Project Report -

Rock Classification with a RAMAN Spectrometer

Name: Tanmay Talreja (m12519565)

Abstract

This project investigates the automated classification of seven mineral species: Calcite, Dolomit, Feldspat, Quarz, Rhodocrosite, Tile, and Albit. The study utilizes **four distinct data modalities based on frame rate: Spectral Images (30 fps), Spectral Images (1 fps), Normalized Spectral Profiles (30 fps), and Normalized Spectral Profiles (1 fps)**. We evaluate several architectures, specifically ResNet50 (a traditional CNN), MobileViT-XS (a Transformer-based hybrid) and Random Forest on flattened images for image data, and 1D-CNN, MLP, and Random Forest for profile data. Our findings demonstrate that image-based models achieve a perfect 100% accuracy, with MobileViT-XS identified as the superior model due to its balance of speed and memory efficiency. Profile-based models achieve near-perfect accuracy (up to ~99.9%) on the 1 fps data, with MLP and Random Forest models emerging as winners for resource-constrained environments.

I. INTRODUCTION

Mineral identification is a critical task in geology and mining engineering, traditionally performed through manual inspection or classical Raman spectroscopy techniques. However, in modern industrial setups, the environment is rarely static; it is often dynamic and fast-paced, making traditional methods slow and prone to error.

This project aims to automate this process by integrating Artificial Intelligence (AI) with Raman spectroscopy. Raman sensors are uniquely suited for this task as they provide detailed molecular "fingerprints" of minerals, which are more informative than standard RGB camera data in identifying chemical structures. By processing **four distinct data modalities: 2D spectral images and 1D spectral profiles, each sampled at 30 fps and 1 fps** through various machine learning pipelines, we identify the most robust architectures for real-time mineral classification.

The objectives of this project are:

1. To understand why Raman spectroscopy is preferred over camera sensors for mineral classification.
2. To identify the limitations of classical Raman classification methods.
3. To apply machine learning models to Raman data.
4. To compare spectral image–based and spectral profile–based approaches.
5. To evaluate different model architectures in terms of accuracy and efficiency.
6. To select a model suitable for scalable and real-world deployment.

II. QUESTION–ANSWER ANALYSIS

A. Why Raman spectroscopy instead of camera sensors?

Raman spectroscopy captures chemical composition rather than surface appearance. Camera sensors only record visual features such as color and texture, which are unreliable for minerals that look similar but differ chemically. Raman sensors are also robust to lighting variations and surface contamination.

B. Limitations of classical Raman classification

Classical Raman classification relies on manual peak detection and rule-based matching. These methods are sensitive to noise, FPS variations, and overlapping peaks, leading to poor generalization in dynamic environments.

C. Role of machine learning in mineral classification

Machine learning enables automatic feature extraction from Raman data, reducing reliance on handcrafted features. ML-based methods are more robust to noise and scale better with increasing data size.

D. Models used for classification

For Spectral Images, we focused on deep learning. ResNet50 provided a baseline for CNN performance, while MobileViT-XS combined the efficiency of MobileNet with the global context of Transformers and Random Forest on flattened images was used for a different approach in image classification. For Spectral Profiles, we implemented a 1D-CNN (utilizing Conv1d and AdaptiveMaxPool1d layers) and a 3-layer MLP with BatchNorm and Dropout to prevent overfitting. Along with Random Forest for a classical approach.

E. Use of spectral profiles instead of images

Experiments show that normalized Raman spectral profiles, are highly effective for classification, achieving near-perfect accuracy (~99.9%) for 1 fps and (~96%) for 30 fps. Profile-based learning significantly reduces memory usage and training time compared to image-based methods.

F. Training strategy

Supervised learning is used due to the availability of labeled mineral classes and clear class boundaries.

G. Loss functions

All Deep Learning models (ResNet50, MobileViT-XS, 1D-CNN, MLP) use Cross-Entropy Loss. This is the standard loss function for multi-class classification tasks, as it internally applies Softmax to the model's logits (raw scores) in frameworks like PyTorch. All Random Forest classifiers use Gini Impurity as the splitting criterion.

General Mathematical Formula

For a single sample, the standard **Cross-Entropy Loss** is defined as:

$$L_{CE} = -\sum_{c=1}^{C} y_c \log(p_c)$$

Where:

- C is the total number of classes.
- y_c is the binary indicator (0 or 1) if class c is the correct classification for the observation (one-hot encoded).
- p_c is the predicted probability that the observation is of class c.

