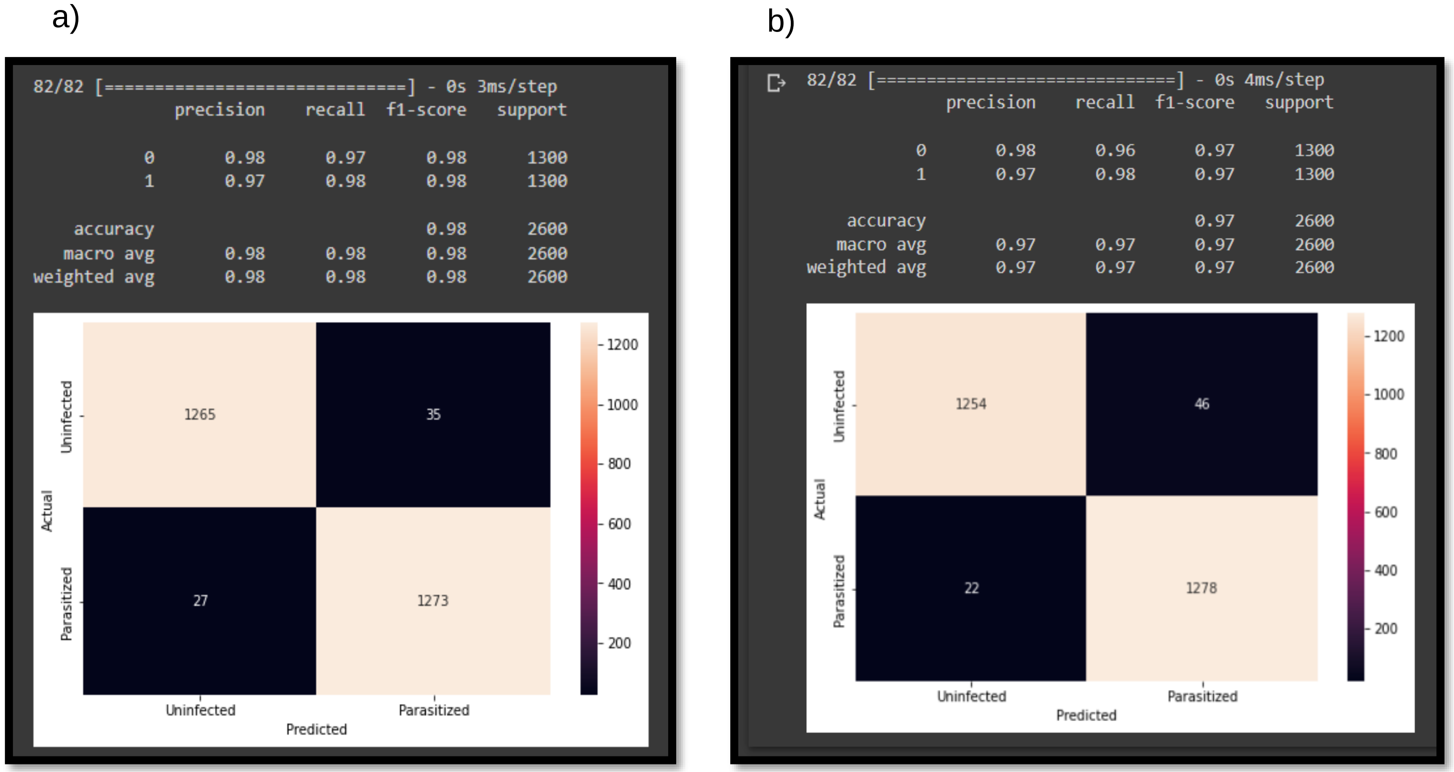
Figure 4 shows the main results obtained for this model, 0.98 was calculated as f1-score, the recall value for uninfected predictions was 0.97 and for parasitized predictions 0.98. Regarding the evolution of accuracy as a function of epochs, Here we can clearly observe that the training and validation accuracy are increasing and we can also notice that validation accuracy is slightly higher than the train accuracy.

