Correlation Analysis between two Networks and Prediction for the Future Impaction

Mengrui Mao

Abstract

This project aims to predict the features of land cover imageries with the model trained with EuroSAT dataset, with the functionality of TorchGeo. To extract the features, multispectral and geospatial transforms are applied to extend the dataset. Through AI process with PyTorch Lightning, the designed model is trained, validated, and tested with EuroSAT dataset, which finally predicts the features of land cover images.

Keywords

TorchGeo, multispectral, geospatial transforms, PyTorch Lightning

1. Introduction

For decades, earth observation satellites, aircraft, and more recently UAV platforms have been collecting increasing amounts of imagery, utilized in land cover mapping (semantic segmentation), deforestation and flood monitoring (change detection), glacial flow (pixel tracking), hurricane tracking and intensity estimation (regression), and building and road detection (object detection, instance segmentation).

By leveraging recent advancements in deep learning architectures like PyTorch Lightning applied in this project, classification and segmentation could be automatically labeled for the tremendous amount of observation imagery. Lightning is used to simplify the design of trainer model which is essential for both the training process and the predict representation. Especially, in this project EuroSAT100 dataset is to train models which predict land cover classes.

For TorchGeo models, there are pre-trained weights listed in model documentation, including Landsat, Sentinel-1, Sentinel-2 and Other Data Sources. While some weights only accept RGB channel input, some weights have been pretrained on Sentinel 2 imagery with 13 input channels and can hence prove useful for transfer learning tasks involving Sentinel 2 data. This project applied "ResNet50_Weights.SENTINEL2_ALL_MOCO".[1]

There are several geospatial transforms implemented before the AI process. Different from other torch vision applications, random transforms are not beneficial, while multispectral transforms and CRS transforms are effective and efficient. The final prediction represents the features of the original imagery, and the format of the representation could be improved and extended in future research.

2. Literature Review

In the current Network Analyst of Esri, there are three applications: Detect Objects Using Deep Learning tool, Classify Pixels Using Deep Learning tool, and Classify Objects Using Deep Learning tool. The deep learning toolset includes the model types: Connect Net, Feature classifier, MaskRCNN, Multi Task Road Extractor, Single Shot Detector, U-Net. Besides, transforms for data augmentation of training and validation datasets is specified in a Json file as input.

Recent large-scale efforts, such as the creation of a global 10 m resolution land cover map or the creation of global 30 m forest maps, pair the huge amount of available remotely sensed imagery with modern GPU accelerated models.[2] TorchGeo is applied in several papers as a method of facilitating research and managing the complexities of geospatial data with deep learning architectures. TorchGeo is a Python library for integrating geospatial data into the PyTorch deep learning ecosystem. TorchGeo is used to create reproducible benchmark results on existing datasets and benchmark our proposed method for preprocessing geospatial imagery on the fly.

3. Data and Methodology

The dataset applied in this project is subset of EuroSAT and LandCover.AI.

EuroSAT is a classification dataset, which is composed of 27,000 64 × 64-pixel Sentinel-2 images covering 13 spectral bands and 10 target classes [3]. The train/val/test splits is defined in Neumann et al. [4]. EuroSAT100 which is subset of EuroSAT contains only 100 images and maintains the same file structure, classes, and train-validation-test split. Each class has 10 images (6 train, 2 validation, 2 test), for a total of 100 images.

LandCover.ai is a semantic segmentation, which is with high resolution (0.5 m/px and 0.25 m/px) RGB aerial imagery from 41 tiles over Poland where each pixel has been classified as one of five land cover classes [5]. The scenes are divided into 10,674 512 × 512-pixel patches and split according to the script on the dataset. [2]

Spectral Indices are widely used in the Remote Sensing community. TorchGeo provide multispectral functions and geospatial transforms to compute popular indices used in remote sensing for analyzing raw imagery, based on a ready-to-use curated list of Spectral Indices for Remote Sensing applications with spectral-indices-table . The following figures, Figure 2 & Figure 3 & Figure 4, are the transforms implemented in this project.

Figure 1: original sample with True Color (RGB)

The figures below are computed with Normalized Difference Vegetation Index (NDVI), Normalized Difference Water Index (NDWI), Normalized Difference Built-up Index (NDBI).

$NDVI = \frac{NIR-R}{NIR+R}$, where Near Infrared (NIR) and Red bands calculate in a value between [-1, 1]. Low NDVI values represents no or unhealthy vegetation and high NDVI values represents healthy vegetation.