PyTorch `nn.CrossEntropyLoss` Formula

In PyTorch, `nn.CrossEntropyLoss` combines **LogSoftmax** and **NLLLoss** (Negative Log Likelihood Loss) into a single, numerically stable class. It expects raw **logits** (unnormalized scores) as input, not probabilities.

For a single sample x (logits) and class label y (class index):

$$\text{loss}(x, y) = -\log\left(\frac{\exp(x_y)}{\sum_{j=1}^{C}\exp(x_j)}\right) = -x_y + \log\left(\sum_{j=1}^{C}\exp(x_j)\right)$$

Where:

- x is the input vector of logits (raw scores) for the sample.
- xy is the logit corresponding to the **true** class y.
- xj is the logit for class j.
- C is the number of classes.
- The term $\frac{\exp(x_y)}{\sum \exp(x_j)}$ represents the **Softmax** probability of the true class.

If calculating the mean over a batch of size N:

$$\ell(x, y) = \frac{1}{N} \sum_{n=1}^{N} l_n$$

Gini Impurity (for Random Forest Models)

Random Forest (and Decision Trees) use Gini Impurity to determine the quality of a split. It measures the likelihood that a randomly selected element from the set would be incorrectly labeled if it were randomly labeled according to the distribution of labels in the set.

Mathematical Formula

The Gini Impurity G for a node containing samples from C classes is:

$$G = 1 - \sum_{i=1}^{C} (p_i)^2$$

Where:

- C is the number of classes.
- p_i is the probability (or fraction) of samples belonging to class i at that specific node.

Splitting Criterion

When a Random Forest splits a node, it seeks to minimize the **weighted Gini Impurity** of the child nodes. The "Gini Gain" or improvement is calculated as:

$$\text{Gini Gain} = G_{parent} - \left(\frac{N_{left}}{N_{total}} G_{left} + \frac{N_{right}}{N_{total}} G_{right} \right)$$

Where:

- N_{total} is the number of samples in the parent node.
- N_{left},N_{right} are the number of samples in the left and right child nodes, respectively.
- G_{left},G_{right} are the Gini impurities of the child nodes.

H. Data preprocessing

Preprocessing includes normalization of Raman intensities, resizing of spectral images, label encoding, and train-test splitting.

I. Evaluation metrics

Accuracy is used as the primary metric. Since all models achieved perfect accuracy (for image data), training time, memory usage, and model complexity were used for comparative evaluation.

III. METHODOLOGY

A. Data Modalities

The project utilized a dataset consisting of 7 mineral classes, including an "Albite" class sourced from a separate directory. Four distinct data modalities were processed, varying by data type and frame rate (FPS):

1. **Spectral Images (30 fps & 1 fps):** .bmp files resized to 224x224 and converted to RGB tensors.
2. **Normalized Spectral Profiles (30 fps & 1 fps):** .csv files representing 1D spectral measurements.

B. Model Architectures

For Spectral Images, we focused on deep learning. ResNet50 provided a baseline for CNN performance, while MobileViT-XS combined the efficiency of MobileNet with the global context of Transformers and Random Forest on flattened images was used for a different approach in image classification. For Spectral Profiles, we implemented a 1D-CNN (utilizing Conv1d and AdaptiveMaxPool1d layers) and a 3-layer MLP with BatchNorm and Dropout to prevent overfitting. Along with Random Forest for a classical approach.

C. Training Configuration

Models were trained using the Adam optimizer with a learning rate of 0.0001 and a batch size of 8 (images) or 16 (profiles). To ensure reproducibility, a global seed was set. Training was conducted for up to 10 epochs for profiles and 5 for images, with early stopping implemented if 100% accuracy was achieved.