As we can see in Figure 5 two more layers were added to the base model before the Flatten layer (resulted in model 1), the first of 100 neurons and the second of 50 neurons. It could be observed in Figure 6 that the false positives (it is desired to minimize this variable) decreased slightly, from 27 to 22 false negatives, however the false positives increased slightly from 35 to 46. The f1-score value decreased from 0.98 to 0.97



```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_3 (Conv2D)           (None, 64, 64, 32)        416

 max_pooling2d_3 (MaxPooling  (None, 32, 32, 32)       0
 2D)

 dropout_4 (Dropout)         (None, 32, 32, 32)        0

 conv2d_4 (Conv2D)           (None, 32, 32, 32)        4128

 max_pooling2d_4 (MaxPooling  (None, 16, 16, 32)       0
 2D)

 dropout_5 (Dropout)         (None, 16, 16, 32)        0

 conv2d_5 (Conv2D)           (None, 16, 16, 32)        4128

 max_pooling2d_5 (MaxPooling  (None, 8, 8, 32)         0
 2D)

 dropout_6 (Dropout)         (None, 8, 8, 32)          0

 dense_4 (Dense)             (None, 8, 8, 100)         3300

 dense_5 (Dense)             (None, 8, 8, 50)          5050

 flatten_1 (Flatten)         (None, 3200)              0

 dense_6 (Dense)             (None, 512)               1638912

 dropout_7 (Dropout)         (None, 512)               0

 dense_7 (Dense)             (None, 2)                 1026

=================================================================
Total params: 1,656,960
Trainable params: 1,656,960
Non-trainable params: 0
_________________________________________________________________
```

Figure 5. Structure of Model 1 (Layers)



a)

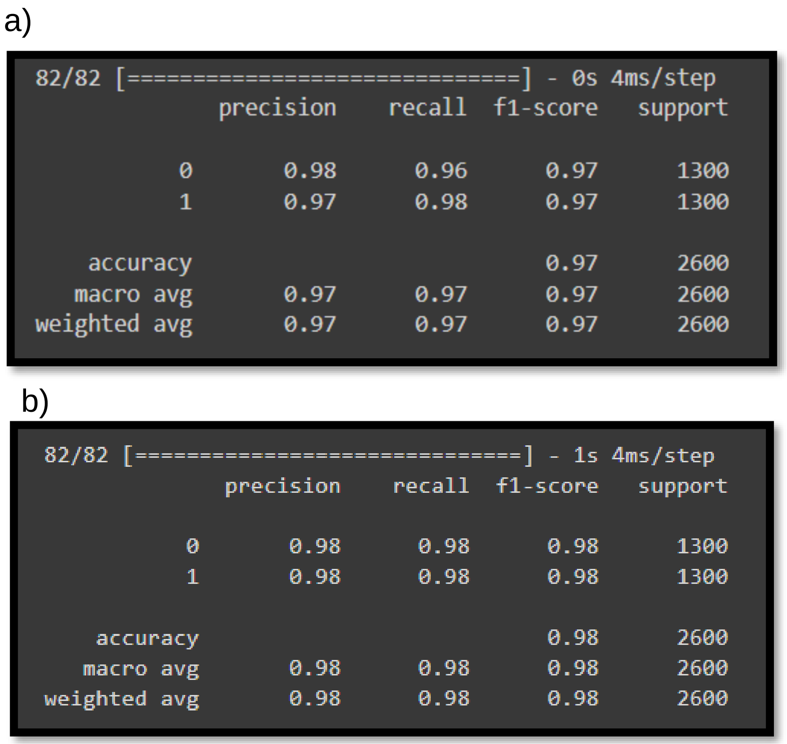
```
82/82 [==============================] - 0s 3ms/step
              precision    recall  f1-score   support

           0       0.98      0.97      0.98      1300
           1       0.97      0.98      0.98      1300

    accuracy                           0.98      2600
   macro avg       0.98      0.98      0.98      2600
weighted avg       0.98      0.98      0.98      2600
```

b)

```
82/82 [==============================] - 0s 4ms/step
              precision    recall  f1-score   support

           0       0.98      0.96      0.97      1300
           1       0.97      0.98      0.97      1300

    accuracy                           0.97      2600
   macro avg       0.97      0.97      0.97      2600
weighted avg       0.97      0.97      0.97      2600
```