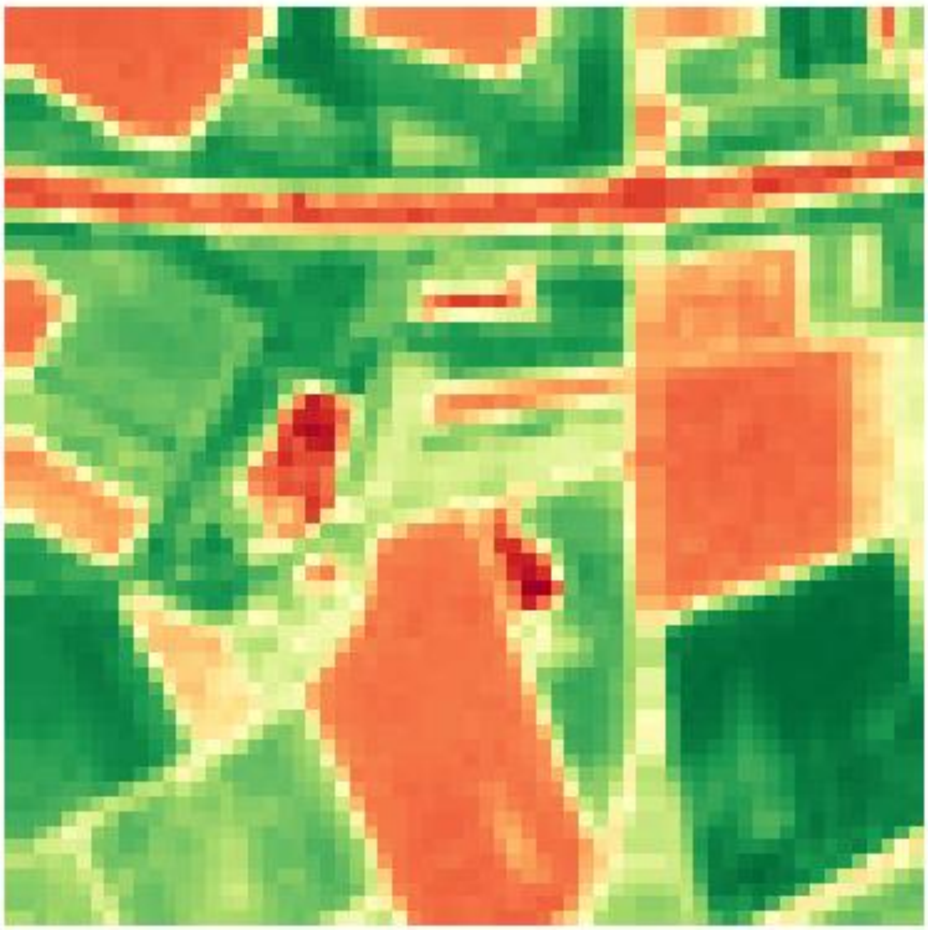

Figure 2: NDVI values

$NDWI = \frac{G-NIR}{G+NIR}$, where Green and Near Infrared (NIR) bands calculate in a value between [-1, 1]. Low NDWI values represent no water and high NDWI values represent water bodies. TorchGeo uses a diverging brown, white, blue-green colormap representing -1, 0, and 1, respectively.

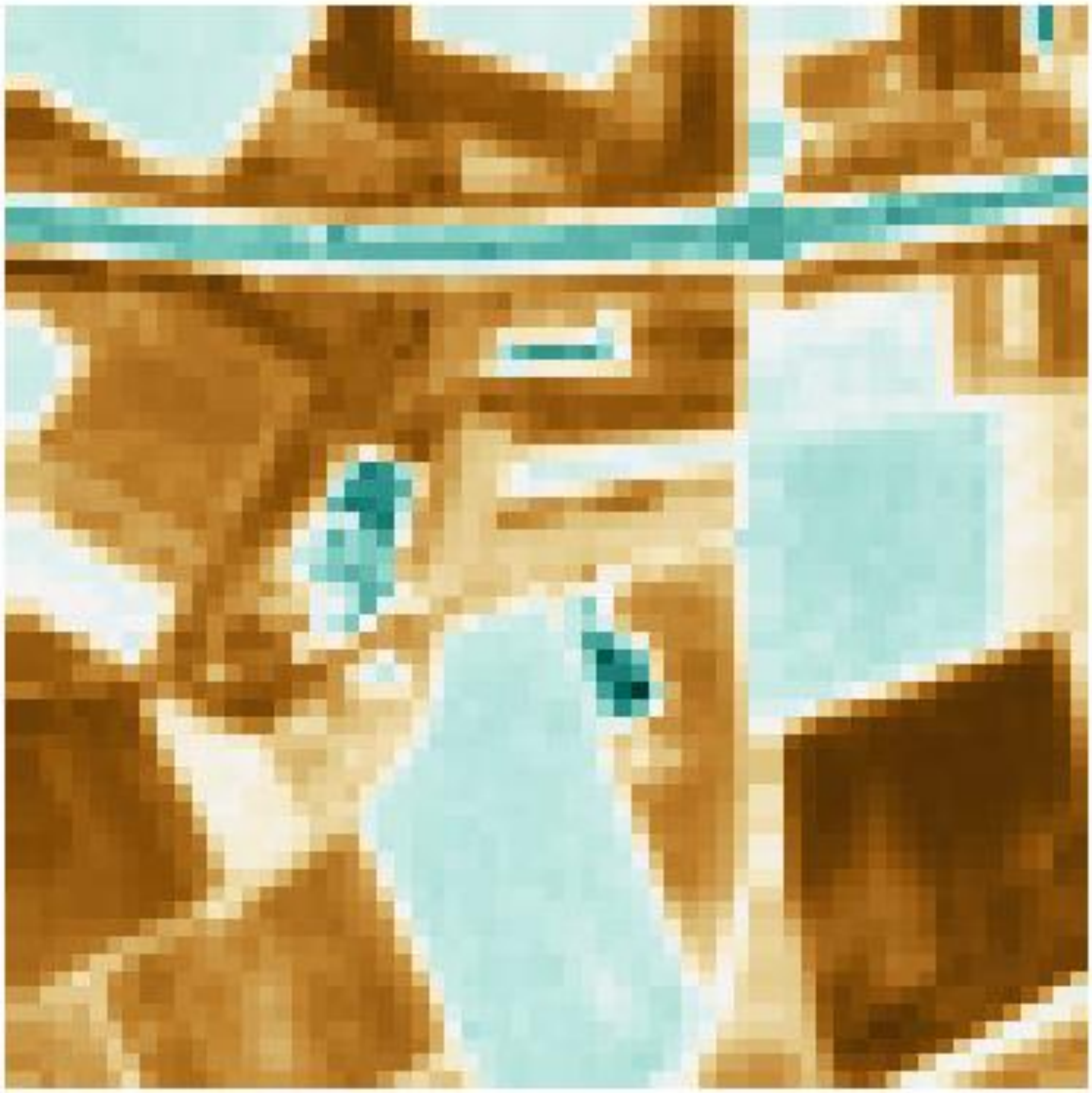

Figure 3: NDWI values

$NDBI = \frac{SWIR-NIR}{SWIR+NIR}$, where Short-wave Infrared (SWIR) and Near Infrared (NIR) bands calculate in a value between [-1, 1].

