Week 1: Data acquisition and development setup

Tasks:

- Download Mars hyperspectral Data (CRISM, M3) and thermal infrared data from NASA Planetary Data System;
- Install Python development environment and necessary libraries (GDAL/rasterio, scikit-learn, TensorFlow/Keras or PyTorch, scikit-fuzzy, etc.);
- Read and run the opengeotutorial example on GitHub to become familiar with data reading and preliminary visualization.--

Refer to the papers:

- A_review_of_machine_learning.pdf (To understand the general framework of machine learning in remote sensing data processing);
- Farahbakhsh-Chandra_RemoteSensing2019.pdf (for ideas related to remote sensing data preprocessing and correction).

Week 2: Data preprocessing and fusion

Tasks:

- Radiometric, atmospheric and geometric correction of the downloaded data (if necessary, ENVI or Python can be called to implement the corresponding algorithm);
- Spatial registration of different data sources (hyperspectral and thermal infrared) was performed to ensure the alignment of image data.
- The data were cropped to the study area, and the bands were resampled (e.g., the SWIR bands of ASTER were resampled to VNIR resolution).

Refer to papers:

- remotesensing-12-01261-v2.pdf (details on ASTER data preprocessing and resampling methods);
- Farahbakhsh-Chandra_RemoteSensing2019.pdf .

Week 3: Feature Extraction and dimensionality reduction

Tasks:

- According to the spectral characteristics of the target minerals (key mineral absorption/reflection features), select the subset of bands of interest;
- PCA, ICA and MNF were used to reduce the dimension of the spectral subset and extract the main features.
- The appropriate threshold was determined by drawing the concentration area (C-A) fractal image, and the target region was binarized.

Refer to the paper:

- remotesensing-12-01261-v2.pdf (Application of comparative and integrated dimensionality reduction methods in Mineral remotesensing);
- A_review_of_machine_learning.pdf .

Week 4: Traditional machine Learning model building

Tasks:

- scikit-learn was used to implement SVM (using RBF kernel) and MLP models to classify the mineral categories of the feature data after dimensionality reduction.
- Parameter tuning (such as cross validation, grid search) was performed, ROC curve was drawn, AUC and other indicators were calculated.

Refer to papers:

- Farahbakhsh-Chandra_RemoteSensing2019.pdf (provides an example of the application of traditional ML methods in remote sensing mineral mapping);
- remotesensing-14-00819-v4.pdf (comparing the performance of CNN with traditional methods).

Week 5: Deep Learning Model Building (CNN)

Tasks:

- TensorFlow/Keras (or PyTorch) was used to build a CNN model, and the network structure was designed (e.g., two convolutional layers + fully connected layers, ReLU activation function and Softmax output layer).
- Data augmentation (rotation, translation, scaling, etc.) was performed on the training data to improve the generalization ability of the model.
- The training, validation, and test sets were divided to initiate preliminary training.

Refer to papers:

- remotesensing-14-00819-v4.pdf (details of the advantages and architecture design of CNN in remotesensing image classification);
- A_review_of_machine_learning.pdf .

Week 6: Model training and optimization

Tasks:

- The hyperparameters of the CNN model (such as convolution kernel size, step size, learning rate, batch size, etc.) were tuned, and the training curve was recorded.
- At the same time, SVM and MLP models were further tuned and compared to optimize model performance.
- ROC, AUC, confusion matrix and other indicators were used to evaluate each model to determine the best model.

Reference papers:

- remotesensing-14-00819-v4.pdf (refer to its training strategies and performance evaluation methods);
- Farahbakhsh-Chandra_RemoteSensing2019.pdf .

Week 7: Spatial distribution mapping and post-processing

Tasks:

- The best classification model was used to generate the pixel-level classification results of Mars minerals, and the mineral spatial distribution map was made based on the classification results.
- Post-processing methods (such as majority rule, fuzzy logic weighted fusion) were used to smooth and correct the classification results, and the final remote sensing evidence layer was generated.
- Libraries such as geopandas, matplotlib or folium were used to visualize the map.

Refer to the papers:

- remotesensing-12-01261-v2.pdf (provides methods for generating evidence layers using dimensionality reduction and fuzzy logic integration);
- Farahbakhsh-Chandra_RemoteSensing2019.pdf .

Week 8: Model validation and results analysis

Task:

- The existing ground truth or field data were used to verify the accuracy of mineral classification and spatial distribution map, and the confusion matrix, ROC curve, predicted area (P-A) map and other evaluation reports were generated.
- The uncertainties, mixed pixel problems and limitations in the model results were analyzed, and the improvement direction was discussed.
- Write the preliminary results analysis report.

Refer to papers:

- remotesensing-14-00819-v4.pdf (section on model evaluation and uncertainty analysis);
- A_review_of_machine_learning.pdf .

Week 9: Code consolidation, documentation, and integration

Tasks:

- Integrate, clean and optimize the whole code base to ensure the seamless docking of each module;
- Write detailed code instructions and manuals, and organize them into Jupyter notebooks (refer to open source tutorials);
- Collate the experimental results, generate the final charts, and write the methods section and appendix to prepare the results for submission.
- Publish the final code and documentation on GitHub.

Refer to the papers:

- remotesensing-14-00819-v4.pdf and remotesensing-12-01261-v2.pdf (provide complete framework and code reproduction ideas);
- Farahbakhsh-Chandra_RemoteSensing2019.pdf (as a reference for the overall technical flow and documentation).