Figure 6. General report of performance and confusion matrix of base model (a)) and model 1 (b))

We tried to improve the performance of model 1 (base model + 2 layers) by replacing the "relu" activation by the "LeakyRelu" activation and also by incorporating "BatchNormalization" instead of "Dropout", which resulted in model 2. An improvement in the performance of model 2 could be observed (Figure 7). The recall value for uninfected predictions increased from 0.96 to 0.98 and the recall value for parasitized predictions remained constant at 0.98. The f1-Score value for false positives and false negatives increased from 0.97 to 0.98. In the other hand, one behavior that is important to take into account is the evolution of accuracy as a function of epochs, we can see in Figure 8 a fluctuating behavior especially in the validation curve which was not observed in model 1, suggesting instability of the model, which could trigger problems to converge.



a)

```
82/82 [==============================] - 0s 4ms/step
              precision    recall  f1-score   support

           0       0.98      0.96      0.97      1300
           1       0.97      0.98      0.97      1300

    accuracy                           0.97      2600
   macro avg       0.97      0.97      0.97      2600
weighted avg       0.97      0.97      0.97      2600
```

b)

```
82/82 [==============================] - 1s 4ms/step
              precision    recall  f1-score   support

           0       0.98      0.98      0.98      1300
           1       0.98      0.98      0.98      1300

    accuracy                           0.98      2600
   macro avg       0.98      0.98      0.98      2600
weighted avg       0.98      0.98      0.98      2600
```

Figure 7. General report of performance of model 1 (a)) and model 2 (b))

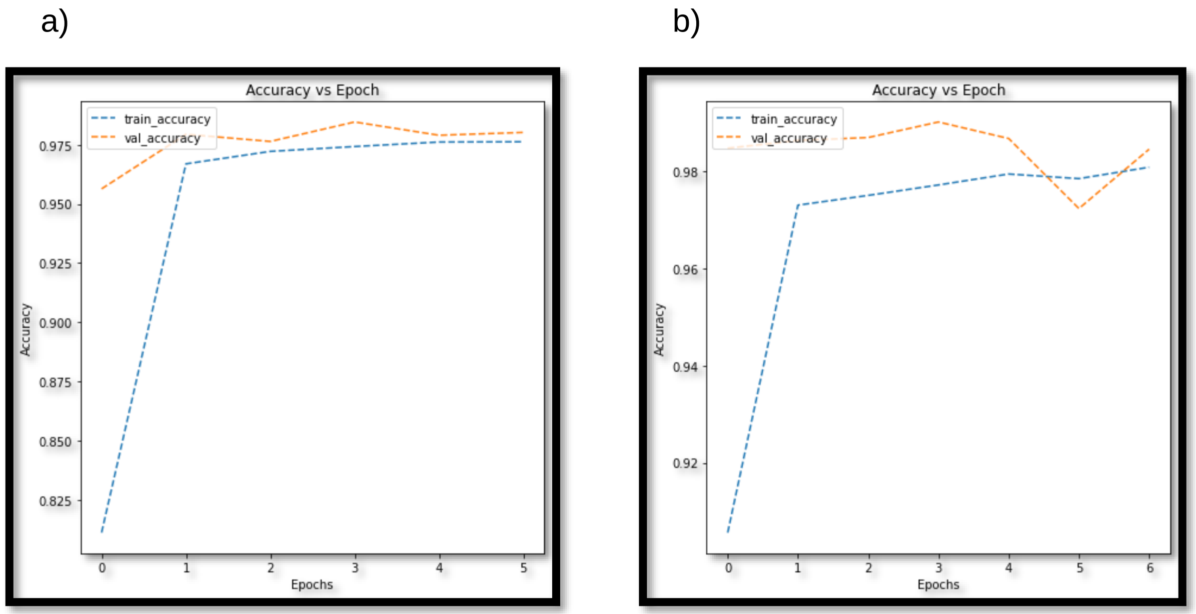
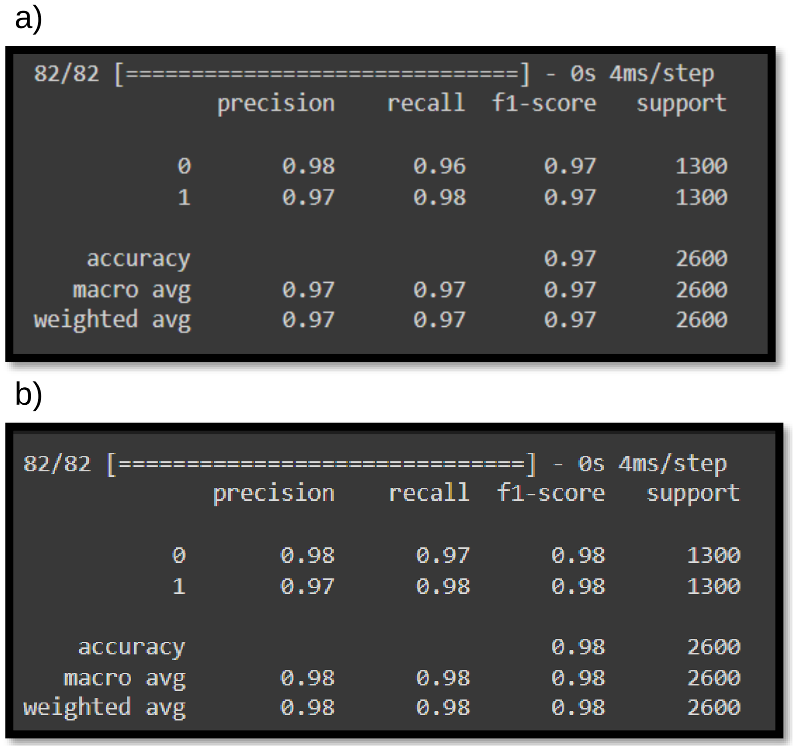


