# Identifying Lunar Craters with Machine Learning

**Destiny Wilson, Tiffany Gao, WaTae Mickey**
**EEPS-DATA 1340 – May 10, 2025**
**Professor Karianne Bergen**

## Abstract

Craters hold the key to a wide variety of important geologic information used to understand the age and impact history of planets. The importance of this information extends beyond the planets themselves and places constraints on their solar systems and the universe as a whole. Counting and classifying craters has traditionally been a manual task, but this is a tedious and error-prone process. In this study, we present a machine learning approach to automate crater classification to reduce human error and increase the accuracy of the crater size-frequency distribution analysis. Using the Lunar Reconnaissance Orbiter dataset, consisting of 502 fresh craters, 869 old craters, and 3,629 images containing neither, we performed feature extraction and attained 768-dimensional embeddings. Dimensionality reduction was applied to the features, and we reduced the dimensionality, keeping 95% variance of our data. Several models were tested on the dataset, with the two most notable being SVM and Random Forest. The accuracy of SVM was 80%, while Random Forest achieved 82% with higher overall precision. The machine learning model positively identified craters effectively, but struggled to detect all craters in the dataset, highlighting areas for future improvement. However, it was able to identify nearly all images without any craters in them at all.

## 1. Introduction

Determining the age of planetary surfaces is an important task in planetary science, and crater counting has long been an integral method by which planetary scientists date a planet's age to illustrate the age of its surrounding solar system and, by extension, the universe. Formed when a meteoroid collides into the surface of a planet, craters give insight into a surface's age through crater-size frequency distribution. Planets with greater crater-size frequency distributions are typically older than those with smaller distributions, thus providing insight into other planetary features (Benedix et al., 2). Traditionally conducted through a manual inspection process in which scientists count the number of visible craters from images of planetary surfaces, this method is highly flawed because of potential errors influencing total crater counts. Such an error is even more inevitable due to the lack of visibility afforded by low-resolution images of planetary bodies and the consequential difficulty of analysis for scientists.

This project focuses on developing a machine learning model that automates the classification process of lunar craters as either old or young based on their morphological features. The main goal of this project is to create a model that can use images to accurately differentiate between these two categories, along with those that fall into either category. By training the model with features extracted from lunar imagery, we aim to provide a tool that can assist planetary scientists in their efforts to refine our understanding of the Moon's geological history.

## 2. Data

The dataset from the Lunar Reconnaissance Orbiter Imagery for LROCNet Moon Classifier was used for this project (Dunkel, 2022). Comprised of 50% resolution imagery from

the Lunar Reconnaissance Orbiter's Narrow Angle Cameras (NACs), this dataset has 277x277 small cutouts from larger lunar images with features categorized based on the following 3 class sets:

1. Fresh crater (impact ejecta)
2. Old crater
3. None

Although the images appear grayscale, they are stored in RGB format and contain three color channels: Red, Green, and Blue. For craters to be classified into the (1) Fresh crater class, they must have ¾ of circumference visible in an image, while images classified into the (2) Old crater class must have a diameter greater than or equal to one-tenth of the image width, and images in the (3) None class depict craters that do not fit into the fresh or old crater classes. Additionally, the developers augmented the training set with horizontal and vertical flips, rotation, brightness adjustments, and weighted sampling for equal class weights. Of all 3,608 images included in the training set, 358 were used to train the Fresh crater class, 594 were used for the Old crater class, and 2,656 images were distributed across the four training sets for the None class. In addition to the training data, the dataset includes test and validation sets. The test set has 779 images, consisting of 89 fresh crater images, 157 old crater images, and 533 images without craters. Meanwhile, the validation set includes 55 fresh crater images, 118 old crater images, and 440 images without craters (Table 1). These validation and test sets were not used until after training our model to allow us to tune hyperparameters and assess model performance on unseen data.

