ABSTRACT

A machine learning model was used to classify malaria blood smears into either uninfected or parasitized. The model uses datasets from the Tensorflow 'malaria' dataset as its test, train, and validation sets.

TensorFlow Datasets has many datasets that can be loaded and used to learn more about image classification and various computer vision machine learning pipelines. One of the datasets hosted here is the "Malaria Dataset".

The Malaria dataset contains a total of 27,558 cell images with equal instances of parasitized and uninfected cells from the thin blood smear slide images of segmented cells.

Remark

This dataset consists of two directories, one is the parasitized cell images and the other is the uninfected cell images. Each of these directories has several images and labels. The label denotes the class to which that image belongs.

To Do

Create a model to classify the different blood smears into either parasitized or uninfected

Transform the best model into a tflite model

Use the model to correctly classify the different images selected from the image list

About

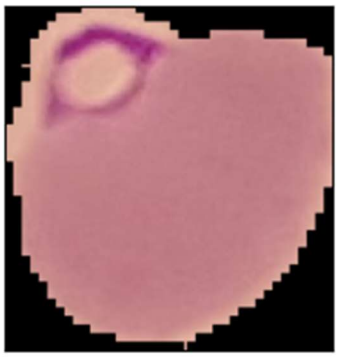
This repository hosts the Jupyter Notebook file that contains the trained model. The notebook contains two model versions both CNN models with different layers and complexities. I used the ensemble approach my decision being informed by the "Wisdom of Masses" analogy. In an ensembled model, two or more models are employed to increase the accuracy and generalization aspect of the model leveraging the strengths of the different models.

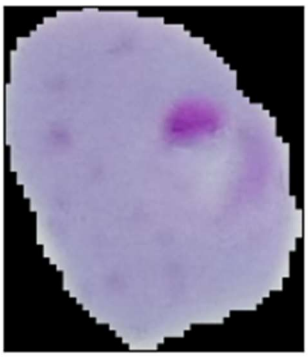
The following steps will allow you to clone the CEMA repository and run the file.

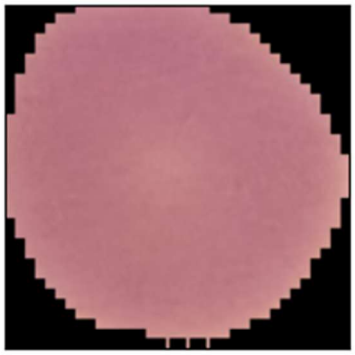
- Copy the link below: CEMA-MALARIA-REPOSITORY
- Create an empty folder.
- Navigate to your terminal preferably GIT BASH within that folder then type in git clone and paste CEMA-MALARIA-REPOSITORY
- This will allow you to clone the repository on your local drive.
- From here run the notebook in your favourite environment.
- You can load the trained model using: tf.keras.models.load_model('malaria_classification_model.h5'): In this case, malaria_classification_model.h5 is the already trained model

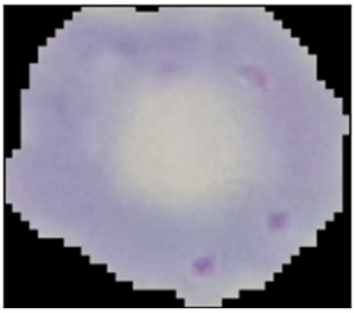
Visualized image cells from the raw dataset

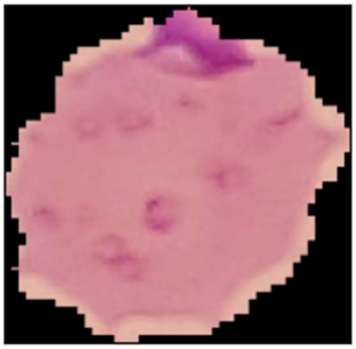

uninfected (1)


parasitized (0)


parasitized (0)

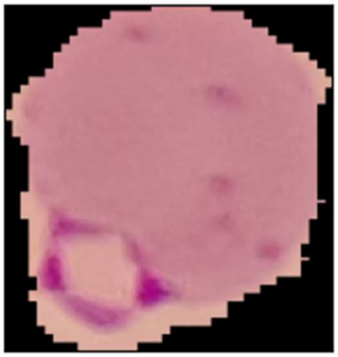

uninfected (1)


uninfected (1)