Low NDBI values represent no urban land and high NDBI values represent urban land. Here we use a terrain colormap with blue, green-yellow, and brown representing -1, 0, and 1, respectively.

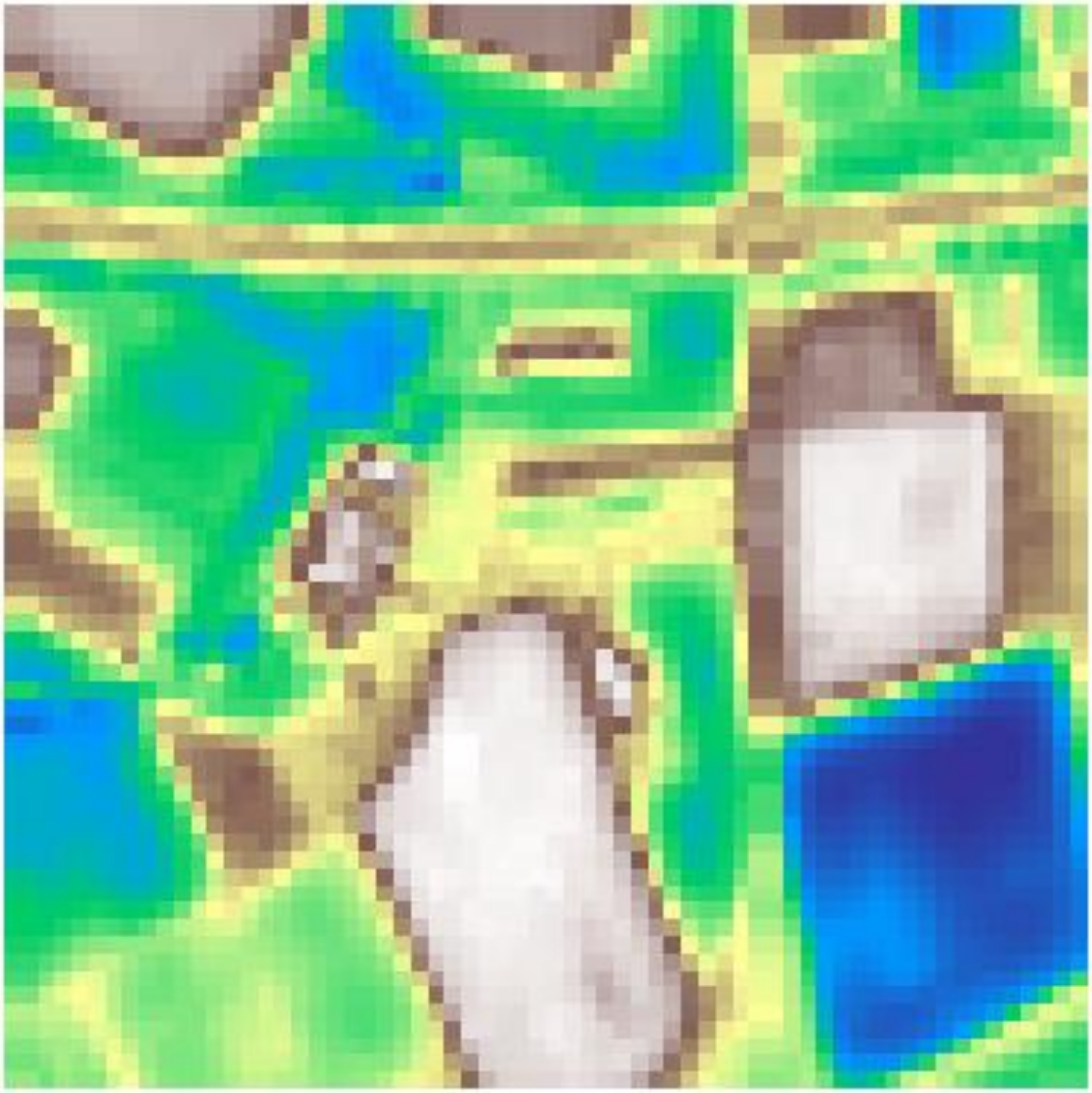

Figure 4: NDBI values

TorchGeo's data augmentation transforms include both the torch vision transforms and multispectral functions. The spectral indices could be chained as appended transformers along with augmentations from Kornia for a single callable during training, as below: MinMaxNormalize, AppendNDBI, AppendNDSI, AppendNDVI, AppendNDWI, RandomHorizontalFlip, RandomVerticalFlip and etc. In this project, for further training, only spectral indices are applied in the transforms.

Figure 5 and Figure 6 show the elements in each item of the dataset, after the transform pipeline.

