Hyperspectral Image Analysis: Spectral Unmixing and Classification

This project focuses on analyzing hyperspectral image data from the **Pavia University dataset**. It is divided into two main parts:

1. **Spectral Unmixing**
2. **Classification**

Project Structure

Files

- `Part 1 Spectral Unmixing.ipynb`: Notebook for performing spectral unmixing, including Least Squares (LS), constrained LS, and LASSO methods.
- `Part 2 Classification.ipynb`: Notebook for classifying hyperspectral image data using machine learning models.

Dataset

- **Pavia University Dataset**
  Files used:
  - `PaviaU_cube.mat` (HSI data)
  - `PaviaU_endmembers.mat` (Endmember matrix)
  - `PaviaU_ground_truth.mat` (Ground truth labels)

Goals

Part 1: Spectral Unmixing

- Implement various unmixing techniques:
  1. **Least Squares (LS)**
  2. **Least Squares with sum-to-one constraint**
  3. **Least Squares with non-negativity constraint**
  4. **Least Squares with both constraints**
  5. **LASSO with sparsity**
- Evaluate results based on:
  - Abundance maps
  - Reconstruction error

Part 2: Classification

Tasks:

(A) Classifier Training and Evaluation

1. **Train each classifier** using 10-fold cross-validation:
   - Compute the **validation error** as the mean of the 10 resulting error values.
   - Report the **standard deviation** of the validation error.
2. **Train classifiers on the entire training set** and evaluate on the test set:
   - Compute the **confusion matrix** (7x7), where element (i,j) represents the number of pixels from class `i` classified as class `j`.
   - Analyze the confusion matrix:
     - **Diagonal dominance** indicates better classification.
     - Identify **poorly separated classes**.
   - Calculate the **success rate**:
     - Success Rate = Sum of diagonal elements/ Sum of all elements in the confusion matrix}}]

(B) Comparative Analysis

- Compare results across classifiers:
  - Relate the confusion matrices.
  - Pay attention to **non-diagonal entries** that are significantly different from zero.
  - Discuss which classifiers struggle with specific class separations.

Classifiers Used: 1. **Naïve Bayes Classifier** 2. **Minimum Euclidean Distance Classifier** 3. **K-Nearest Neighbor (KNN) Classifier** 4. **Bayesian Classifier**

Tools and Technologies

- **Python**
  - NumPy, SciPy, scikit-learn, and Matplotlib
- **MATLAB Files**: For handling `.mat` data

How to Run

1. Upload the dataset files to the notebook environment.
2. Open and execute the respective notebooks (`Part 1` and `Part 2`).
3. Follow the outputs and visualizations for analysis.

Results and Analysis

Detailed results include: - Visualization of abundance maps for each method in Part 1. - Classification accuracy, confusion matrices, and success rates for each model in Part 2. - Comparative analysis of classifier performance.