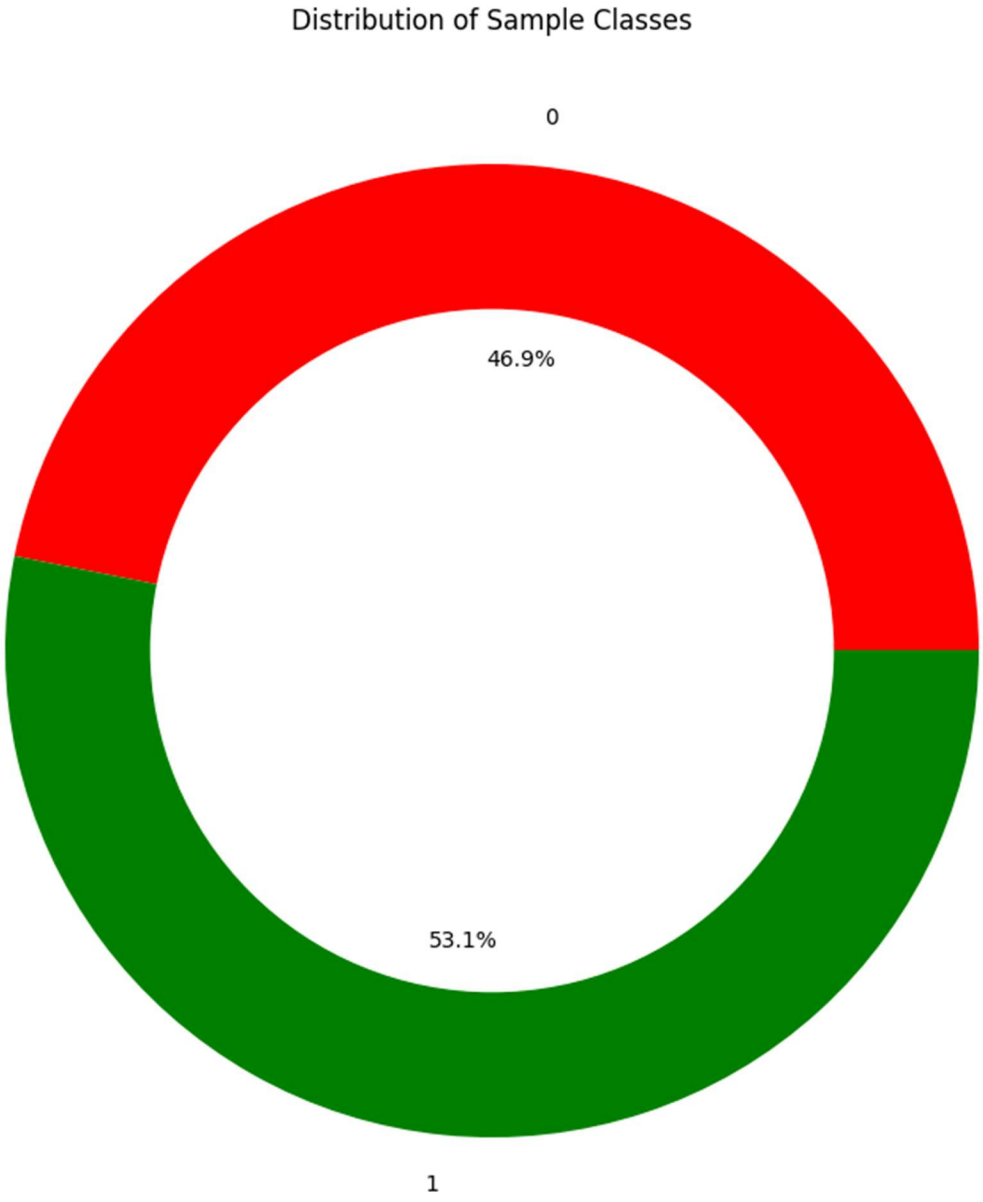

parasitized (0)


uninfected (1)


uninfected (1)


parasitized (0)