Figure 5: RGB image after transform

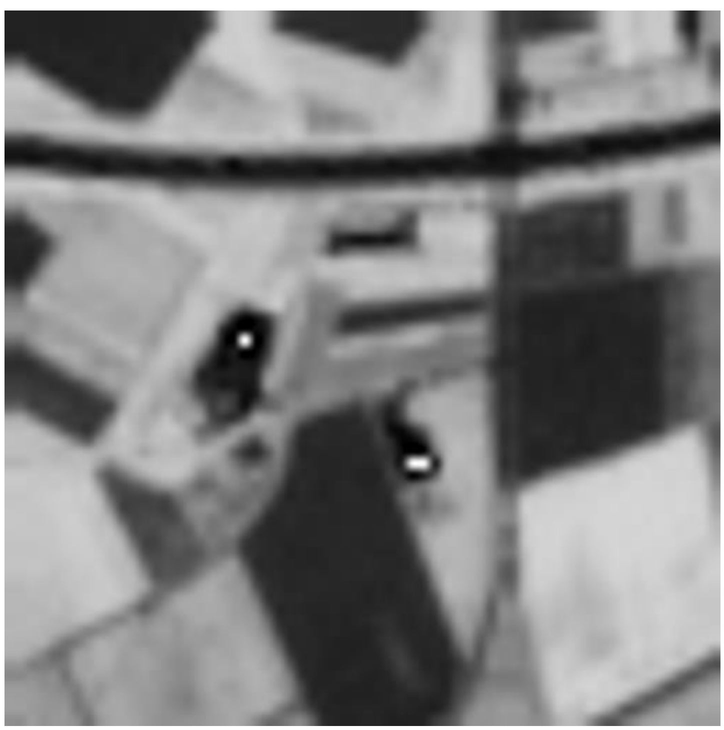

Figure 6: NDVI image after transform

4. Results

The training process is designed with the parameters below and results as Figure 7 & Figure 8 & Figure 9 & Figure 10.

```
batch_size = 10
num_workers = 2
max_epochs = 10
model="resnet50",
loss="ce",
in_channels=13,
num_classes=10,
lr=0.001,
patience=5
```

