Final Exam: Image Processing
Tree Species Classification

Academic Year 2024-2025

1 Project Overview

This project focuses on implementing and comparing methods for tree species classification using point cloud data and multi-view approaches. Students must evaluate and compare the performance of various classification techniques to determine the most effective approach for this task.

2 Data Acquisition and Preparation

2.1 Dataset Download

Download the dataset from:
`https://data.goettingen-research-online.de/dataset.xhtml?persistentId=doi:10.25625/FOHUJM`

Important Note: The download cannot be completed in a single operation. Divide the process into steps and ensure accuracy in the final count for each class. Pay careful attention to avoid errors during this process!

At the end of this operation, you should have **7 directories**, with each directory named after a tree species class.

Species	**Number of Trees**
Beech	164
Red Oak	100
Ash	39
Oak	22
Douglas Fir	183
Spruce	158
Pine	25
Total	**691**

Table 1: Number of trees per species in the dataset.

2.2 Data Visualization

To explore the dataset effectively:

- **2D Visualization:** Use `matplotlib` for 2D data exploration and analysis
- **3D Visualization:** Use `Open3D` for comprehensive 3D point cloud visualization

2.3 Data Splitting

- **Test Set:** Use the samples described in `test.csv`
- **Training Set:** Use the remaining data (80% of total) for training purposes

3 Classification Methods

3.1 Indirect Methods: Via 2D (Multi-view Approaches)

Use `PyTorch` and `torchvision` for pretrained models:

1. **Multi-view** + {Classical Descriptors} + {Classical Machine Learning Model}: Combine multi-view images with traditional feature descriptors and classical ML models
2. **Multi-view + CNN from Scratch:** Train a convolutional neural network from scratch using multi-view data
3. **Multi-view** + {CNN-pretrained} **for Fine-tuning:** Use pretrained CNN models and fine-tune them for tree species classification
4. **Multi-view** + {CNN-pretrained} + {Classical Machine Learning Model}: Extract features using pretrained CNNs and apply classical ML models for classification
5. **Multi-view** + {CNN-pretrained} **for Transfer Learning:** Apply transfer learning techniques with pretrained CNN models
6. **Multi-view** + {Classical Descriptors} + **CNN (Data Fusion):** Combine classical descriptors with CNN features (CNN can be {pretrained or trained from scratch})

3.2 Quasi-Direct Methods

Use descriptors directly on 3D point clouds:

- {3D Descriptors} + {Classical Machine Learning}: Extract 3D features from point clouds and apply classical ML algorithms

3.3 Direct Methods

Use deep learning methods that learn directly from 3D point clouds:

Point Cloud Sampling: For PointNet, sample **1024 points** using the **Farthest Point Sampling (FPS) technique**.

1. **PointNet:** Implementation available at: `https://github.com/charlesq34/pointnet`
2. **Dynamic Graph Convolution Neural Network (DGCNN):** Implementation available at: `https://github.com/WangYueFt/dgcnn`

3.4 Bonus Methods (Optional)

1. **Multi-view + Dynamic Graph Convolution Neural Network (Multi-modal Learning):**
   Combine multi-view approaches with DGCNN for enhanced performance
2. **DGCNN for Feature Extraction + `{Classical Machine Learning Model}` for Classification:**
   Use DGCNN to extract features and classical ML models for final classification

4 Evaluation Metrics

Your models will be evaluated using the following comprehensive metrics:

- **Overall Accuracy**
- **Balanced Accuracy**
- **F1 Score**
- **Precision for Each Class**
- **Execution Time**
- **Number of Parameters (Model Complexity)**

5 Optimization and Evaluation

Model Optimization: Optimize your models to achieve the best possible accuracy and results. Consider hyperparameter tuning, data augmentation, and ensemble methods where appropriate.

Evaluation Criteria: The evaluation will be based on the quality of your results. Focus on achieving high performance across all metrics while considering computational efficiency and model interpretability.

6 Implementation Guidelines

- Items between {} in the original specification will vary from group to group
- Ensure proper documentation of your implementation with clear code comments
- Provide comprehensive comparative analysis between different methods
- Include visualization of results and confusion matrices for all methods
- Discuss the advantages and limitations of each approach in detail
- Consider computational requirements and scalability of each method

7 Deliverables

Submit the following components:

1. Complete implementation of all selected classification methods.
2. Comprehensive evaluation report including all specified metrics (e.g., accuracy, precision, recall, F1-score).
3. Data and results visualization (including confusion matrices and relevant performance plots).
4. Comparative analysis and detailed discussion of findings.
5. Well-documented source code with a clear structure and sufficient comments.
6. A short written report structured as follows:
   - **Introduction**: Present the problem being addressed, the motivation, objectives, and a brief overview of the proposed methods. Provide background and cite related work using BibTeX.
   - **Methods**: Describe the datasets, preprocessing steps, and the classification algorithms used. Include hyperparameters, evaluation strategies, and implementation details (e.g., frameworks like PyTorch, Scikit-learn, etc.).
   - **Results and Discussion**: Present and compare the performance of each method using quantitative metrics and visualizations. Discuss insights, anomalies, and potential reasons for observed results.
   - **Conclusion**: Summarize the findings, state the best-performing methods, and propose future improvements or work.
   - **References**: Use BibTeX to format all citations properly.
7. Use the following official arXiv/BioRxiv LaTeX template for your report: **Style and Template for Preprints – arXiv / bioRxiv on Overleaf**.
8. A presentation of a maximum of **20 slides** summarizing the main objectives, methodology, key results, and conclusions.