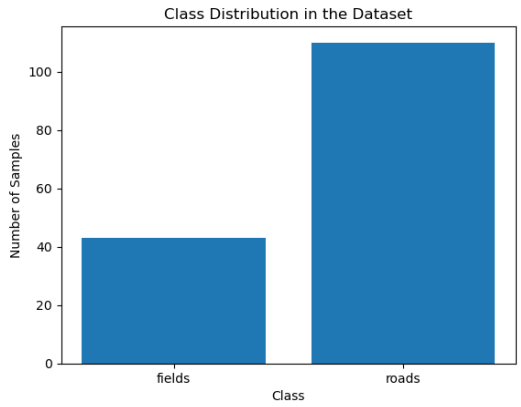
Technical Assessment

Process for completing the test

First of all I tried to manipulate in a Notebook file (that I submitted) the dataset and understand how it was constructed, I immediately understood that the dataset was imbalanced as you can see below:



But to better understand the consequences of this kind of dataset, I did a training in order to evaluate the performances using the accuracy metric and the resnet18 model. Just as I expected, the accuracy on the test set was high due to the difference in sample content between classes.
We can't trust the accuracy in this case, so I tried other metrics that are more representative in image classification such as recall, precision and f1_score and by these metrics I understood that my model had bad performances.

To handle the consequences of the imbalanced dataset I used two methods:

- the data augmentation techniques such as flip, rotation and color
- the class weight technique by using $weight\ class = \frac{total\ sample}{class\ sample}$ for each classes

and I created the Data loaders for my training and testing set.

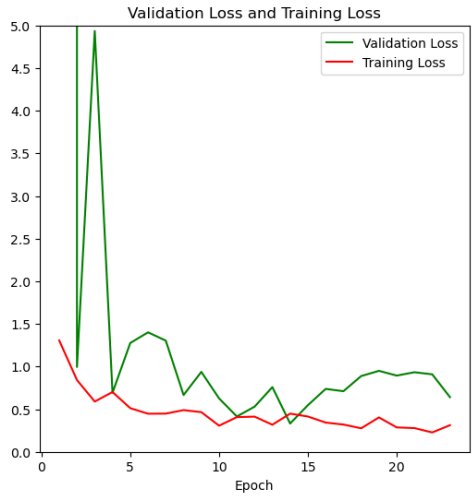
I compared 3 models Resnet18, Efficientnetb0 and Vision transformers vit_b_16, and I realised that the complexion the model, the more risk of overfitting causes the dataset is small and simple. Resnet18 was both less complex and not too simple, so I chose to use it with pretrained weights.
In my training process to reduce the risk of overfitting I used two techniques:

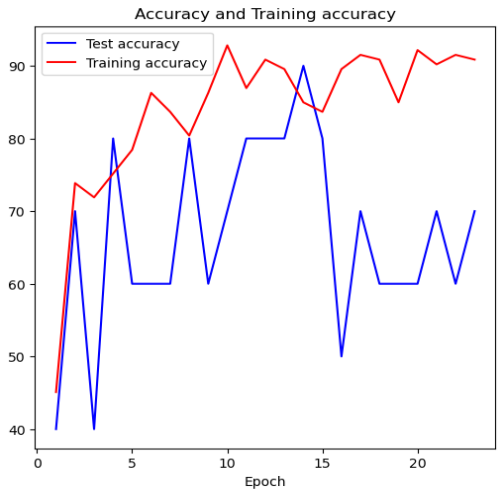
- the dropout
- the learning rate scheduler

I also used CrossEntropy as the loss function cause it's popular in classification task, I used it with the class weight explained above to handle the imbalanced data.

I tried manually differents hyperparameters to train my model but the results aren't good enough so in order to choose the best hyperparameters such as optimizers, dropout, leaning rate, scheduler, batch_size et number of epoch , i used an hyperparameter tuning tools called 'Optuna' where i prioritize the f1 score as metrics to maximize.

Below are my training and testing curves.





As you can see there is no overfitting and the model seems generalize well, since the validation loss and training loss decrease according to the same dynamic while the training and testing accuracy increase.

the model has good performances for the best epoch as you could see below or in the Notebook on github :

- Test dataset accuracy: 90.000%
- Precision: 92.000%, Recall: 90.000%, F1 Score: 90.101%