Figure 8. Evolution of accuracy as a function of epochs for validation and train data for model 1 (a)) and model 2 (b))

The implementation of data augmentation technique to improve the performance of model 1 was proposed. The performance of model 1 was improved by incorporating the data augmentation technique, resulting in model 3. Since it is well known that it is a tool that allows us to increase our training dataset to improve accuracy, generalization, and control overfitting.

An improvement in the performance of model 3 was observed as is shown in Figure 9. The recall value for uninfected predictions increased from 0.96 to 0.97 and the recall value for parasitized predictions remained constant at 0.98. The f1-Score value for false positives and negatives went from 0.97 to 0.98. If we observe the evolution of accuracy as a function of epochs (Figure 10), we can appreciate a behavior without abrupt fluctuations in both curves (validation and train), which suggests stability of the model 3, which could indicate that the model could have less difficulties in converging.



a)

```
82/82 [==============================] - 0s 4ms/step
              precision    recall  f1-score   support

           0       0.98      0.96      0.97      1300
           1       0.97      0.98      0.97      1300

    accuracy                           0.97      2600
   macro avg       0.97      0.97      0.97      2600
weighted avg       0.97      0.97      0.97      2600
```

b)

```
82/82 [==============================] - 0s 4ms/step
              precision    recall  f1-score   support

           0       0.98      0.97      0.98      1300
           1       0.97      0.98      0.98      1300

    accuracy                           0.98      2600
   macro avg       0.98      0.98      0.98      2600
weighted avg       0.98      0.98      0.98      2600
```

Figure 7. General report of performance of model 1 (a)) and model 3 (b))