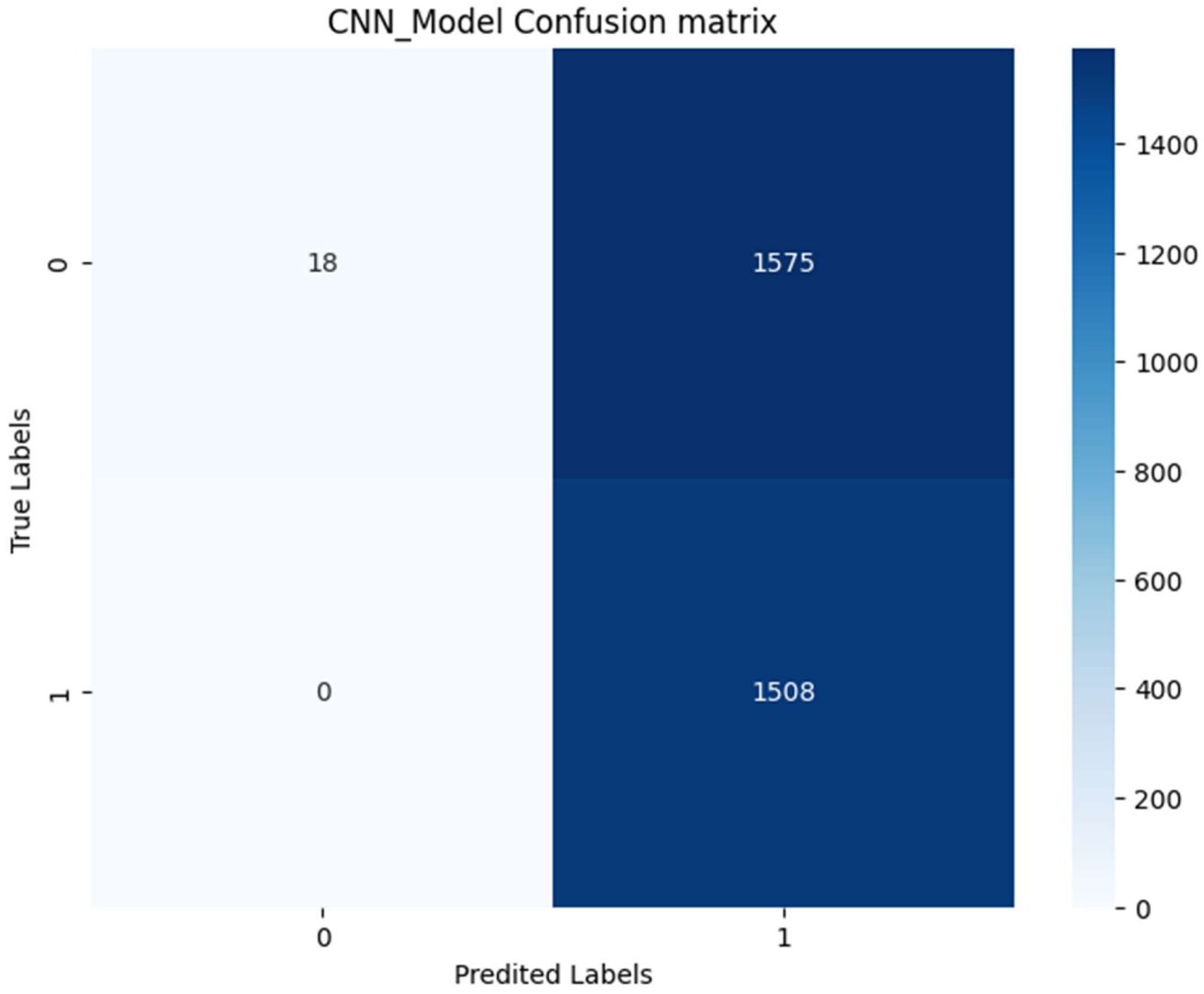
Class Distribution



There were two categories:

1. Parasitized [was appended to class 0]
2. Uninfected [was appended to class 1]

CNN model Version 1 confusion matrix (accuracy = 49.21%)