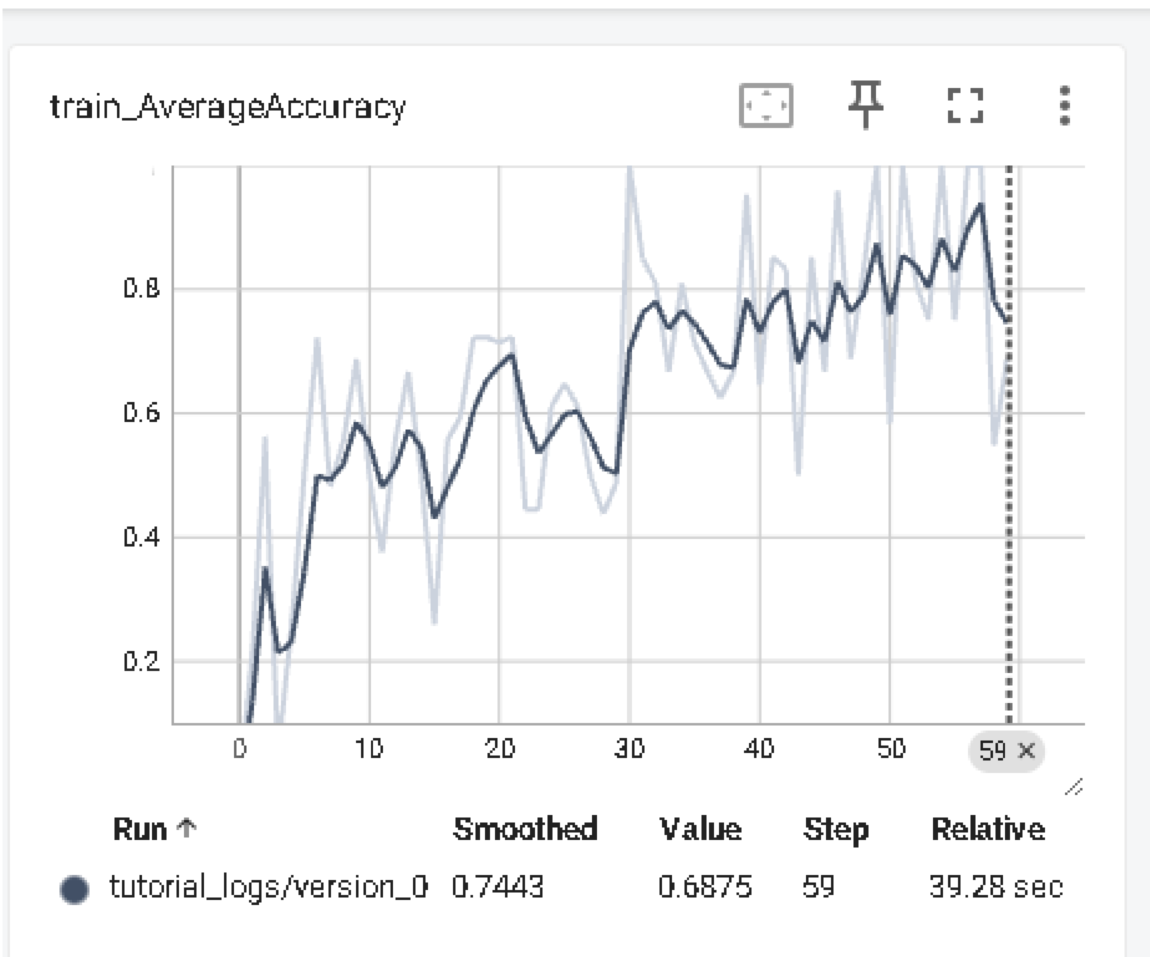

Figure 7: Training accuracy result with EuroSAT100 dataset

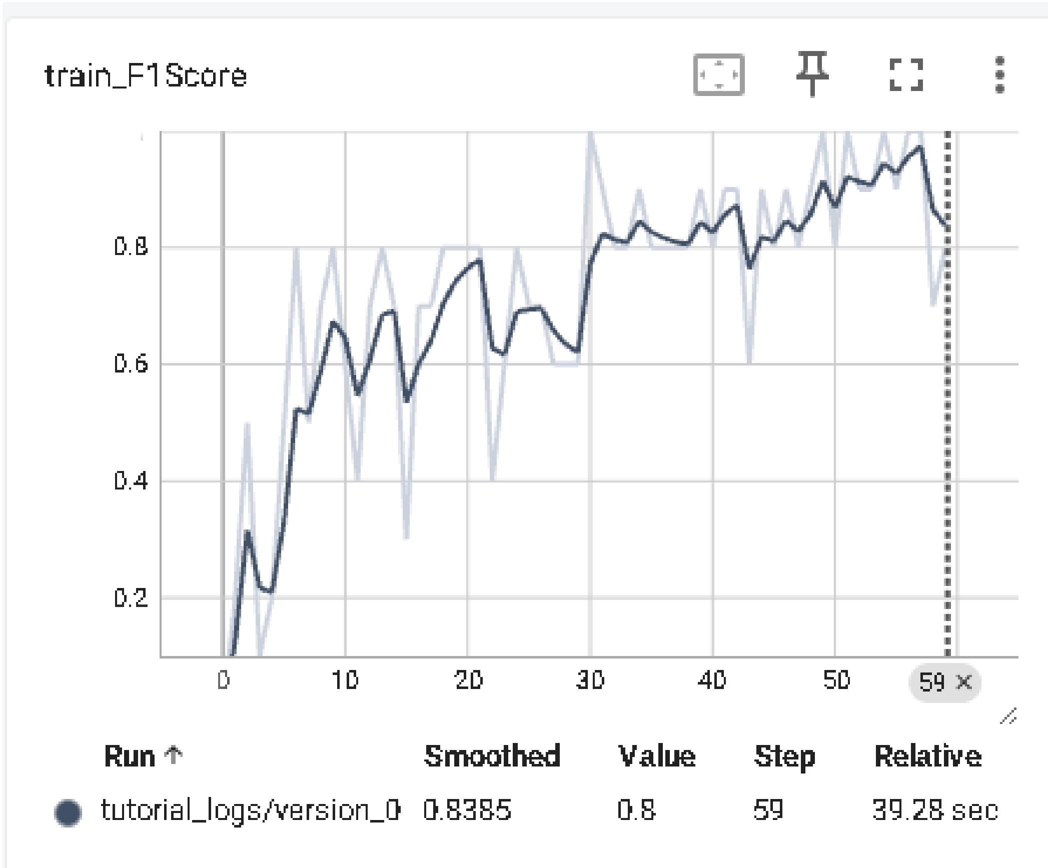

Figure 8: Training F1 result with EuroSAT100 dataset

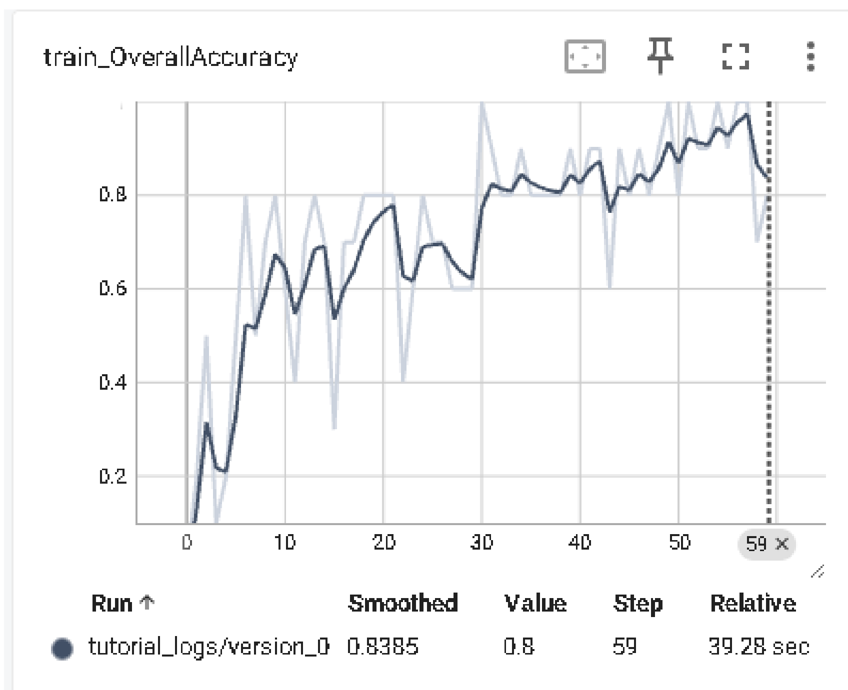

Figure 9: Training overall accuracy result with EuroSAT100 dataset

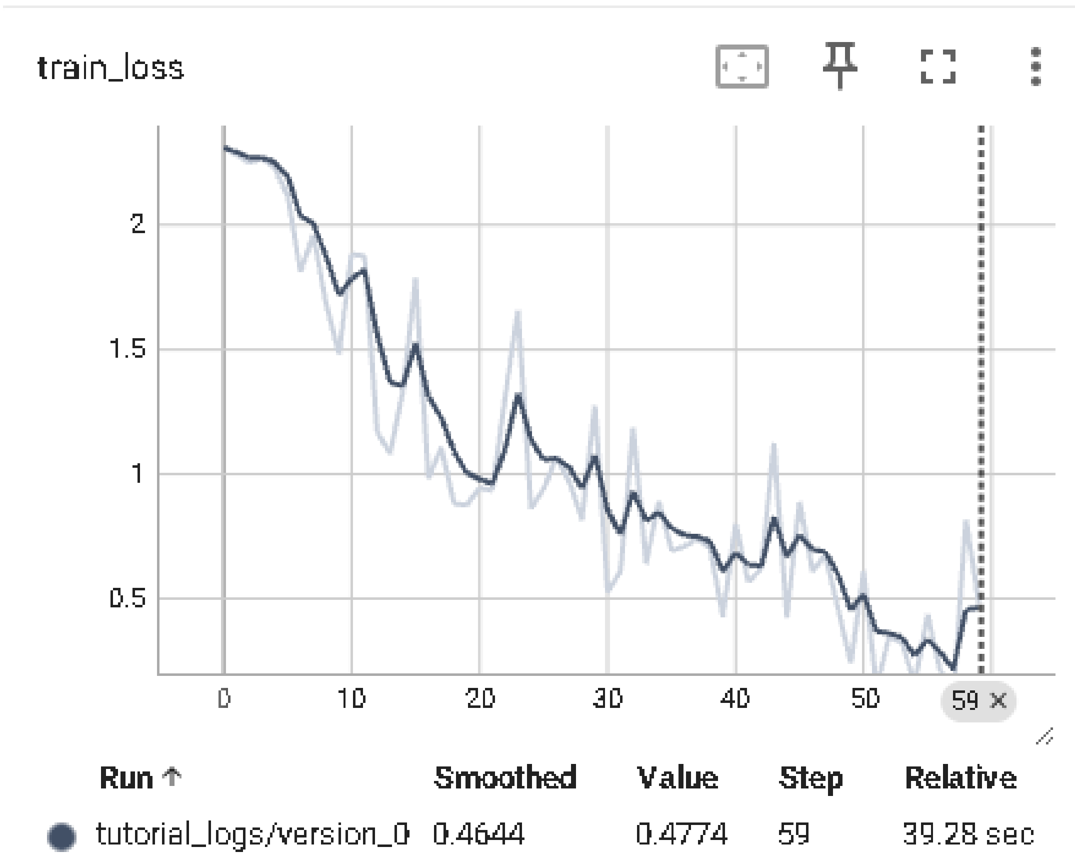

Figure 10: Training loss result with EuroSAT100 dataset

The validation results are Figure 11 & Figure 12 & Figure 13 & Figure 14 & Figure 15, while test results as Figure 16.

Validate metric	DataLoader 0
val_AverageAccuracy	0.5
val_F1Score	0.5
val_JaccardIndex	0.37000003457069397
val_OverallAccuracy	0.5
val_loss	1.5991766452789307

```
[{'val_loss': 1.5991766452789307,
  'val_AverageAccuracy': 0.5,
  'val_F1Score': 0.5,
  'val_JaccardIndex': 0.37000003457069397,
  'val_OverallAccuracy': 0.5}]
```