| **Dataset Split** | **Fresh Crater Images** | **Old Crater Images** | **Non-Crater Images** |
|---|---|---|---|
| Training | 358 | 594 | 2,656 |

| | | | |
|---|---|---|---|
| Validation | 89 | 157 | 533 |
| Test | 55 | 118 | 440 |

*Table 1. Distribution of Images Across Dataset Splits*

## 3. Methods

Our general approach to this project begins by utilizing a pre-constructed and defined split of the crater image dataset (3,608 training, 779 test, and 613 validation samples). These subsets are stored in separate folders and processed independently to avoid data leakage. We use a pre-trained Vision Transformer (ViT) model to extract 768-dimensional image embeddings from the crater images for each subset. These embeddings represent our initial feature set and are saved as CSV files. We performed feature selection through Principal Component Analysis (PCA) on the training data. This ensures our validation and test sets remain completely untouched during feature engineering, allowing proper evaluation of how well our models generalize to unseen crater morphology. We will then compare identifying craters manually to the list of craters identified by the model using classification models like Support Vector Machines (SVM) and Random Forest.

### 3.1 Data Processing

Our dataset consists of grayscale images of lunar craters (each 277 x 277 pixels in size) previously labeled as either new or old based on crater freshness indicators—the most prominent indicators being ejecta brightness and morphology. Each image was then preprocessed using a pre-trained Vision Transformer (ViT) feature extractor from the transformers library. This feature extractor transformed the image into a fixed-length embedding vector of 768 dimensions. This means that the ViT model turns the whole image into a single vector with 768 entries or features, with each feature representing some abstract characteristic that the model “noticed” about the

image. The ViT captures high-level image features like shape, texture, and spatial patterns by building on lower-level pixel information through self-attention mechanisms.

After feature extraction, we applied dimensionality reduction to better visualize the dataset. First, we used Principal Component Analysis (PCA) on the 768-dimensional ViT embeddings while preserving 95% of the variance. The first two principal components (PC1 and PC2) accounted for 34.4% and 8.9% of the variance, respectively. While this captures only a portion of the total variance, the 2D projection revealed discernible clustering patterns between fresh/ejecta and old craters. Fresh craters with visible ejecta tend to occupy a distinct plot region, while old craters form a somewhat separate cluster. In contrast, the "None" class, which corresponds to background or ambiguous terrain, overlaps significantly with ejecta and old crater clusters (Fig. 2).

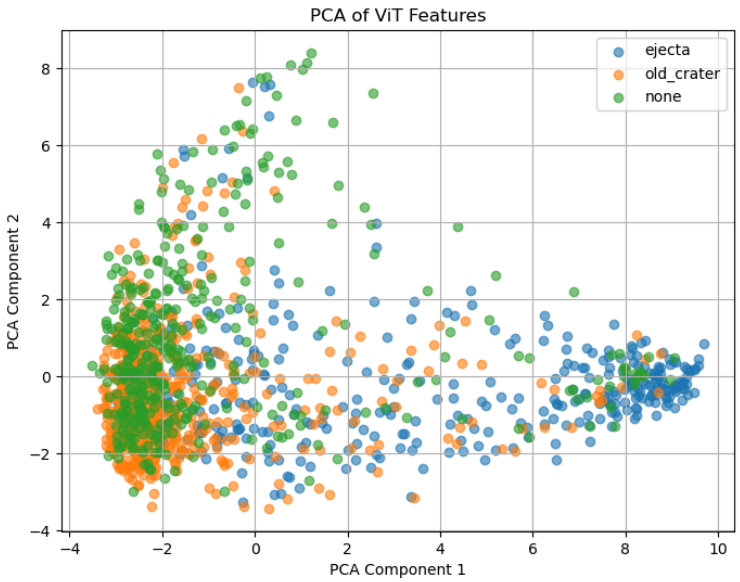


*Figure 1. PCA of ViT Features*

We also applied distributed stochastic neighbor embedding (t-SNE) for nonlinear dimensionality reduction and visualization. The t-SNE plot further emphasized the separation

between fresh/ejecta and old craters. Once again, the “None” class appears to be diffused across both the fresh/ejecta and old crater clusters (Fig. 2).