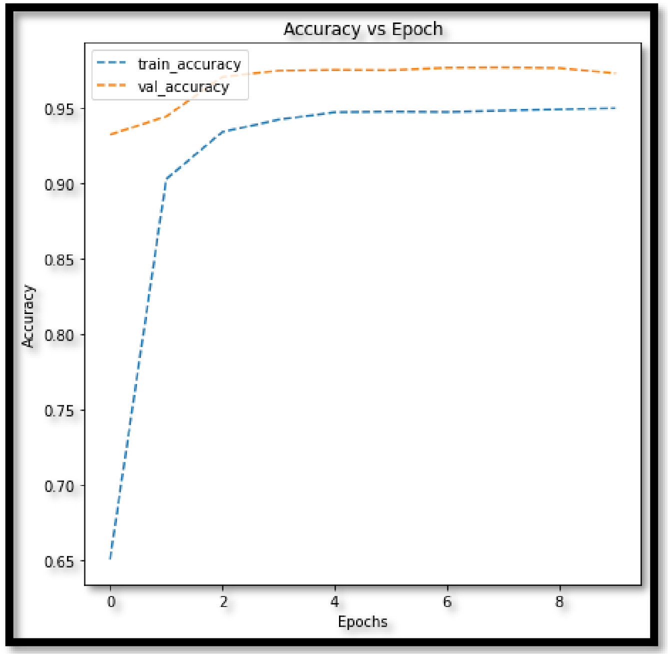
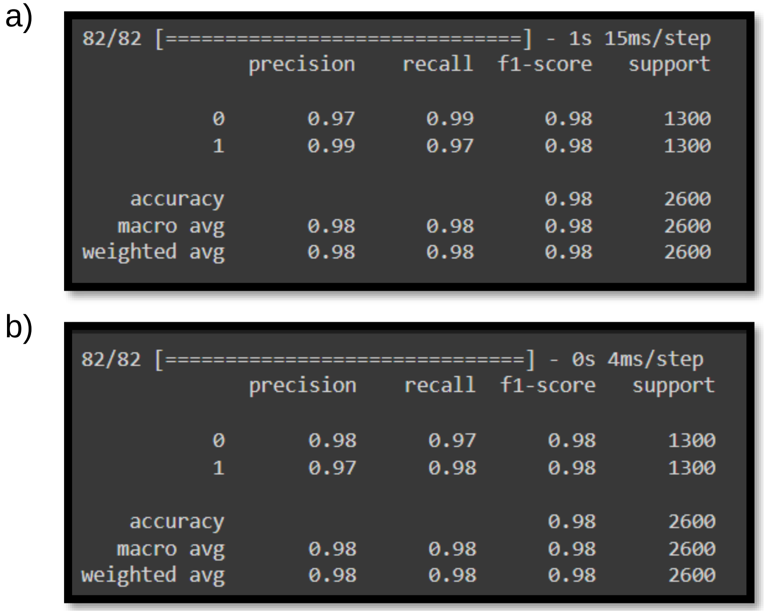


Figure 8. Evolution of accuracy as a function of epochs for validation and train data for model 3

It was decided to test the pre-trained VGG-16 (transfer learning) model for our imaging system. As we noticed in Figure 11 a slight increase in the recall for uninfected predictions with respect to model 3 from 0.97 to 0.99, but decreased the recall for parasitized predictions from 0.98 to 0.97, maintaining an f1-scpre of 0.98 for both models. However, if we observe the behavior of the accuracy variable as a function of epochs (Figure 12), we can identify a fluctuating behavior in the validation curve indicating a possible instability of the model, which may cause it not to converge in different conditions. The decrease in the performance of model 4 with respect to model 3 may be caused by different factors. One of the most affecting factors is that the VGG-16 model was not created for this particular application and some layers may not be helpful for this application.



a)

```
82/82 [==============================] - 1s 15ms/step
              precision    recall  f1-score   support

           0       0.97      0.99      0.98      1300
           1       0.99      0.97      0.98      1300

    accuracy                           0.98      2600
   macro avg       0.98      0.98      0.98      2600
weighted avg       0.98      0.98      0.98      2600
```

b)

```
82/82 [==============================] - 0s 4ms/step
              precision    recall  f1-score   support

           0       0.98      0.97      0.98      1300
           1       0.97      0.98      0.98      1300

    accuracy                           0.98      2600
   macro avg       0.98      0.98      0.98      2600
weighted avg       0.98      0.98      0.98      2600
```

Figure 11. General report of performance of model VGG-16 (a)) and model 3 (b))