Figure 11: Validation results with EuroSAT100 dataset

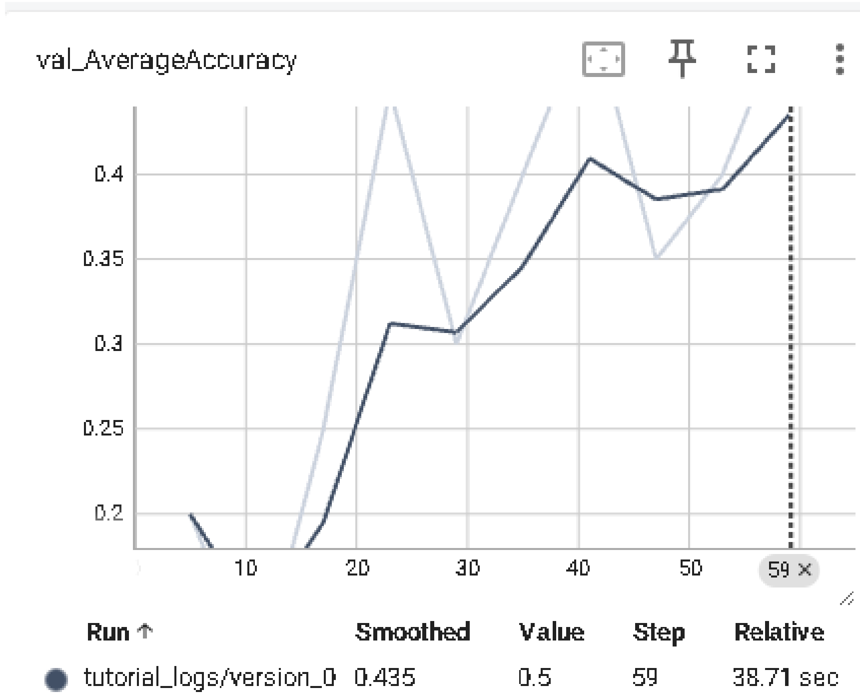

Figure 12: Validation average accuracy results with EuroSAT100 dataset

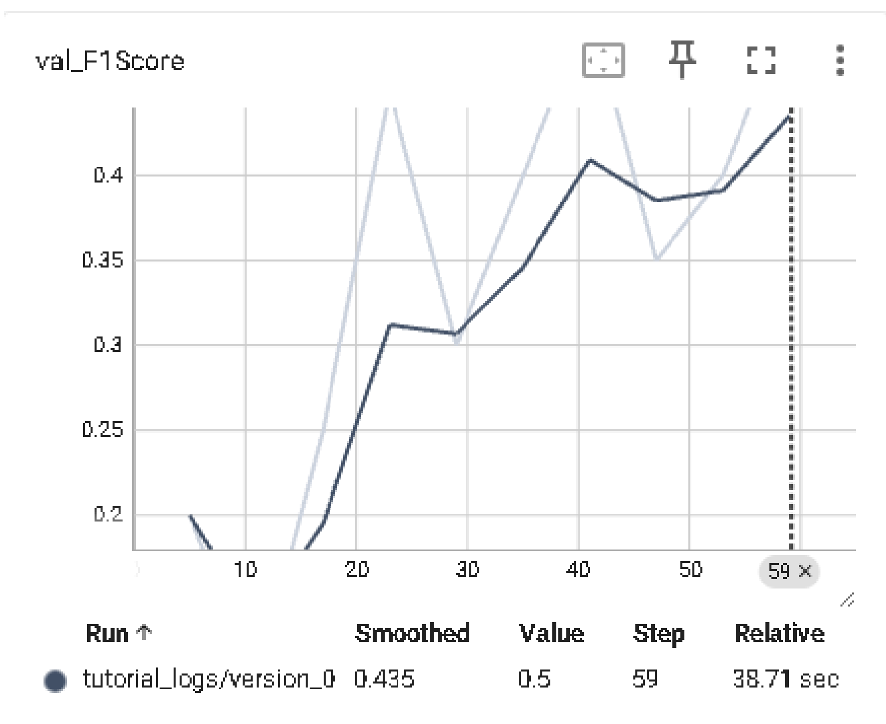

Figure 13: Validation F1 results with EuroSAT100 dataset

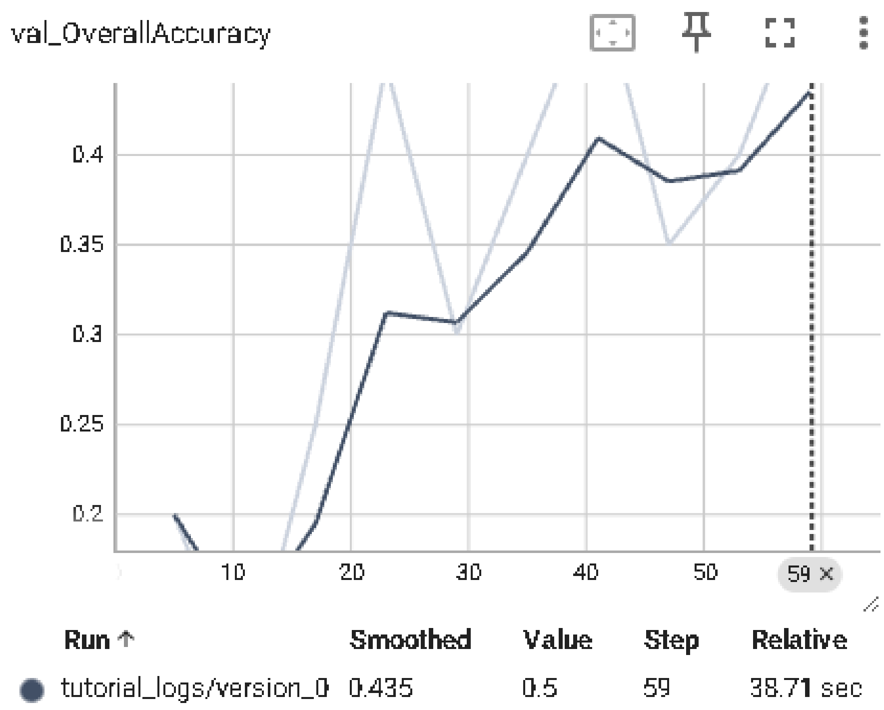

Figure 14: Validation overall accuracy results with EuroSAT100 dataset

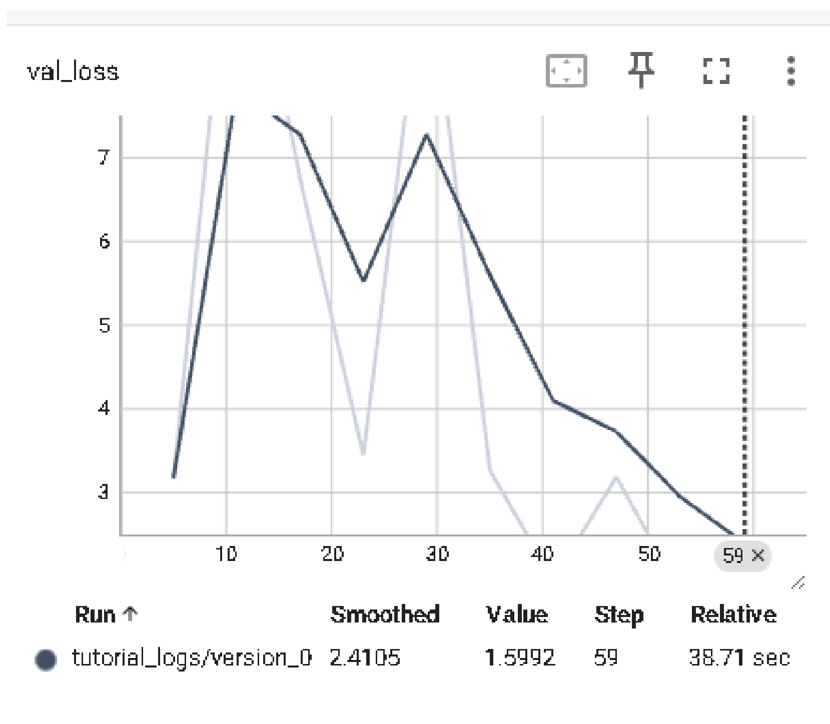

Figure 15: Validation loss results with EuroSAT100 dataset

Test metric	DataLoader 0
test_AverageAccuracy	0.550000011920929
test_F1Score	0.550000011920929
test_JaccardIndex	0.42000001668930054
test_OverallAccuracy	0.550000011920929
test_loss	1.1995702981948853

```
[{'test_loss': 1.1995702981948853,
  'test_AverageAccuracy': 0.550000011920929,
  'test_F1Score': 0.550000011920929,
  'test_JaccardIndex': 0.42000001668930054,
  'test_OverallAccuracy': 0.550000011920929}]
```

Figure 16: Test results with EuroSAT100 dataset

The prediction is from LandCover.AI dataset. Figure 10 shows one of the input items in the dataset, which includes both its image and its mask in this item. The prediction represents the features extracted from this input item, as in Figure 11, which are the positions of trees in this case calculated by the model trained with EuroSAT100 dataset.

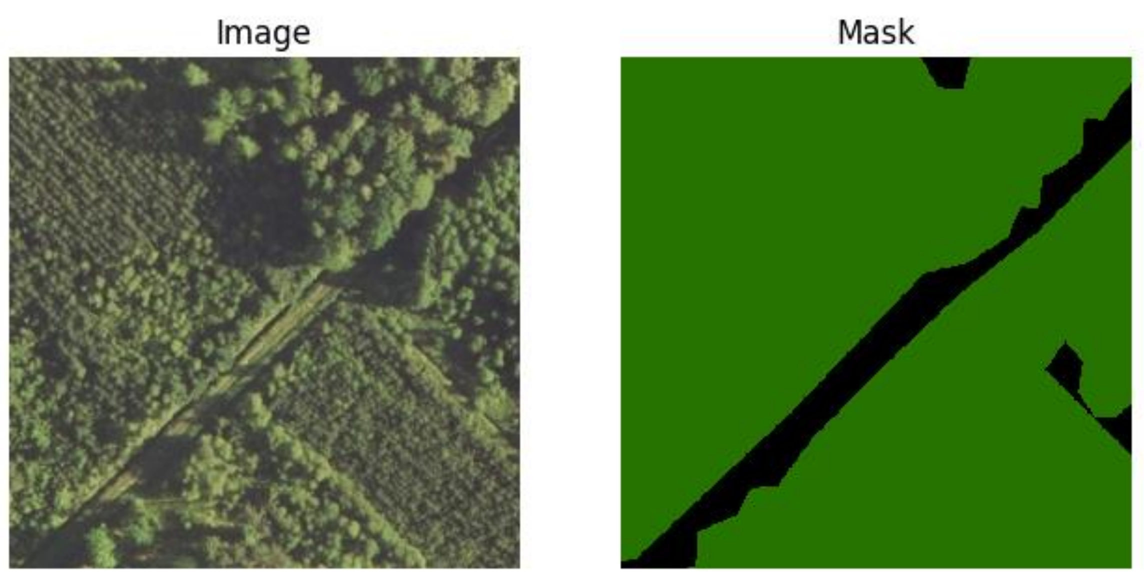

Figure 10: Original input with LandCover.AI dataset

Figure 11: Prediction result with input in Figure 10

5. Discussion