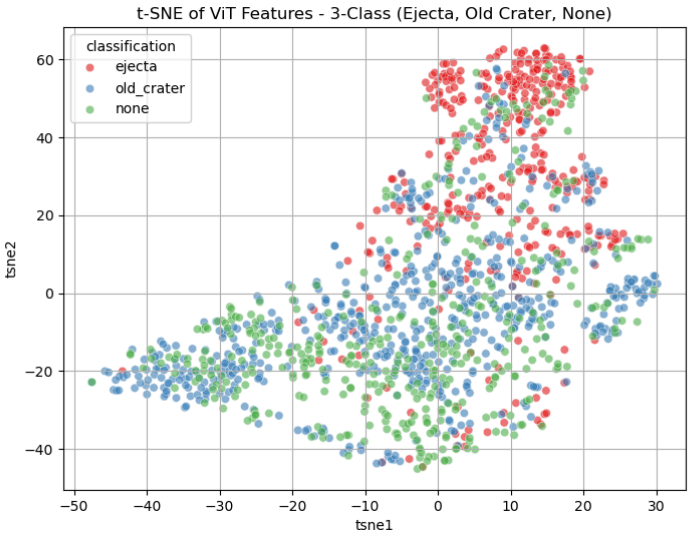


*Figure 2. t-SNE of ViT Features*

### 3.2 Classification Model Selection

We chose supervised classification models to distinguish between ejecta (fresh), old, and non-crater (background) classes. We selected K-Nearest Neighbors (KNN) as our baseline model due to its simplicity, interpretability, and minimal hyperparameter tuning. While KNN offered moderate performance, it struggled to distinguish background from crater classes.

In addition to KNN, we explored several other classifiers to assess performance improvements with more complex or non-linear models. These included Random Forest Classifier, an ensemble-based, non-linear model that can better capture interactions between features and is robust to noise. We used 200 trees with class balancing enabled. Another classifier was Support Vector Machine (SVMs) with RBF kernel to assess a non-linear decision boundary in high-dimensional space.

Additionally, we tested Decision Trees and AdaBoost classifiers. AdaBoost was tested with both Logistic Regression and Decision Tree-based learners. The performance of these models and classifiers will be expanded on in the Model Analysis section.

Our project uses supervised learning to classify our data into the (1) fresh crater, (2) old crater, and (3) non-crater classes. Using KNN, SVMs, Random Forest, and AdaBoost as our primary machine learning algorithms, we assess each model's performance based on accuracy, precision, and recall. These metrics are produced for each classification class, with Class 0 representing images without lunar craters, Class 1 representing images with old craters, and Class 2 representing young craters (ejecta).

### 3.3 Model Evaluation and Visualization

To evaluate model performance, we computed classification metrics including precision, recall, F1-score, and accuracy using classification_report from sklearn.metrics. We examined the confusion matrix to assess how well the model distinguishes between old and new craters. We also examined the confusion matrix further in the model evaluation.

## 4. Results

### 4.1 Baseline Approach

Our baseline model was a KNN classifier with the n_neighbors hyperparameter set to 17. We derived our *n_neighbors* value by implementing the GridSearchCV hyperparameter tuning method with the *n_neighbors* value range set from 1 to 21. Based on this range, 17 was identified as the optimal *n_neighbors* value. Our KNN model achieved 78% accuracy and a precision above 75% across all three lunar classification classes. Thus, we selected it as our baseline

because of the simplicity of implementation and reasonable performance with minimal hyperparameter tuning.

