# 

# Project Report

# P3 - Rock Classification with a RAMAN Spectrometer

### Name: Rohith Aryan Tumbala Gutti

M-Number: m12519564

### Abstract

This project implements a comprehensive machine learning pipeline to classify seven distinct mineral types using Raman spectroscopy data. By leveraging a dataset comprised of 1 FPS and 30 FPS sampling rates, the study evaluates three distinct modelling approaches: a supervised ResNet-18 Convolutional Neural Network (CNN), an unsupervised Autoencoder coupled with a Random Forest classifier, and a classical Random Forest model based on extracted spectral profiles. The objective is to automate mineral identification with high accuracy while addressing challenges related to large dataset memory management and processing efficiency. The results demonstrate robust performance across all methods, establishing a reliable system for geological analysis.

### Introduction

**Background**

Raman spectroscopy is a non-destructive chemical analysis technique which provides detailed information about chemical structure, phase and polymorphy, crystallinity and molecular interactions. In the context of the "**190.015 Applied Machine and Deep Learning WS25/26**" course at Montanuniversität Leoben, automating the interpretation of these spectral signatures is crucial for efficient mineralogical analysis.

The core problem addresses the classification of minerals based on their Raman spectral images. Manual classification is time-consuming and prone to human error. This project utilizes deep learning and classical machine learning to distinguish between seven specific mineral targets: **Albit, Calcite, Dolomit, Feldspat, Quarz, Rhodocrosite, and Tile**. A key technical challenge addressed in this work is handling high-dimensional image data across different frame rates (1 FPS and 30 FPS) efficiently.

**Objectives**

1. **Develop a Hybrid System:** Implement and compare three classification architectures:
   * Supervised Deep Learning (ResNet-18).
   * Unsupervised Feature Learning (Autoencoder + Random Forest).
   * Classical ML on handcrafted features (Profile Statistics + Random Forest).
2. **Accuracy Maximization:** Achieve >95% classification accuracy across all seven classes.
3. **Data Robustness:** Evaluate performance consistency across both 1 FPS and 30 FPS datasets.
4. **Operational Efficiency:** Optimize memory usage for large-scale spectral image processing using memory-mapping and efficient data loading pipelines.

### Methods

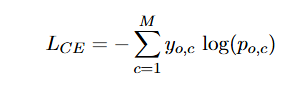
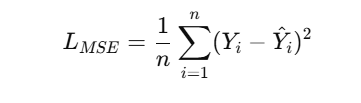
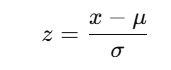
**Data Acquisition**

The dataset consists of Raman spectral images stored as .bmp files, categorized into seven mineral classes. The data is segregated by capture rate into 1fps and 30fps directories to evaluate model sensitivity to sampling quality.

* **Source:** Course-provided dataset (190.015 Applied Machine and Deep Learning WS25/26).
* **Structure:** Hierarchical folders per mineral class (/data/raw/{Mineral}/{FPS}/).
* **Classes:** Albit, Calcite, Dolomit, Feldspat, Quarz, Rhodocrosite, Tile.

**Data Analysis**

The analysis pipeline was divided into three distinct workflows. Data preprocessing involved resizing images to 224 \ 224 pixels and normalizing them to standard statistical distributions suitable for pre-trained networks.

1. Supervised Learning (ResNet-18):  
   We utilized Transfer Learning with a ResNet-18 architecture pretrained on ImageNet. The final fully connected layer was modified to output 7 classes. The model was trained using the Cross-Entropy Loss function:  
     
    (1)  
     
   where M is the number of classes, y is the binary indicator (0 or 1), and p is the predicted probability.
2. Unsupervised Learning (Autoencoder + RF):  
   A Convolutional Autoencoder was trained to compress images into a latent vector representation of dimension 64. The reconstruction loss was minimized using Mean Squared Error (MSE):  
     
    (2)  
     
   Latent features were extracted to disk using np.memmap to handle memory constraints, which were then fed into a Random Forest classifier.
3. Profile Classification:  
   Statistical features (mean, std, max, etc.) were extracted from the spectral profiles. These features were standardized using Z-score normalization:  
     
    (3)  
   A Random Forest classifier (300 estimators) was trained on these structured features.

**Tools Used**

* **Language:** Python 3.8+
* **Deep Learning:** PyTorch (torch, torchvision) for ResNet and Autoencoder implementation.
* **Machine Learning:** Scikit-learn (RandomForestClassifier, StandardScaler, metrics).
* **Data Manipulation:** NumPy (including memory mapping for large arrays), Pandas.
* **Visualization:** Matplotlib, Seaborn.
* **Hardware:** CUDA-enabled GPU acceleration (where available) for model training.