The difference between these approaches is likely entirely due to different learning rate selection in the hyperparameter search, and data augmentation in the in-domain pre-training setup.[2] Compared with the results in the TorchGeo paper [2], whose best result across EuroSAT datasets is trained with ImageNet pre-trained models and a low learning rate. Besides, transforms play a significant role in the feature extraction, especially in geospatial transforms multispectral indices emphasis the features and shrink the noise and extended spectral indices would improve the performance of the model.

6. Conclusion

With benchmark datasets and multispectral transforms, classification and segmentation can be achieved and improved through AI process with PyTorch Lightning.

Reference

[1] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with Momentum Contrastive learning. arXiv preprint arXiv:2003.04297, 2020.

[2] Adam J. Stewart, Caleb Robinson, and Isaac A. Corley. TorchGeo: Deep Learning With Geospatial Data. http://dl.acm.org/doi/pdf/10.1145/3557915.3560953

[3] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. 2019. EuroSAT: A novel dataset and deep learning benchmark for land use and land cover classification. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 12, 7 (2019), 2217–2226.

[4] Maxim Neumann, Andre Susano Pinto, Xiaohua Zhai, and Neil Houlsby. 2019. In-domain representation learning for remote sensing. arXiv preprint arXiv:1911.06721 (2019)

[5] Adrian Boguszewski, Dominik Batorski, Natalia Ziemba-Jankowska, Tomasz Dziedzic, and Anna Zambrzycka. 2021. LandCover.ai: Dataset for Automatic Mapping of Buildings, Woodlands, Water and Roads From Aerial Imagery. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 1102–1110.