LiveEO MLOps Engineer Challenge

In this task you will receive a trained computer vision model with the necessary code to run it. Also you will receive some test images to run inference. This is for you to check the output and the necessary requirements to run the code. The task for you will be to set up a docker container and a continuous integration/development pipeline on a bare git repository.

**ML Code**

In this attached [folder](https://drive.google.com/file/d/1_K4SzsgbaoW8kuQvyFG-Pkcnj__3S73s/view?usp=share_link), you will find all needed code and the dataset for this challenge.

**Folder Structure**

Once you download and unzipped the folder, you will see the main ***LiveEO\_MLOps\_Challenge*** folder which comprises 3 folders and 2 files:

1. “***scripts***”: Contains data loader, models and main file that are required to run this code.
2. “***test\_images***”: Contains 2 sub-folders that contain “*images*” and “*labels*” for a test run. These images can be used to execute the test run for this code.
3. “***trained\_models***”: Contains the 2 trained model checkpoints for this model. For the main task you will only need to use 1. For one of the optional tasks you might use 2 (described later)
4. “***README***”: Readme file contains all necessary information and guidance that you need in case you want to run and explore the code.
5. “***requirements***.***txt***”: Requirements text file contains all necessary packages that need to be installed. You can add or change this file depending upon the docker you want to create.

The test images can be opened and explored in the [QGIS](https://www.qgis.org/en/site/) software package or locally if your system supports tiff files, but it’s not strictly necessary to complete the challenge. With python you can also use the “*rasterio”* package to explore the tiff files.

For this challenge, the imagery georeferencing or model performance isn’t important, please focus more on operations with docker and CI/CD pipeline.

**Tasks**

Once you get the data, you will have to perform the following steps:

1. Explore the code and understand how to run inference on the test images *(recommended time 1-2h)*
2. Set up the docker image that can be built, use it to execute inference test runs and save the outputs from the model. *(recommended time 2-4h)*
3. Set up a new git repository (hosted on github, gitlab, or other service of your choice - however please ensure the repo is private) and push the code there into the master branch *(recommended time 20min)*
4. Set up the CI/CD pipeline with features of your choice on the repo and make sure it runs as expected when pushing to the master branch. *(*r*ecommended time 1-2h)*
5. Prepare a report (r*ecommended time 1h)*

You are free to choose any possible way to create docker images, add more functionalities and features that you prefer for the CI/CD pipeline. The only condition is that all needs to eventually end up on the git repository.

You can also do some optional tasks based on your interest:

1. Model serving with docker on your preferred platform
2. Setting up evaluation metrics and displaying results using Docker
3. Model tagging / versioning with different checkpoints
4. Selecting the best performing model checkpoint automatically with docker output and ci/cd pipeline. (if you set up evaluation metrics)

As your solution, please send us a link to the repository which includes:

* Docker file that is ready to use
* CI/CD yml file
* Jupyter notebook or text/word file that explains:
  + Instructions for us how to reproduce and build the docker image and test the CI/CD pipeline.
  + What has been done and why certain methods are chosen.
  + Future possible features and functionalities that can be added

The overall goal here is to be creative with CI/CD pipeline and explanation of possible features which can be added in the future. Overall, we care more about how you approach the problem, rather than the actual results.

If you have any questions, feel free to contact [akash.sachdeva@live-eo.com](mailto:axel@live-eo.com)