Data Type	FPS	Model	Accuracy	Training Time	Memory Usage
Image	**30**	**MobileViT-XS**	**100%**	**Fast**	**Low**
Image	30	ResNet50	100%	Moderate	High
Image	30	Random Forest	100%	Moderate	High
Image	**1**	**MobileViT-XS**	**100%**	**Fast**	**Low**
Image	1	ResNet50	100%	Moderate	High
Image	1	Random Forest	100%	Moderate	High
Profile	**30**	**Random Forest**	**~96.3%**	**Instant**	**Very Low**
Profile	**30**	**MLP**	**~95.7%**	**Very Fast**	**Very Low**
Profile	30	1D-CNN	~85.4%	Very Fast	Very Low
Profile	**1**	**Random Forest**	**~99.9%**	**Instant**	**Very Low**
Profile	1	**MLP**	**~99.7%**	**Very Fast**	**Very Low**
Profile	1	1D-CNN	~99.5%	Very Fast	Very Low

IV. RESULTS AND DISCUSSION**Analysis: The experimental results demonstrate a clear picture of model performance across data modalities and frame rates (FPS).**

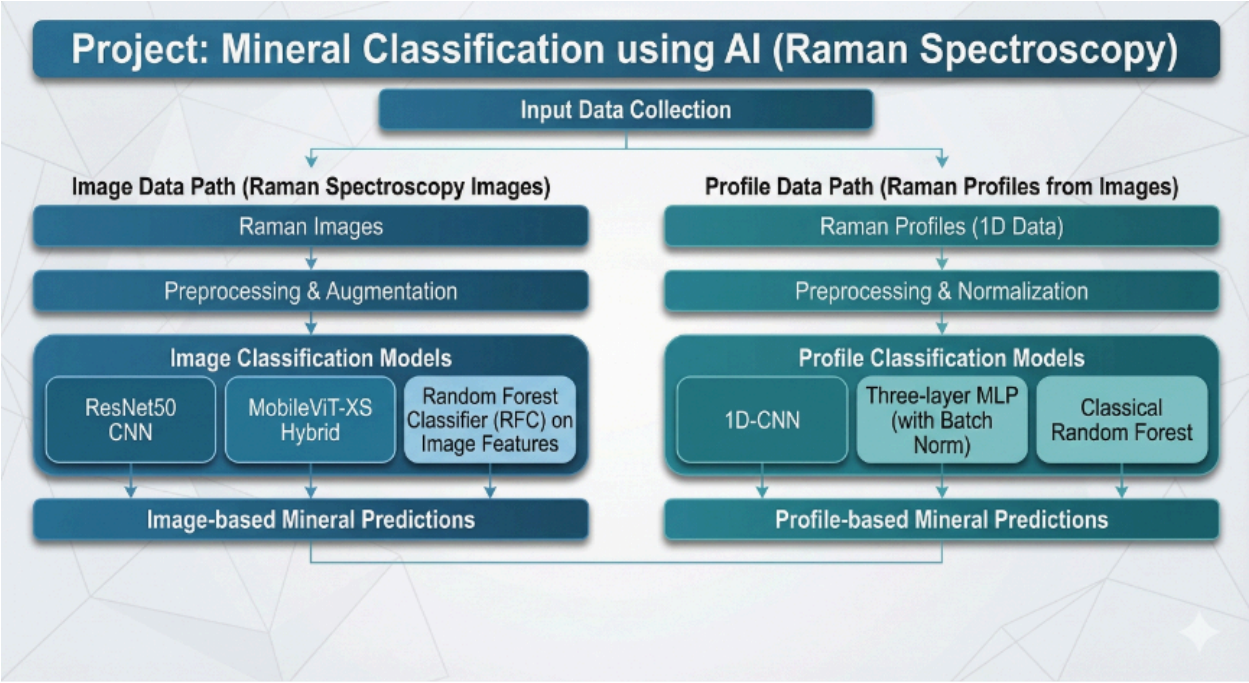
- **Image Data (30fps & 1fps):** All evaluated architectures: MobileViT-XS, ResNet50, and Random Forest on flattened images achieved a perfect **100% accuracy** on both the 30fps and 1fps datasets. This indicates that the spectral images contain highly distinct and robust features for each mineral class. **MobileViT-XS** is the superior model for image-based classification, as it maintains perfect accuracy while offering a better balance of speed and low memory usage, making it ideal for mobile or edge deployment compared to the heavier ResNet50 or the high RAM usage of the Random Forest approach.
- **Profile Data (30fps):** On the high frame-rate vector profile data, the statistical and simple deep learning models performed best: **Random Forest** led with **~96.3% accuracy**, closely

followed by **MLP (~95.7%)**. The 1D-CNN trailed significantly at ~85.4%, suggesting that complex spatial features are less critical or harder to learn at this frame rate compared to the statistical patterns leveraged by Random Forest and MLP.