After identifying KNN as our baseline, we attempted more complex models equipped to handle high-dimensional data, the first of which was non-linear SVM. We selected SVMs because they could utilize their non-linear decision boundary to better recall distinctions between (Class 1) old and (Class 2) young craters. Using GridSearchCV with *C* hyperparameter values 0.1, 1.0, 10, 20, 50, and 70, we identified 50 as the best-performing value of the *gamma* hyperparameter rather than scale, 0.1 or 0.001.

Subsequently, we used Random Forest because of its ability to handle large amounts of features with high accuracy and the algorithm's positive performance on similar research, namely "A Machine Learning Approach to Crater Classification from Topographic Data" by Qiangyi Liu et al. We concluded that Random Forest's ability to create numerous trees and observe different subsets of the data while considering different features would have a positive performance outcome. Additionally, we identified radial basis function (RBF) as the best performing type for the *kernel* hyperparameter compared to a linear, polynomial, or sigmoid type, and 0.01 as the performance outcome. Once again, we applied GridSearchCV to tune the model hyperparameters, identifying 200 as the appropriate value for *n_estimators* compared to 100 and 500, and 20 as the best performing *max_depth* over None, 10, and 20, and sqrt, as the appropriate value for *max_features* over log2.

Based on insights from Chen et al.'s research, "Impact crater recognition methods: a review," we attempted AdaBoost with Decision Tree as a base estimator because it was an alternative ensemble machine learning method that often performs with similar classification accuracy to Random Forest (Chen et al., 13). However, its reliance on a base estimator and a

greater number of parameter requirements presented a more challenging hyperparameter tuning process. Our selection of Decision Tree for the *estimator* hyperparameter was informed by the model's non-linear decision boundary, which we hoped would perform well upon encountering the non-linear relationship of crater features. Results from GridSearchCV revealed that a *max_depth* value of 5 would perform better than values 1, 2, 3, and 10 for the Decision Tree estimator. Meanwhile, an *n_estimators* value of 100 would produce better results than 50 and 200, and a *learning_rate* value of 1.0 would exceed the performance of a 0.01 or 0.1 hyperparameter value. After running our models on the test data with these selected hyperparameters, we attained the model results in Table 2.

| **Model** | **Accuracy** | **Class 0** | | **Class 1** | | **Class 2** | |
|---|---|---|---|---|---|---|---|
| | | **Recall** | **Precision** | **Recall** | **Precision** | **Recall** | **Precision** |
| K-Nearest Neighbors | 75% | 91% | 77% | 21% | 56% | 72% | 76% |
| Non-Linear Support Vector Machines (SVM) | 83% | 89% | 83% | 48% | 53% | 76% | 90% |
| Random Forest | 82% | 96% | 79% | 43% | 82% | 64% | 94% |
| AdaBoost | 75% | 83% | 80% | 52% | 50% | 68% | 81% |

*Table 2. Model Performance on Test Data*

### 4.2 Model Analysis of Results

According to the results in Table 2, SVMs attained the highest overall accuracy at 83% and best recall for Class 2 at 76%. Meanwhile, Random Forest achieved the highest recall for Class 0 and precision for Classes 1 and 2. AdaBoost with Decision Tree Logistic as the estimator obtained comparable accuracy results to our KNN baseline, with 75% and the highest recall for

Class 1. As our baseline, KNN reflected common trends across our models by performing within the range of accuracy demonstrated by our other selected algorithms, illustrating a trend of high recall and precision for Classes 0 and 2, with comparatively poor performance on Class 1. The Confusion Matrices depicted in Table 3 reflect these trends.

| | **KNN** | | |
|---|---|---|---|
| | Predicted | | |
| | Class 0 | Class 1 | Class 1 |
| True Class 0 | 80 | 4 | 4 |
| True Class 1 | 19 | 5 | 0 |
| True Class 2 | 5 | 0 | 13 |

| | **Random Forest** | | |
|---|---|---|---|
| | Predicted | | |
| | Class 0 | Class 1 | Class 1 |
| True Class 0 | 81 | 2 | 1 |
| True Class 1 | 12 | 9 | 0 |
| True Class 2 | 9 | 0 | 6 |

