Deep Learning and Computational Vision for Satellite Imagery Classification

PyTorch

NVIDIA
CUDA

Overview

Objective

Develop a model with an accuracy exceeding 80% for classifying satellite imagery into 10 distinct land cover categories.

Data

EUROSAT RGB is the RGB version of the EUROSAT dataset based on Sentinel-2 satellite images covering 13 spectral bands and consisting of 10 classes with 27000 labeled and geo-referenced samples. Classes: AnnualCrop, Forest, HerbaceousVegetation, Highway, Industrial, Pasture, PermanentCrop, Residential, River, SeaLake

huggingface.co/datasets/blanchon/EuroSAT_RGB

https://arxiv.org/abs/1709.00029

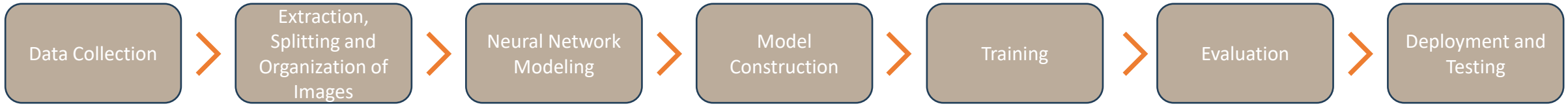
https://github.com/phelber/EuroSAT

Tools

Hugging Face (data collection), Git (version control), Python (programming), PyTorch (deep learning framework), NVIDIA CUDA (GPU processing), Jupyter Notebook, and VSCode (development environment).

Process flowchart

Model Development



Code Structure

01_DataManipulation.ipynb

- Extract and download dataset.
- Organize images for training/testing.
- Map land cover types to IDs.
- Dataset description and download link

02_DLModeling.ipynb

- Construct neural network with torch.nn.
- Define layers and activation functions.
- Modularize code in landneuralnetwork.py for easy reuse

03_TrainingModel.ipynb

- Preprocess data.
- Initialize model using defined structure.
- Implement Cross-Entropy Loss and Adam optimizer.
- training process and evaluate accuracy.
- Save model post-training.

04_DeployModel.ipynb

- Load trained model with preserved weights.
- Classify new images to assess applicability.
- Process test images for model compatibility.

Model and Training

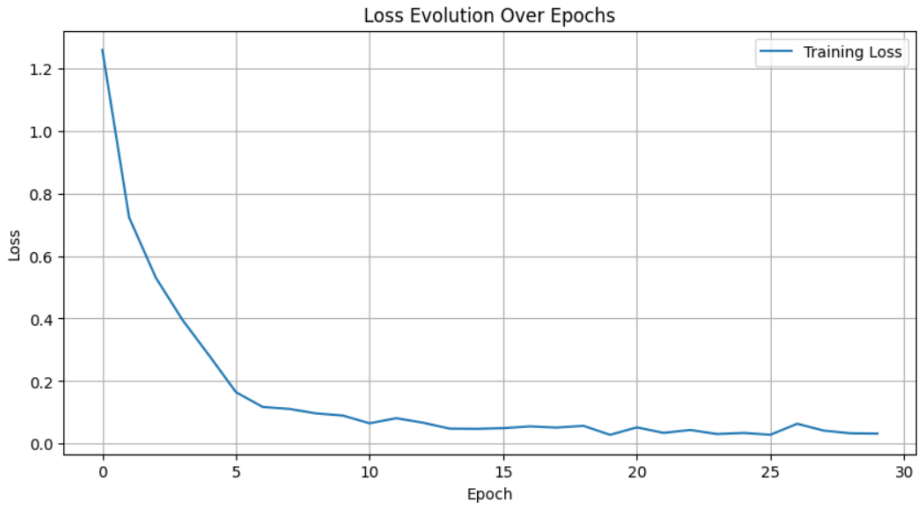
Model Structure

```
LandClassifierNet(
  (conv1): Conv2d(3, 64, kernel_size=(3, 3), stride=(1, 1))
  (conv2): Conv2d(64, 128, kernel_size=(3, 3), stride=(1, 1))
  (conv3): Conv2d(128, 256, kernel_size=(3, 3), stride=(1, 1))
  (dropout1): Dropout(p=0.25, inplace=False)
  (dropout2): Dropout(p=0.5, inplace=False)
  (fc1): Linear(in_features=215296, out_features=2048, bias=True)
  (fc2): Linear(in_features=2048, out_features=512, bias=True)
  (fc3): Linear(in_features=512, out_features=128, bias=True)
  (fc4): Linear(in_features=128, out_features=10, bias=True)
)
```

This neural network, constructed with PyTorch for land cover classification, is composed of the following layers:

- Three convolutional layers (conv1, conv2, conv3) with increasing filter sizes (64, 128, 256) and kernel size 3x3 for feature extraction. Each convolutional layer is followed by a ReLU activation function and uses a stride of (1,1).
- Two dropout layers (dropout1, dropout2) with probabilities of 0.25 and 0.5, respectively, to prevent overfitting by randomly zeroing some of the elements of the input tensor.
- Four fully connected layers (fc1, fc2, fc3, fc4) that decrease in neuron count from 215296 to 2048, then to 512, and to 128, with the final layer reducing to 10 output features, corresponding to the 10 classes of land cover. The presence of bias in all fully connected layers enables the model to learn appropriate shifts in the decision boundary.

Training



Accuracy

82.21%

Deploy

The model has been saved to 'nn_model.pth' and is now being imported into the development environment to simulate a deployment. Below are images from the test database, along with their respective classifications as predicted by the model.

Prediction: Pasture

Prediction: Industrial

Prediction: Residential

Prediction: Pasture

Prediction: Pasture

Prediction: PermanentCrop

Prediction: SeaLake

Prediction: Industrial

Links

Project

github.com/muriloms/land-use-cover-classification

Dataset

huggingface.co/datasets/blanchon/EuroSAT_RGB
github.com/phelber/EuroSAT

Course

datascienceacademy.com.br/bundle/formacao-engenheiro-de-inteligencia-artificial-4

Tools

pytorch.org
developer.nvidia.com/cuda-toolkit
anaconda.com