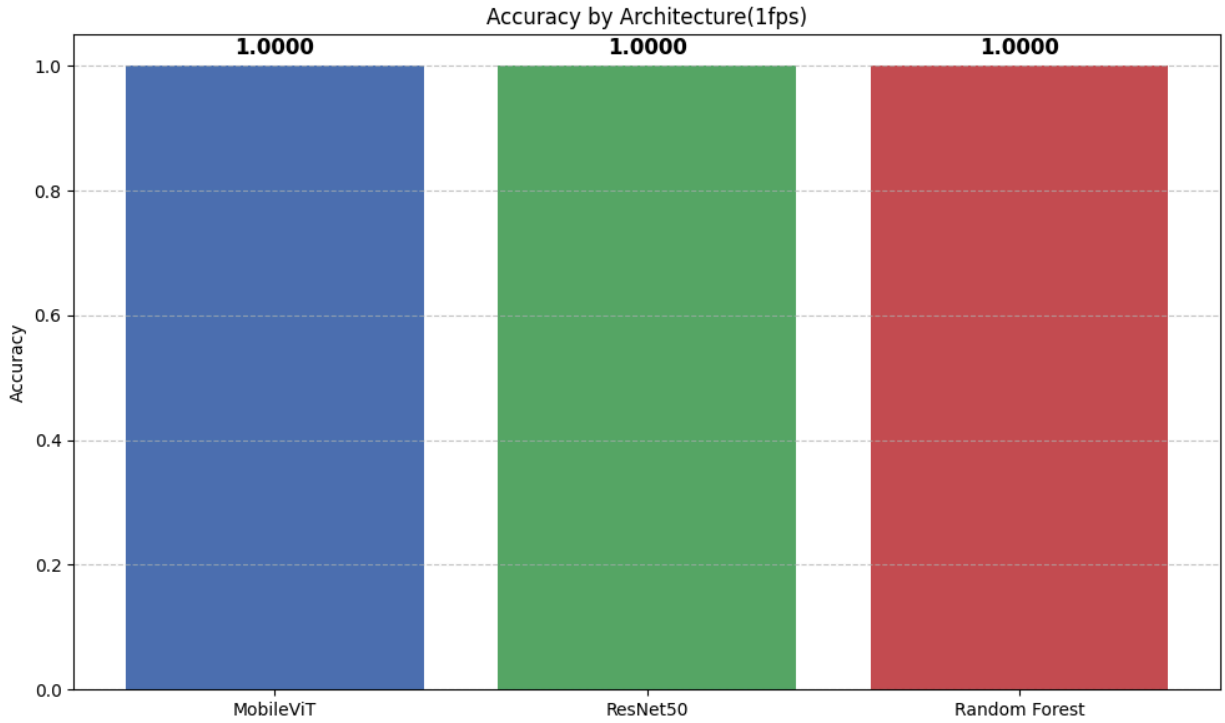
- **Profile Data (1fps):** Reducing the frame rate to 1fps significantly improved results across all profile-based models. **Random Forest (~99.9%)** and **MLP (~99.7%)** achieved virtually perfect classification accuracy, demonstrating that the 1fps profile data contains cleaner or more distinct signal patterns. The 1D-CNN also showed immense improvement, reaching ~99.5%.

