Dear applicant, below you can see 2 different tasks that we propose to you as a part of the technical interview. Please choose only one of them according to your preferences.

Task 1 - Nutrient Deficiency:

The special dataset is available at:
https://registry.opendata.aws/intelinair_longitudinal_nutrient_deficiency/

The dataset contains 386 field imagery data and the corresponding mask that shows the areas in the field that has nutrient deficiency. The aim is to predict the nutrient deficiency mask using the input imagery. Feel free to read the associated paper, that you can find in the given list.

Task 2 - Field State Classification:

The special dataset is available at s3 public bucket:
s3://intelinair-internship/field_state_classification/

The dataset various flight imagery and the task is to predict the field state given the imagery.
The true labels you can find in the file:
s3://intelinair-internship/field_state_classification/final_data.CSV
In the file, you should use `flight_code` and `ActualStatus` columns.

What do we expect to receive as a solution?

- *A clearly organized GitHub repository (.py files, you can use JupyterNotebooks for visualization purposes, but all the core code must be in .py files). The code should be written using Python3.8 and according to the pep8 standards.*
- *The code should contain separate parts for data processing and training of the DeepLearning models.*
- *Since the data is big and not all people have access to GPU's we do not expect to see high accuracies or something like that, we want to see a working pipeline (you can train a model on your machine only 1 - 2 epochs, just for being sure that there are no bugs in the code) and different purposed ideas.*
- *The code should have also an inference script that should use an already trained model and a new sample for making a prediction.*
- *Be prepared for the discussion of the approaches and code.*
- *It would be nice to make a repo as a pip installable package (not on PyPi, we can install also direct from GitHub using pip).*
- *It would be nice to see the DeepLearning part coded using PyTorch library, but if you feel that you can do better if choose another library, feel free to choose your preferred one.*