CA2 Project Report l00177814

A Machine Learning Pipeline to Categorise Cells Parasitised by Malaria

Introduction to MLOps

MLOps or machine learning operations is a core function of Machine Learning. This aspect of machine learning is focused on streamlining the processes related to training, deploying and maintaining machine learning models.

Directed Acyclic Graphs are a powerful and prevalent tool used in MLOps. The paper (Tennant *et al.*, 2021) searched through health research papers published between 1999 to 2017 where 234 articles were identified that utilised DAGs. DAGs are Acyclic which means they do not create a loop and flow from start to finish.

As stated in the paper (Symeonidis *et al.*, 2022) MLOps is still a relativity new field which builds on the existing field of DevOps, the main components of which are Continuous Integration(CI) and Continuous Delivery(CD). The need to apply the ability to repeatedly update and improve code lead to the creation of MLOps as a way of applying DevOps within machine learning models(Alla and Adari, 2021) .

The best known description of MLOps is the ToughWorks model. This model focuses on the three components of data collection, selection and preparation for a ML model (*MLOps Challenges in Multi-Organization Setup: Experiences from Two Real-World Cases | IEEE Conference Publication | IEEE Xplore*, no date).

The review paper, (Kreuzberger, Kühl and Hirschl, 2023)has listed some of the most utilised Technologies used in both the open source and commercial spaces. The technologies presented and used in this project were open source and so those are the technologies that will be reviewd and discussed in this report.

Description of the Machine Learning Pipeline implemented

The pipeline implemented in this project was based on Docker images and Docker Compose. This was chosen as custom python docker images can be updated and published to a public or secure docker repository for access remotely. This allows for images to be created and tested locally before being published. If this images are published with the latest tag it will update the pipeline on startup. Docker Compose was chosen to manage the services as it allows for future integration with Airflow as the YAML file that dictates the running order of services and volume or mount creation can be used to generated Airflow DAGS. A dataset was chosen from Kaggle and download in the first service.

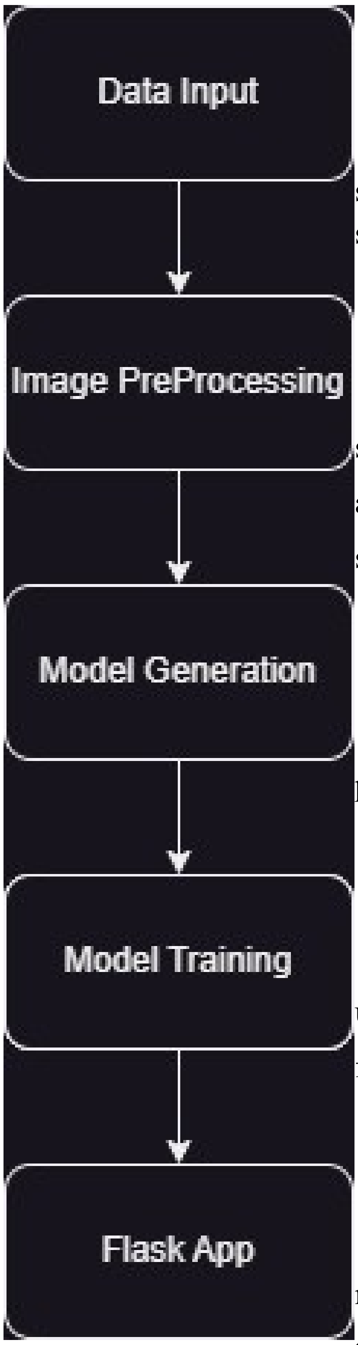

Figure 1: MLOps Pipeline

Data processing

The data that is downloaded from Kaggle is zipped in a file. This service extracts this file and saves it to the mount to be used by subsequent services.

Image Preprocessing

This service takes the file from the previous service and performs a series of manipulations that skew, rotate, zoom and slide the image to avoid over fitting the model on the existing training set. This service saves these files to the local storage.

Model Generation

A multilayer convoluted machine learning model was created and pickled for use in the training service.

Model Training

The model that was generated in the previous service is opened unpickled and trained on the training dataset that was generated in the first service

Displaying Results on Flaskapp

The final trained model was pickled and saved this allowed for the model to be used in the flask app that was created as the last service in the pipeline. Using a flask app allowed users to test the accuracy of the model by selecting images from the local directory that are diagnosed and gives the confidence of the model.