V. VISUALISATIONS

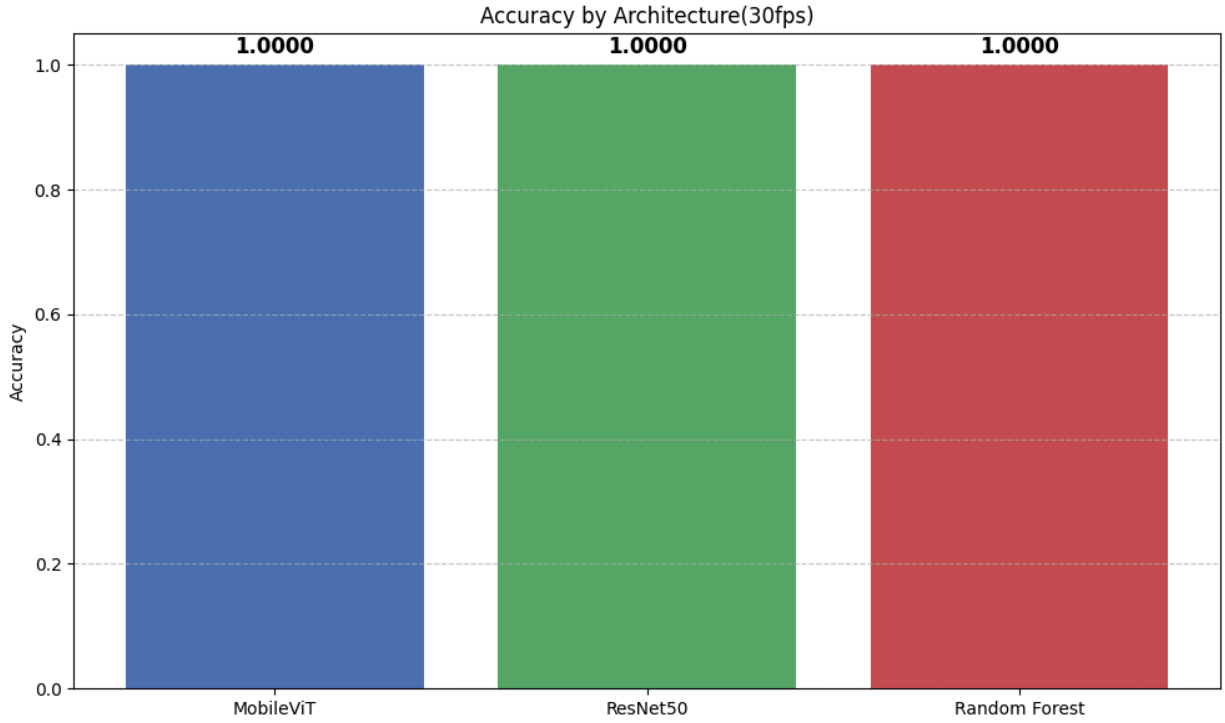
1. Overall Project Overview



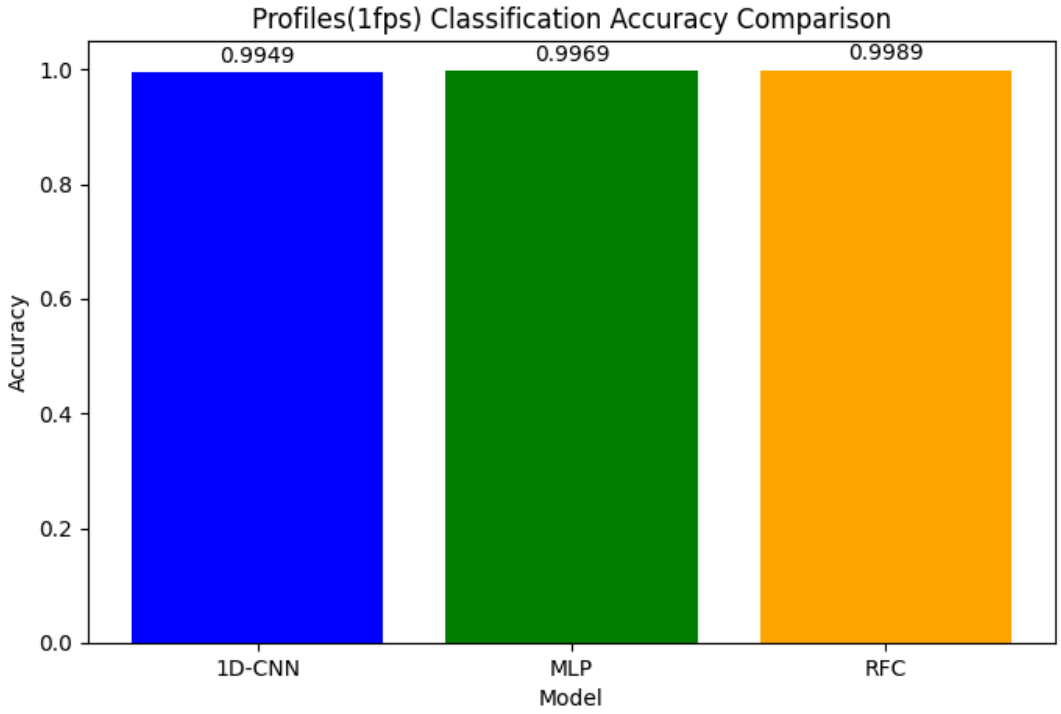
2. Accuracy for Images(1fps)