Model training output

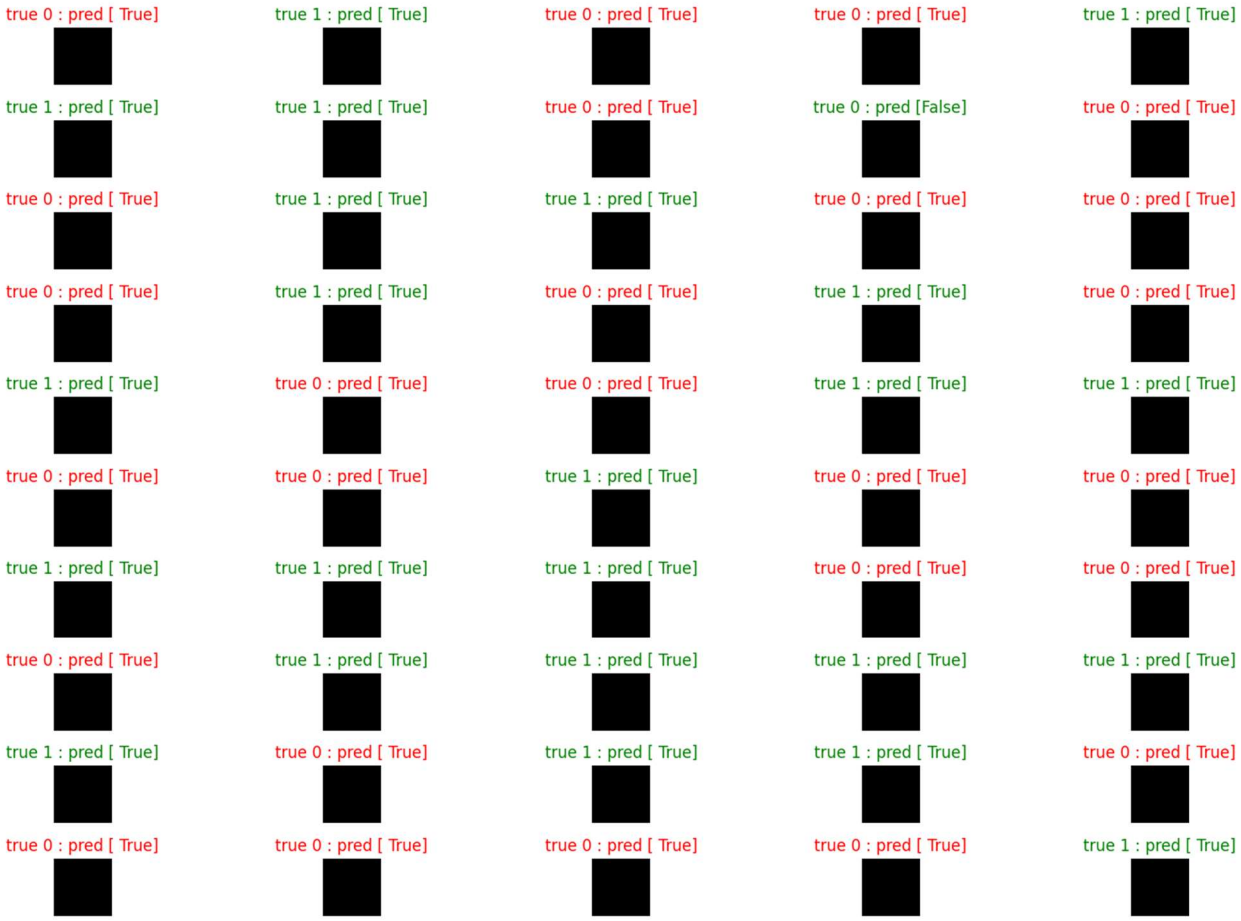
Training

```
In [61]: malaria_train_model = cnn_model.fit(
             X_train, y_train,
             epochs=no_epochs,
             verbose=1,
             validation_data=(X_val, y_val),
             callbacks = [checkpoint_cb, early_stopping_cb, lr_scheduler]
         )
```

```
Epoch 1/5
467/467 [==============================] - 1843s 4s/step - loss: 0.6802 - accuracy: 0.5998 - val_loss: 9.4372 - val_accurac
y: 0.4920 - lr: 0.0100
Epoch 2/5
467/467 [==============================] - 1648s 4s/step - loss: 0.6447 - accuracy: 0.6432 - val_loss: 1.0229 - val_accurac
y: 0.5224 - lr: 0.0089
Epoch 3/5
467/467 [==============================] - 1620s 3s/step - loss: 0.6399 - accuracy: 0.6468 - val_loss: 0.6956 - val_accurac
y: 0.5303 - lr: 0.0079
Epoch 4/5
467/467 [==============================] - 1644s 4s/step - loss: 0.6330 - accuracy: 0.6520 - val_loss: 0.6922 - val_accurac
y: 0.5080 - lr: 0.0071
Epoch 5/5
467/467 [==============================] - 1650s 4s/step - loss: 0.4404 - accuracy: 0.7941 - val_loss: 1.1794 - val_accurac
y: 0.4962 - lr: 0.0063
```

As seen, the accuracy and prediction are not very accurate. The images below sampled from 50 images will attest that the model did not work correctly and did have an overfitting problem. The output of the model is seen below:

Conclusion: Out of 50 samples, 27 of them were incorrectly predicted. Therefore, there was a need to retrain the model with different parameters.


true 0 : pred [True]
true 1 : pred [True]
true 0 : pred [True]
true 0 : pred [True]
true 1 : pred [True]
true 1 : pred [True]
true 1 : pred [True]
true 0 : pred [True]
true 0 : pred [False]
true 0 : pred [True]
true 0 : pred [True]
true 1 : pred [True]
true 1 : pred [True]
true 0 : pred [True]
true 0 : pred [True]
true 0 : pred [True]
true 1 : pred [True]
true 0 : pred [True]
true 0 : pred [True]
true 1 : pred [True]
true 1 : pred [True]
true 0 : pred [True]
true 0 : pred [True]
true 1 : pred [True]
true 0 : pred [True]
true 0 : pred [True]
true 1 : pred [True]
true 1 : pred [True]
true 1 : pred [True]
true 0 : pred [True]
true 0 : pred [True]
true 0 : pred [True]
true 1 : pred [True]
true 1 : pred [True]
true 1 : pred [True]
true 1 : pred [True]
true 1 : pred [True]
true 0 : pred [True]
true 1 : pred [True]
true 1 : pred [True]
true 1 : pred [True]
true 0 : pred [True]
true 0 : pred [True]
true 0 : pred [True]
true 0 : pred [True]
true 1 : pred [True]