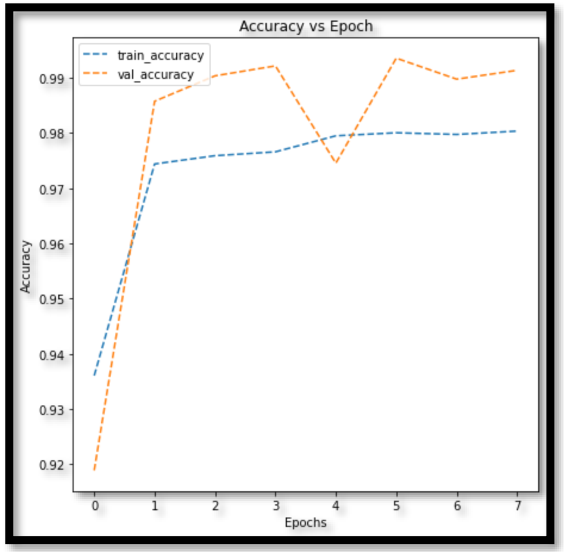
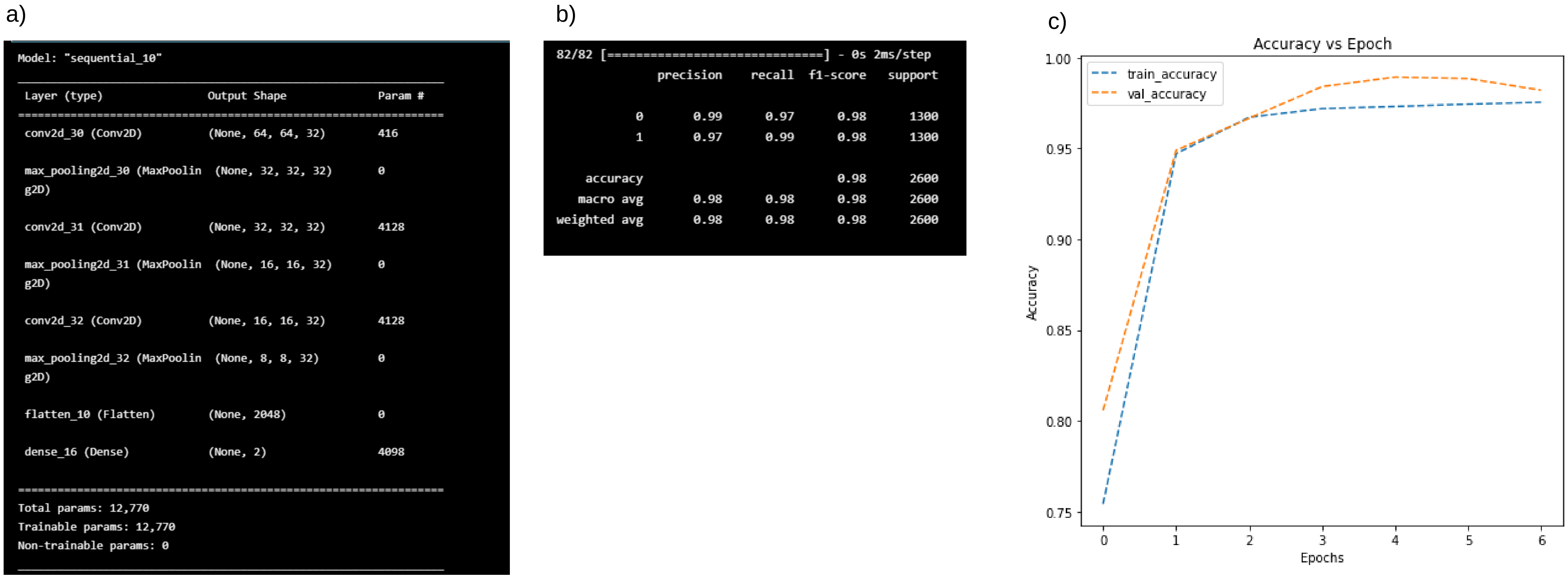


Figure 12. Evolution of accuracy as a function of epochs for validation and train data for model VGG-16

Proposal for the final solution design

Another aspect to study is the elimination of dense layers in the model, especially the ones after the Conv2D layers. The Final Model was developed based on the hypothesis that we have enough parameters with the three Conv2D layers in model 3 to have an efficient model and this could decrease computational time and prevent overfitting. In addition, the train images were modified with the Gaussian Blurring technique in order to make our model more robust. Degrading image quality as the Gaussian Blurring methodology does can help make our machine learning models more resilient to the harsh realities they will encounter in real-world situations