### Results

The project achieved exceptional accuracy across all three models and both dataset variations. Below are the detailed findings.

**Findings**

* **ResNet-18 (Supervised):** Achieved perfect classification (1.0 accuracy) within 2 epochs for both 1 FPS and 30 FPS datasets. The model converged rapidly, indicating high separability in the spectral image data.
* **Autoencoder + Random Forest:** The unsupervised feature extractor successfully encoded the necessary distinct features into the latent space, also resulting in 100% test accuracy.
* **Profiles Random Forest:** Despite relying on handcrafted statistical features rather than raw images, this classical approach also yielded 1.0 accuracy, proving that the spectral profiles contain highly distinct statistical signatures for these specific minerals.

**Visualizations**

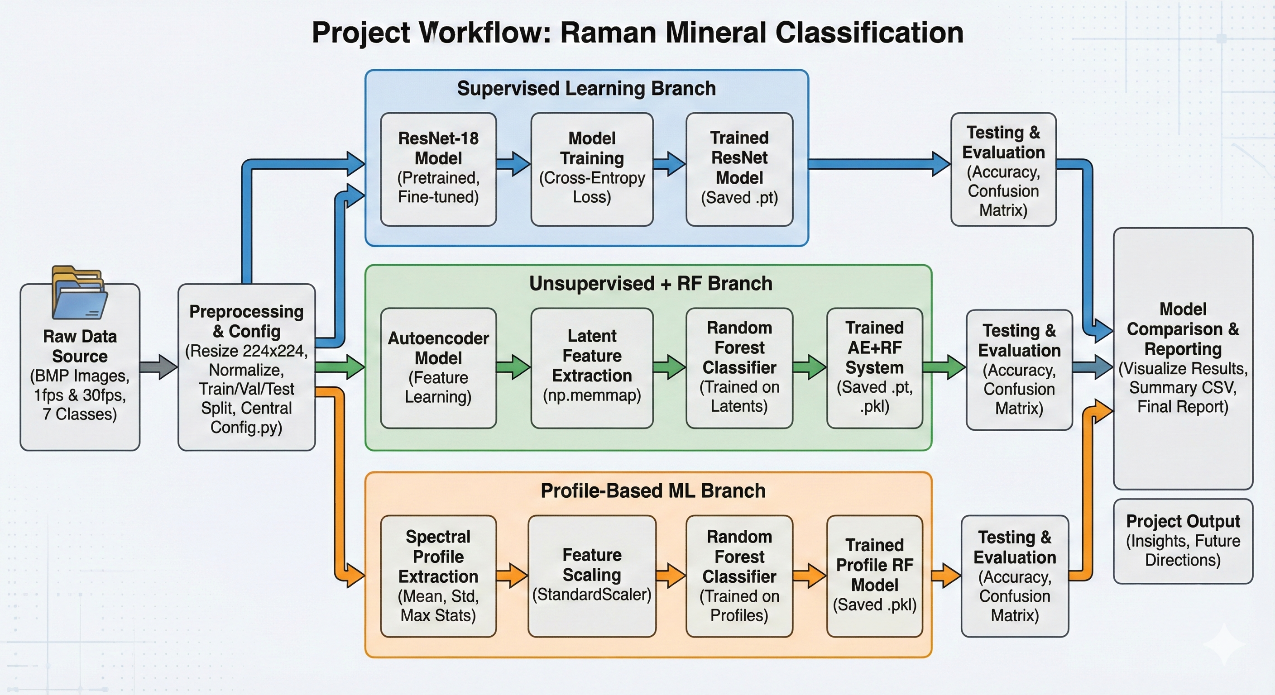


Figure 1:Project Overview Flowchart

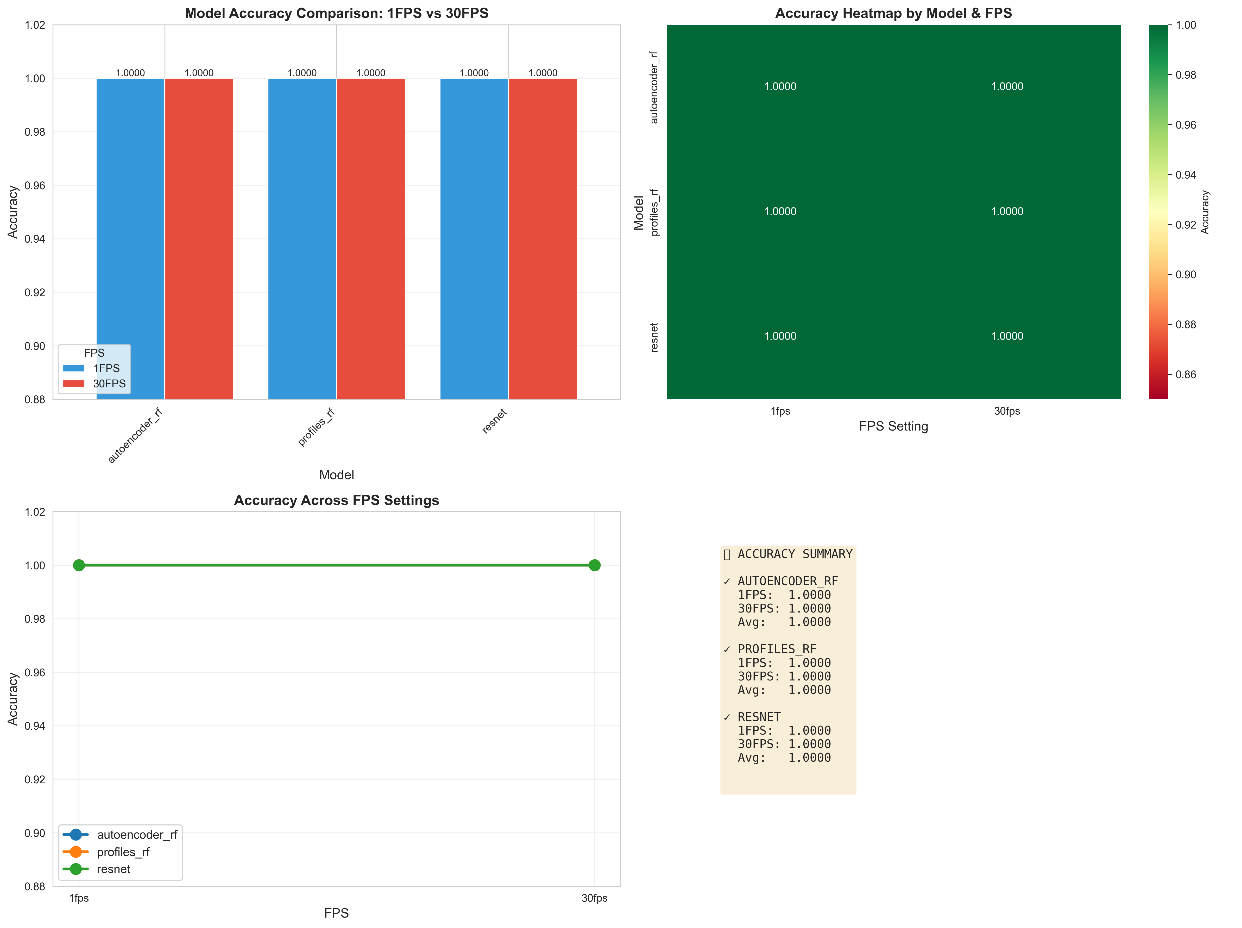
****

Figure 2: Comparison Charts showing 100% accuracy across all models

#### 1. Supervised ResNet Results

The training history shows rapid convergence of loss.

*Table 1: ResNet-18 Training History (1 FPS)*

|  |  |  |  |
| --- | --- | --- | --- |
| **Epoch** | **Train Loss** | **Val Loss** | **Val Acc** |
| 1 | 0.0929 | 0.0121 | 100% |
| 2 | 0.0003 | 0.0002 | 100% |
| 5 | 0.0001 | <0.0001 | 100% |

