MALARIA

Introduction

Malaria is an infectious disease cautious by protozoan parasites of the genus plasmodium and it is a major global health threat.

The standard way of diagnosing malaria is by visually examining blood smears for parasite-infected red blood cells under the microscope by qualified technicians.

This method is inefficient, and the diagnosis depends on the experience and the knowledge of the person doing the examination, hence Convolutional Neural network gives the best output. The aim of the collection of the dataset was to reduce the burden for microscopists in resource-constrained regions and improve diagnostic accuracy using an CNN algorithm to detect and segment the red blood cells.

Why CNN?

The Convolutional Neural Networks (CNNs) have been used to classify malaria parasites from blood smear images automatically and successfully gave a good result.

The use of CNN as a feature extractor shows better performance than transfer learning.

Data source:

We used archived images acquired from Kaggle Malaria classification dataset where it has nearly 28000 train and test data.

Malaria Cell Images Dataset (kaggle.com)

SPECIFICATIONS:

IDE: Google Collab

GPU: 2.8.0

Important libraries: TensorFlow, keras, matplotlib, streamlit, h5py, pandas, NumPy,

Deployment: Streamlit, VS code

Accuracy with different epochs value:

EPOCHS	ACCURACY (%)	LOSS (%)	VALIDATION ACCURACY (%)	VALIDATION LOSS (%)
2	93.66	19.61	93.27	20.47
5	93.97	18.61	93.56	18.51
10	91.56	17.02	93.77	16.99
12	91.11	16.89	93.98	16.01

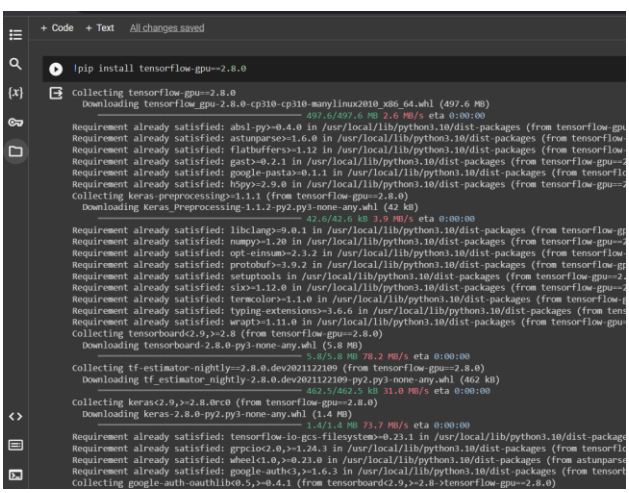
Optimiser:

The optimiser that is used is called Adaptive Moment Estimation (Adam) which combines ideas from both RMSProp and Momentum. It computes adaptive learning rates for each parameter.

EPOCHS:

```
Epoch 1/2
<ipython-input-13-fdda96df0b73>:1: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future version. Please use `Model.fit`, which supports generators.
  history = model.fit_generator(generator = train_datagen,
689/689 [==============================] - 385s 558ms/step - loss: 0.2228 - accuracy: 0.9298 - val_loss: 0.2047 - val_accuracy: 0.9332
Epoch 2/2
689/689 [==============================] - 386s 560ms/step - loss: 0.1961 - accuracy: 0.9366 - val_loss: 0.2072 - val_accuracy: 0.9327
```

GOOGLE COLLAB:



VISUAL STUDIO:

streamlit(app).py 6 | #Title Untitled-1 9+ | img = get_img_as_base64("image.jpg") Untitled-2 1 | streamlit.ipynb

C: > Users > user > Desktop > streamlit(app).py > ...

```python
#importing libraries
import streamlit as st
import tensorflow as tf
import numpy as np
from PIL import Image, ImageOps

#Title
st.write("""
        # Malaria Cell Classification
        """
        )

#upload file
upload_file = st.sidebar.file_uploader("Upload Cell images", type = "png")

#side button
Generate_pred = st.sidebar.button("Predict")

#importing the model
model = tf.keras.models.load_model("C:/Users/user/Downloads/malaria_det.h5")

#conditions
def import_n_pred(image_data, model):
    size = (128,128)
```

DEPLOYMENT:

CONCLUSION:

Malaria cell image classification uses deep learning faster than the most traditional ML models. With Streamlit, the prediction results are presenting good results.