Discussion and Analysis of techniques used

Docker Containerization

Customer docker images were created that contained the code needed to complete the services object in its Python file. The code needed to set up the docker image in the Dockerfile and a list of requirements in a requirements.txt file. These custom images are built from the base image python:3.6.

The environment variables required as authentication for Kaggle were not saved in the published images but the environment variables of the Docker file. This was done to keep these authorisation token values confidential.

Publishing the images allows for the deployment of the model to a virtual machine, server or other local machine with only the Docker Compose file.

Containerization is a powerful tool as it allows for individual services to be updated

Docker Compose

A Docker Compose YAML file was created which listed the service that were required to run. There was a condition set that a service would not run until the previous service completed successfully.

It was chosen to use a Docker compose file as an airflow Docker image could be created to use the Docker YAML file to create and display an Airflow DAG. Using Docker Compose allowed for the environment variables to be saved locally and remain secure even when the required images are built and published.

As the dataset selected is quite large it is downloaded and saved locally and the location of the download is mounted as a bind to the service outlined in the YAML to all service to act as a shared data location that is not constrained by the size limits of a Docker Volume.

Tensorflow

Tensor flow is an open source machine learning platform. It has been mentioned in several papers and listed in review papers as a popular option for machine learning and MLOps.

In this implementation the image data generator, Sequential model, and Keras layers tools were used. The ImageDataGenerator library is used to preprocess images for machine learning models. In this implementation the images were rescaled, rotated, sheared and flipped. This provides new training data to the model and helps to prevent overtraining of the model.

The sequential model libraries and Keras layers were used to create the model with a number of convoluted layers and a single binary layer as the classifications in this project are binary; parasitised or non-parasitised. The model is pickled to allow it to be saved to the local mount for use in further services.

Flask

The flask app created allows for the upload of individual images which will be categorised using the trained model. The output shown on the flask app will be the category of the image as classified by the model and the models confidence will be displayed as number between 0 and 1. The divide between the classifications has been set at .5 so the closer an images classification is to .5 to less certain the algorithm is that a given image belongs to one classification or the other.

Pickle

Pickling in Python is a method of serializing data (*pickle — Python object serialization*, no date) that allows for data to be saved and stored this is particularly useful with MLOps as the trained model can be saved and accessed from a web app.

Conclusions and future works

For future improvements and developments of this project I would like to create a Docker Airflow image that would control the pipeline and display further information such as the DAG and running services. In an Airflow image the pipeline can be scheduled. Scheduling a pipeline can keep a model up to date as further information is gathered keeping a model up to date and increasing its accuracy by providing the most recent data.

The pipeline created has a long run time that I would like to work on reducing.

Bibliography:

Alla, S. and Adari, S.K. (2021) ‘What Is MLOps?’, in S. Alla and S.K. Adari (eds) *Beginning MLOps with MLFlow: Deploy Models in AWS SageMaker, Google Cloud, and Microsoft Azure*. Berkeley, CA: Apress, pp. 79–124. Available at: https://doi.org/10.1007/978-1-4842-6549-9_3.

Kreuzberger, D., Kühl, N. and Hirschl, S. (2023) ‘Machine Learning Operations (MLOps): Overview, Definition, and Architecture’, *IEEE Access*, 11, pp. 31866–31879. Available at: https://doi.org/10.1109/ACCESS.2023.3262138.

MLOps Challenges in Multi-Organization Setup: Experiences from Two Real-World Cases | IEEE Conference Publication | IEEE Xplore (no date). Available at: https://ieeexplore.ieee.org/abstract/document/9474388 (Accessed: 5 June 2023).

pickle — Python object serialization (no date) *Python documentation*. Available at: https://docs.python.org/3/library/pickle.html (Accessed: 5 June 2023).

Symeonidis, G. *et al.* (2022) ‘MLOps - Definitions, Tools and Challenges’, in *2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC). 2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC)*, pp. 0453–0460. Available at: https://doi.org/10.1109/CCWC54503.2022.9720902.

Tennant, P.W.G. *et al.* (2021) ‘Use of directed acyclic graphs (DAGs) to identify confounders in applied health research: review and recommendations’, *International Journal of Epidemiology*, 50(2), pp. 620–632. Available at: https://doi.org/10.1093/ije/dyaa213.

Libraries used

tensorflow

opencv-python

pickle-mixin

kaggle

zipfile36

subprocess.run

split-folders

pathlib

pytest-shutil

flask

Pillow