

Documentation Task 2

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_YYYYMMDD_hhmmss/ # New run of fine-tuning script
├── 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

Task 1

Unzip raw files and split data into train, test, val set. Split files and classname-index associations are saved to `DAT_DIR`.

```
python -m src.task1.split_data -D <ABSOLUTE_PATH_TO_DAT_DIR>
```

Task 2

This project has fine-tuned [ResNet50](#) model on image classification on the [EuroSAT Dataset](#). The model is provided as `.pt` file in `results/task2_finetuned/best_model.pt`.

Reproduce results

Load the saved model at `./results/task2_finetuned/best_model.pt`. Runs predictions on model on test split and prints results to terminal. Creates images for top/bottom 5 best scoring models for 3 classes.

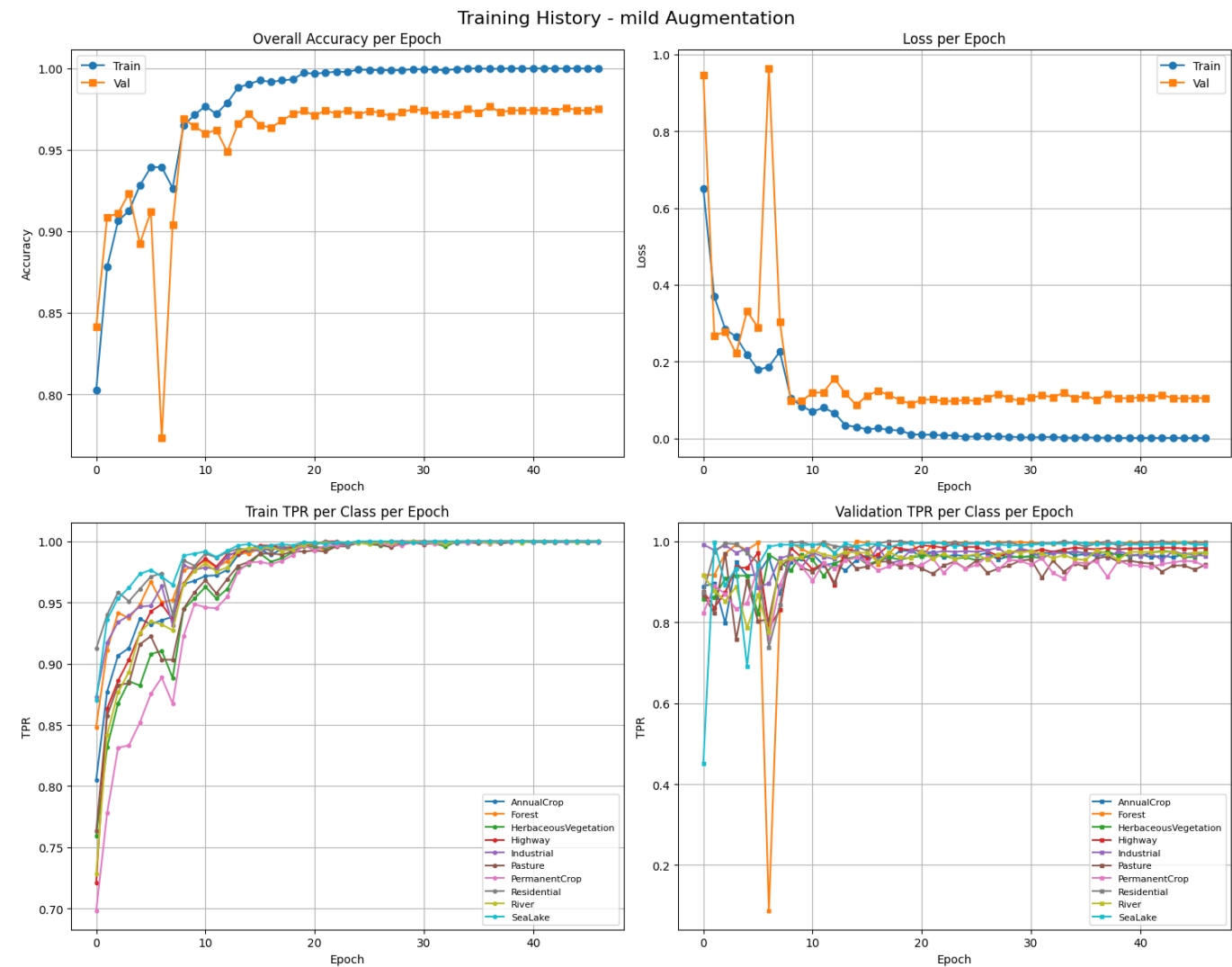
```
python -m src.task2.reproduce -D <ABSOLUTE_PATH_TO_DAT_DIR>
```

Fine-tune your own model

You can fine-tune ResNet50 model for RGB data. When script is run, a new directory named `run_YYYYMMDD_hhmmss` is created and hyper parameters from `constants.py` are used for training.

```
python -m src.task2.fine_tune -D <ABSOLUTE_PATH_TO_DAT_DIR>
```

