Documentation Task 3

Students

- Elisa Gilbert (3780947)
- Jan-Niklas Forster (3781088)

Project Structure

```
.
  dat/  # Default directory for dataset
  doc/  # Assignment documentation
  hpc/  # Slurm-scripts for high performance computing
  results/  # Results task 2
     task2_finetuned/  # Fine-tuned model for hand-in (Task 2)
         best_model.pt  # Trained model from the two augmentations
         test_logits_best_model.npy  # Saved logits on test split
         training_history_**.png  # Training history for each augmentation
         training_results.json  # JSON dump for training history on both augmentation
         test/  # Directory which is created when reproduction script is run
     task3_finetuned/  # Fine-tuned model for hand-in (Task 3)
         best_model.pt  # Trained model from the two augmentations
         test_logits_best_model.npy  # Saved logits on test split
         training_history_**.png  # Training history for each augmentation
         training_results.json  # JSON dump for training history on both augmentation
         test/  # Directory which is created when reproduction script is run      run_YYYY
  src/  # Source code for task 1 to 3
      task1/
      task2/
      task3/
      util/
```

How to run the code

Getting Started

- Create a virtual environment python -m venv .venv
- Activate virtual environment source .venv/bin/activate
- Install requirements pip install -r requirements.txt
- Download zipped data files (EuroSAT_MS.zip and EuroSAT_RGB.zip) to a DAT_DIR directory of your choice. This DAT_DIR directory must be provided as argument when running the script (see below). This repository provides a default DAT_DIR at /dat.
- Set working directory to project root

Fine-Tune model

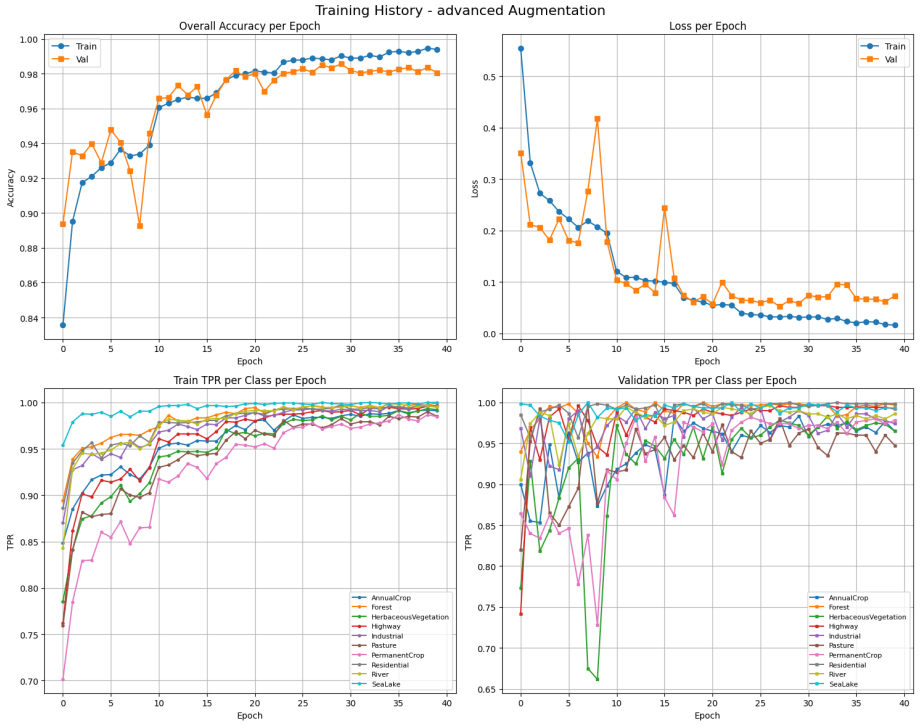
You can fine-tune ResNet50 model for TIF data. When script is run, a new directory named `run_YYYYMMDD_hhmmss` is created and hyper parameters from `constants.py` are used for training. The corresponding channels, that were used, are defined in `constants.py` as well.

```
python -m src.task3.fine_tune -D <ABSOLUTE_PATH_TO_DAT_DIR_>
```

Reproduce results Load the saved model at `./results/task3_finetuned/best_model.pt`. Runs predictions on model on test split and prints results to terminal.

```
python -m src.task3.reproduce -D <ABSOLUTE_PATH_TO_DAT_DIR_>
```

Training Performance



Test Performance


```
{
    "overall_accuracy": 0.9837037037037037,
    "per_class_accuracy": [
        0.9616666666666667,
        0.9983333333333333,
        0.9816666666666667,
```

```
        0.992,
        0.962,
        0.9575,
        0.984,
        1.0,
        0.994,
        0.9966666666666667
    ]
}
```

Task 3 Explaination

Model

This project has fine-tuned a custom late-fusion model, based on the ResNet50 model on image classification on the multispectral images of the EuroSAT Dataset. The model is provided as `.pt` file in `results/task3_finetuned/best_model.pt`. Late-fusion was achieved by utilizing two ResNet50 models. The final classification layers were removed and a custom classifier was introduced.

```
self.fusion_classifier = nn.Sequential(
            nn.Linear(4096, 1024),
            nn.ReLU(),
            nn.Dropout(0.3),
            nn.Linear(1024, num_classes),
        )
```

A custom `forward` function receives the 6 choosen layers and forwards them to the corresponding modelbranch in the beginning.

Normalization

In order to normalize the multispectral images, the values of the training set are read, the `mean` and `std` are calculated, and saved in `results/run_YYYYMMDD_hhmmss/ms_stats.json`. When loading the images for fine tuning or prediction they are normalized using `NormalizeMultiChannel`.

Means and Stds of Channels

```
{
  "mean": [
    0.02066490612924099,
    0.01705300435423851,
    0.01589866168797016,
    0.014462830498814583,
    0.018317895010113716,
    0.030530009418725967,
    0.03615779057145119,
```

```
    0.035059064626693726,
    0.011179571971297264,
    0.00018502790771890432,
    0.027794938534498215,
    0.01707024872303009,
    0.03960534557700157
  ],
  "std": [
    0.0037437360733747482,
    0.0050926427356898785,
    0.006034678313881159,
    0.00906955637037754,
    0.008655871264636517,
    0.013093757443130016,
    0.016497185453772545,
    0.016990570351481438,
    0.006172210909426212,
    7.190550968516618e-05,
    0.015275572426617146,
    0.011570730246603489,
    0.01870332658290863
  ]
}
```