As shown in Figure 13 the final model shows an increase in the recall variable for parasitized predictions from 0.98 to 0.99 while maintaining a value of 0.98 in the f1-score variable, with respect to model 3. With respect to the behavior of the accuracy as a function of epochs, a behavior without abrupt fluctuations is observed, which may suggest a certain ease of convergence with respect to models with the presence of fluctuations. Finally the decrease of layers to only 3 convolutional layers with 12770 trainable parameters and the use of Gaussian Blurring in the training images (resulting in the final model), gave the best results (based on the recall variable for parasitized predictions equal to 0.99 and f1-score of 0.98).



a)

```
Model: "sequential_10"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_30 (Conv2D)          (None, 64, 64, 32)        416

 max_pooling2d_30 (MaxPoolin  (None, 32, 32, 32)       0
 g2D)

 conv2d_31 (Conv2D)          (None, 32, 32, 32)        4128

 max_pooling2d_31 (MaxPoolin  (None, 16, 16, 32)       0
 g2D)

 conv2d_32 (Conv2D)          (None, 16, 16, 32)        4128

 max_pooling2d_32 (MaxPoolin  (None, 8, 8, 32)         0
 g2D)

 flatten_10 (Flatten)        (None, 2048)              0

 dense_16 (Dense)            (None, 2)                 4098

=================================================================
Total params: 12,770
Trainable params: 12,770
Non-trainable params: 0
_________________________________________________________________
```

b)

```
82/82 [==============================] - 0s 2ms/step
              precision    recall  f1-score   support

           0       0.99      0.97      0.98      1300
           1       0.97      0.99      0.98      1300

    accuracy                           0.98      2600
   macro avg       0.98      0.98      0.98      2600
weighted avg       0.98      0.98      0.98      2600
```

c)

Figure 13. Structure (Layers) (a)), general report of performance (b)) and evolution of accuracy as a function of epochs (c)) for validation and train data relative to the final model

Limitations and Recommendations for Further Analysis

The main limitation of our neural network-based model is that they tend to require a larger volume of data to train the model. Complexity of learning for large tasks, the more a network needs to learn, the more complicated it will be to teach it. In addition, there is a natural social resistance to adopt emerging technologies versus traditional methods, especially in the health area.

For further analysis it is recommended that the VGG-16 transfer learning (ex. VGG-16) model be studied in greater depth to elucidate its advantages and limitations in this application.

Recommendations for Policy

The above analysis clearly demonstrates the development of an effective computer vision model for detecting malaria, which is able to distinguish parasitized and uninfected blood cells. The resulting model reported almost no false negatives at a very high accuracy (0.98). This model is expected to help in general and in particular children under 5 years of age, pregnant women and HIV/AIDS patients in the detection of malaria. This by standardizing the operation and quality of malaria diagnosis, which is expected to significantly reduce the economic costs and increase the accessibility of diagnosis. Current methods rely too much on professional expertise to distinguish healthy and unhealthy blood cells, in addition the process is often a tedious, time-consuming process, and the diagnostic accuracy can be adversely impacted by inter-observer variability.

It is proposed that, for the implementation of the studied model, a transition plan be carried out, where the objective is not to displace the professional detection personnel, but rather that the detection personnel together with data science professionals monitor the studied model for its validation in the field. If this first phase is successfully completed, the model could gradually become functional with minimal monitoring.

Finally, the application of models similar to the one studied in this work could be functional especially in the branch of imaging used in the health area, which is very diverse and has a very important impact on vital aspects of society such as public health and its economy.

References

1.World Health Organization
2. Great learning material
3. Joseph N., The Importance of Blur as an Image Augmentation Technique
4. Pulfer, E. (2019). Different Approaches to Blurring Digital Images and Their Effect on Facial Detection. Computer Science and Computer Engineering Undergraduate Honors Theses Retrieved from https://scholarworks.uark.edu/csceuht/66

Appendix

Confusion matrix and definitions

Parasitized	Uninfected
TP	FP
FN	TN

		Parasitized	Uninfected
Precision	=	TP/(TP+FP)	TN/(TN+FN)
Recall	=	TP/(TP+FN)	TN/(TN+FP)

Accuaracy = (TP+TN)/(TP+FP+FN+TN)

f1-score = 2*Precision*Recall/Precision+Recall