Training Performance



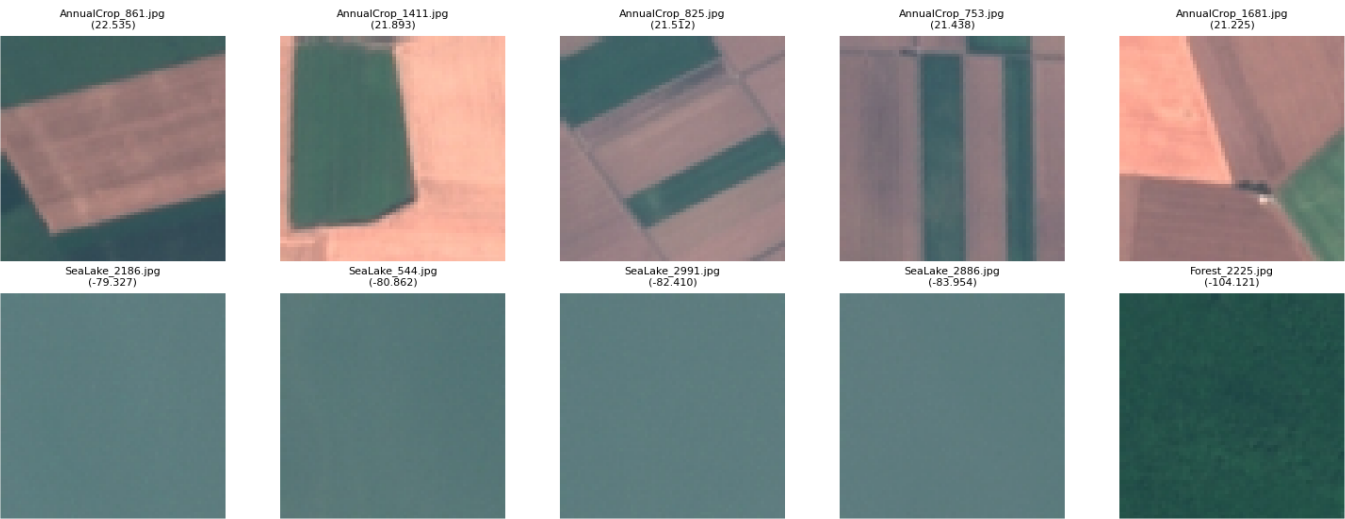
Test Performance

```
{
  "overall_accuracy": 0.9729629629629629,
  "per_class_accuracy": [
    0.9716666666666667,
    0.9916666666666667,
    0.98,
    0.978,
    0.96,
    0.955,
    0.922,
    0.9966666666666667,
    0.958,
    0.9983333333333333
  ]
}
```

Top & Bottom 5 Images of Classes 0 to 2

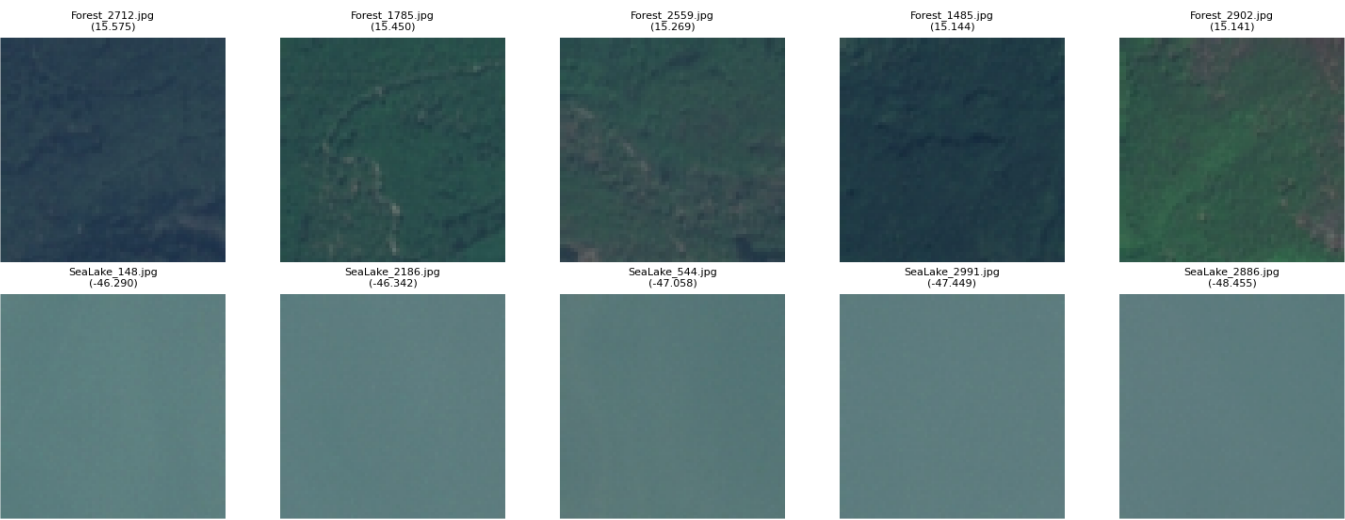
0 - Annual Crop

Top-5 (left) and Bottom-5 (right) Images for AnnualCrop



1 - Forest

Top-5 (left) and Bottom-5 (right) Images for Forest



2 - Herbaceous Vegetation

Top-5 (left) and Bottom-5 (right) Images for HerbaceousVegetation