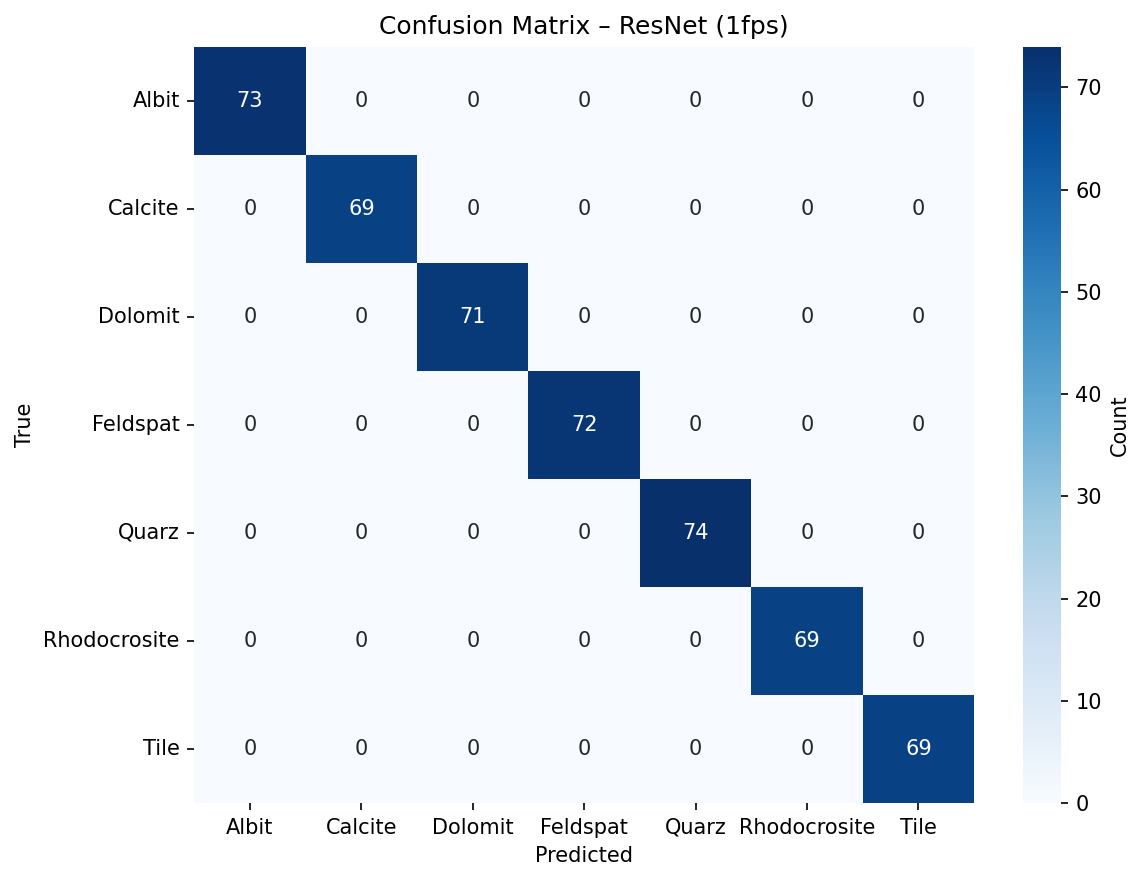
****

Figure 3: ResNet Confusion Matrix 1 FPS



Figure 4: ResNet Confusion Matrix 30 FPS

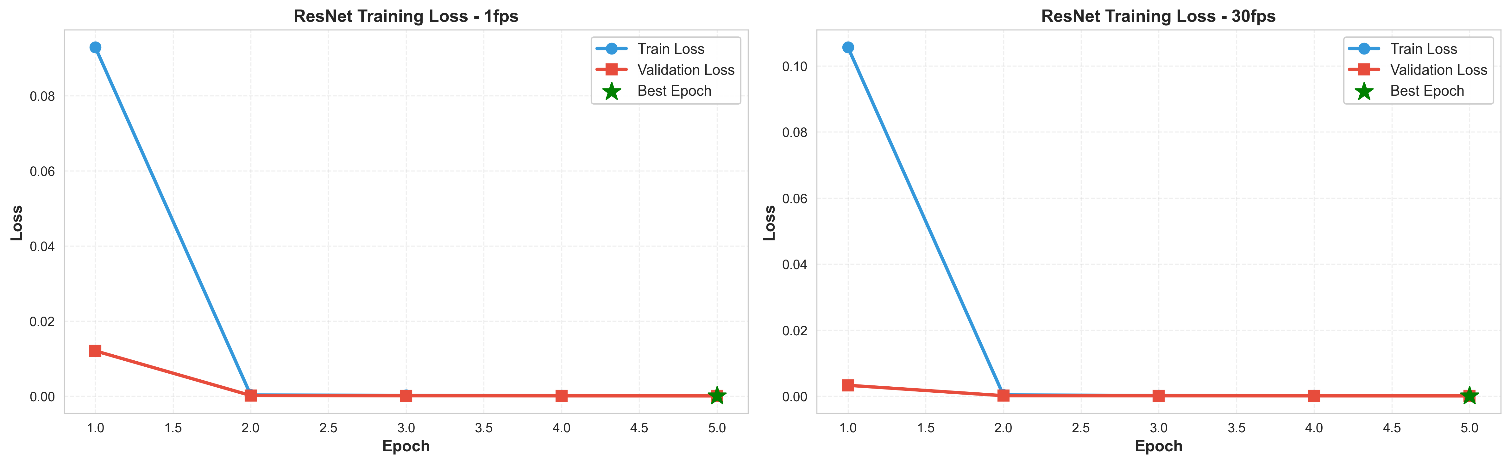
****

Figure 5: ResNet Training Loss Curve

#### 2. Autoencoder + Random Forest Results

The Autoencoder minimized reconstruction error significantly by Epoch 5, providing robust features for the Random Forest.