*Table 3. Confusion Matrix for Baseline vs. Random Forest*

Comparing Class 0 and Class 1 metrics reveals that all models performed better on Class 1 than Class 0. This discrepancy suggests that the models exhibit bias towards Classes 0 and 2, the non-crater and young crater classes, reflecting a failure to distinguish old and young craters. While all models have room for improvement, we identified Random Forest as our current best-performing model due to its demonstration of 82% precision in Class 1, a score exceeding the performance of every other model for the class by nearly 30%. While its accuracy is slightly below that of SVM (83%), its consistent precision across the three classes illustrates its ability to capture the complex distinctions between aged craters. Our proposed models performed consistently better among all class metrics compared to our baseline KNN, suggesting that fine-tuning the Random Forest model in future work would result in better performance, especially for Class 1 precision and recall.

## 5. Discussion & Conclusions

Throughout this project, we encountered several challenges, both technical and conceptual, that shaped our approach to crater classification. One of the central issues was distinguishing between old craters (Class 1) and fresh craters with visible ejecta (Class 2). Unlike fresh craters that exhibit high albedo and defined ejecta blankets, old craters often blend into the lunar surface due to erosion and degradation. This makes Class 1 particularly difficult to distinguish from non-crater terrain (Class 0; Fig. 1 and 2). Our general assumption that all labeled examples were correctly identified by the original dataset providers could have contributed to this challenge and biased results in edge cases or complex terrain. In future adaptations of our work, assessing the dataset for mislabeling would be useful to mitigate errors and improve model accuracy.

Another shortcoming is that we did not incorporate K-Means or other clustering methods, which may have helped in unsupervised grouping of morphologically similar craters before classification. Research conducted by Huang et al. (1998) proposes extensions to K-Means that may be appropriate for our high-dimensional, feature-based crater data. However, we did not implement K-Means in this phase due to time constraints, implementation complexity, and prioritization of supervised learning models. Moreover, our application of hyperparameter tuning methods like GridSearchCV offered an intuitive avenue to optimize hyperparameters for all models, but we acknowledge that more rigorous validation may have provided better insight into generalizability.

While our results present significant model training progress, there is still room for improvement involving our model training metrics for our selected classification models and opportunities for further research and implementation of other algorithms. Addressing these

shortcomings will not only enhance model performance but also contribute to a more accurate and reliable framework for automated lunar crater classification.

## 6. References


Benedix, G. K., Lagain, A., Chai, K., Meka, S., Anderson, S., Norman, C., ... & Tan, T. (2020). Deriving surface ages on Mars using automated crater counting. *Earth and Space Science*, *7*(3), e2019EA001005.

Chen, D., Hu, F., Zhang, L., Wu, Y., Du, J., & Peethambaran, J. (2024). Impact crater recognition Methods: A review. *Science China Earth Sciences*, *67*(6), 1719-1742. 10.1007/s11430-023-1284-9.

Emily Dunkel. (2022). Lunar Reconnaissance Orbiter Imagery for LROCNet Moon Classifier (v1.0.0) [Data set]. Zenodo. https://doi.org/10.5281/zenodo.7041842.

Liu, Q., Cheng, W., Yan, G., Zhao, Y., & Liu, J. (2019). A Machine Learning Approach to Crater Classification from Topographic Data. *Remote Sensing*, *11*(21), 2594. https://doi.org/10.3390/rs11212594.

"Random Forest Hyperparameter Tuning in Python." *GeeksforGeeks*, 2025, www.geeksforgeeks.org/random-forest-hyperparameter-tuning-in-python/.

Zhexue Huang. (1998) Extensions to the k-Means Algorithm for Clustering Large Data Sets with Categorical Values. Data Min. Knowl. Discov. 2, 3 (September 1998), 283–304. https://doi.org/10.1023/A:1009769707641.