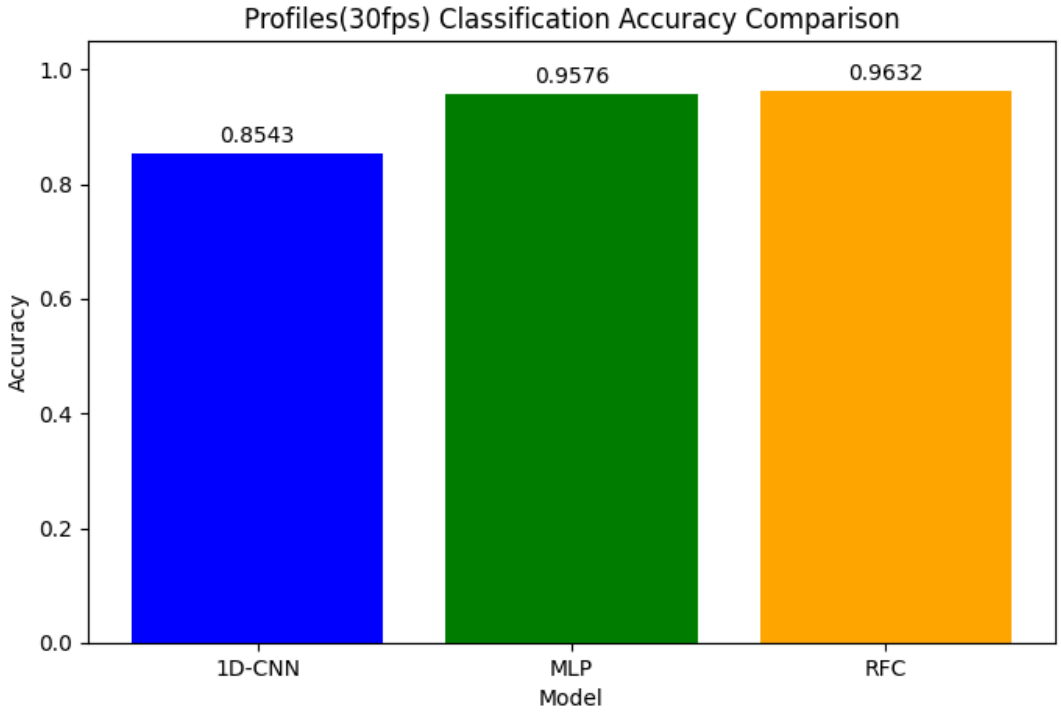
3. Accuracy for Images(30fps)



4. **Accuracy for Profiles(1fps)**



5. **Accuracy for Profiles(30fps)**



V. CONCLUSION

This project demonstrates that Artificial Intelligence can effectively classify mineral species using Raman spectroscopy. The choice of the optimal model depends on the operational constraints of the deployment environment, specifically the data modality and the required processing speed (FPS).

- **Spectral Images** consistently provided the most robust data, enabling **100% accurate classification** across all tested models (ResNet50, MobileViT-XS, Random Forest) at both 30fps and 1fps. The **MobileViT-XS** architecture is the recommended choice for image-based classification due to its superior balance of accuracy, speed, and memory efficiency.
- **Normalized Spectral Profiles** offer high computational efficiency and respectable accuracy. While the 30fps profiles achieved respectable performance of ~96% with Random Forest and MLP, the **1fps profiles** proved to be nearly as robust as the image data, achieving **near-perfect accuracy** (~99.9% with Random Forest and ~99.7% with MLP). For scenarios requiring extremely high computational efficiency and low resource usage, especially with lower-frame-rate or pre-processed profile data, **Random Forest** and the **MLP** are the recommended, highly-accurate, and resource-friendly models.

ACKNOWLEDGMENT

I would like to thank my supervisor, Sai Puneeth Reddy Gottam, for his guidance throughout this project. Finally, I would like to thank the developers of open-source libraries such as PyTorch, Scikit-Learn, and timm.

I utilized Large Language Models to assist in structuring this report and refining the project code.
Example Prompts:

- "Add comments to this code and verify that it works correctly."
- "Rewrite the methodology and results sections in IEEE format while maintaining technical accuracy."
- “Create a flowchart image of the Mineral Classification using AI project”