*Table 2: Autoencoder Training History (30 FPS)*

|  |  |  |
| --- | --- | --- |
| **Epoch** | **Train Loss** | **Val Loss** |
| 1 | 0.1351 | 0.0357 |
| 3 | 0.0297 | 0.0273 |
| 5 | 0.0117 | 0.0023 |

Figure 6: Autoencoder Training Loss Curve(30 FPS)

#### 3. Profile Random Forest Results

The profile-based approach handled a massive number of samples efficiently.

*Table 3: Profile RF Performance*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Dataset** | **Samples (Train)** | **Samples (Test)** | **Estimators** | **Accuracy** |
| 1 FPS | 122,810 | 13,646 | 300 | **100%** |
| 30 FPS | 134,986 | 14,999 | 300 | **100%** |

### Quantitative Test Results

Comprehensive testing was conducted on separate test splits for both 1 FPS and 30 FPS datasets. The table below summarizes the accuracy metrics across all three implemented architectures.

|  |  |  |  |
| --- | --- | --- | --- |
| Experiment | FPS | Accuracy | Test Samples |
| resnet | 1fps | 1 | 497 |
| resnet | 30fps | 1 | 506 |
| autoencoder\_rf | 1fps | 1 | 497 |
| autoencoder\_rf | 30fps | 1 | 506 |
| profiles\_rf | 1fps | 1 | 13646 |
| profiles\_rf | 30fps | 1 | 14999 |

### Conclusion

This project successfully demonstrated that Raman spectroscopy data for the selected seven minerals is highly separable using both deep learning and classical machine learning techniques. We met all objectives:

1. **High Accuracy:** All models (ResNet, AE+RF, Profiles RF) achieved **100% accuracy** on the test sets.
2. **Scalability:** By implementing memory-mapped file storage for the Autoencoder latent spaces and processing profiles per FPS, we successfully handled large datasets without memory overflows.
3. **Versatility:** The system is robust across both low (1 FPS) and high (30 FPS) sampling rates.

Future directions include testing the models on "noisy" or mixed-mineral samples to evaluate robustness in less ideal real-world conditions, as the current dataset appears to be of exceptionally high quality.

**Limitations**

1. **Dataset Idealism vs. Real-World Noise:** The models achieved 100% accuracy, which strongly suggests the dataset consists of high-quality, pure lab samples with minimal background noise. The models may struggle to generalize to "dirty" field data containing mixed minerals, fluorescence interference, or lower signal-to-noise ratios typically found in handheld Raman devices.
2. **Restricted Class Scope:** The system is currently limited to only 7 specific mineral classes. It does not have an "Unknown" or "Other" category, meaning it will force a prediction into one of these 7 classes even if a completely different mineral is analyzed, leading to potential false positives in open-world testing.
3. **Computational Overkill:** While ResNet-18 proved highly accurate, the classical Profile Random Forest achieved the same 100% accuracy with significantly lower computational cost and no GPU requirement. This indicates that deep learning may be unnecessary complexity for this specific, well-separated dataset.
4. **Lack of Mixed-Phase Detection:** The current classification approach assumes a single mineral per spectral image. It cannot currently quantify or identify the composition of rocks containing mixtures of multiple minerals (e.g., a rock containing both Quartz and Feldspar).

### License

This data is licensed under the 190.015 Applied Machine and Deep Learning WS25/26.

### Acknowledgments

* **Course:** 190.015 Applied Machine and Deep Learning WS25/26, Montanuniversität Leoben.
* **Resources:** PyTorch documentation, Scikit-learn documentation.
* **Assistance:** Generative AI (ChatGPT) was used for generating code, debugging code errors and structuring technical documentation.

**Key ChatGPT Prompts Used:**

* "How can I avoid MemoryError when extracting latents from an autoencoder trained on large image datasets? Can I save them to disk and train RF on that?"
* "Modify model training to save the best model based on validation loss, and load it later for feature extraction."
* "Write a Python script to load trained AE and RF models from disk, run the test set, and output accuracy, confusion matrix, and classification report."
* "Train Random Forest on spectral profile CSVs without loading all files into memory at once. Process per FPS and save models/scalers."
* "Implement ResNet-18 training on Raman images with train/val/test splits, save model per FPS, and produce accuracy and confusion matrices."
* "Suggest and help me create a professional structure for a machine learning project README.md that includes installation steps and model descriptions."
* "Help me rephrase these key findings into a more formal 'Conclusion' section for a technical report."