Year: 2024
Computer Vision Master

Classification of Volcanoes on Venus Using Classical Machine Learning Algorithms

Isabelle Rodrigues Vaz de Melo[1], Marcos Roberto Chindelar de Oliveira Leite[1], Bruno d'Almeida Franco[1], Leandro Wong Kang San[1]
[1]PUC-Rio, Rio de Janeiro, Brazil

Abstract

The work uses classic machine learning algorithms to address a classification problem regarding the presence of volcanoes on the surface of Venus, using images generated by the Magellan spacecraft. Preprocessing techniques are discussed, such as the use of Gaussian filters and Wavelets. Additionally, intrinsic augmentation of the images is performed to balance the classes, avoiding imbalance in classifier performance. The models used were SVM, Decision Trees, XGBoost, and KNN. A final result of 95% recall on the test set was achieved, demonstrating the effectiveness of the techniques used.

Keywords: Machine Learning, Computer Vision, Image Processing

1 Introdução

The Magellan mission, launched by NASA on May 4, 1989, was one of the most significant planetary exploration initiatives of its time. Its primary objective was to map the surface of Venus using Synthetic Aperture Radar (SAR) due to the impossibility of direct visual observation caused by the dense cloud layer enveloping the planet. Magellan mapped approximately 98% of Venus's surface with high resolution, revealing complex geological formations such as mountains, volcanic plains, impact craters, and extensive tectonic deformation regions. This data was crucial for advancing the understanding of Venus's geology and evolution, as well as providing valuable insights into the formation and dynamics of rocky planets.

With the advent of artificial intelligence (AI) and computer vision, space exploration is undergoing a technological revolution. AI is capable of processing large volumes of data, such as those generated by the Magellan mission, significantly faster and more efficiently than the techniques available at the time. This enables detailed, real-time analysis of the geological and atmospheric characteristics of planets, allowing the identification of patterns that might go unnoticed by traditional methods.

In addition to data analysis, AI plays a critical role in the autonomy of space missions. In extreme environments, such as the surface of Venus or Mars, where conditions are challenging and communication with Earth can be limited or delayed, the ability of spacecraft to make autonomous decisions is essential. AI can, for example, identify safe landing areas, avoiding dangerous terrain, and adapt the mission in response to unforeseen changes in environmental conditions. Complementarily, computer vision allows probes and rovers to navigate the terrain accurately, collecting essential data and avoiding obstacles.

The current work focuses on the use of classical machine learning algorithms to perform the supervised classification task. The presence or absence of a volcano is classified, thus optimizing the probe's work for faster inference applications and helping to avoid potential damage in the event of landings in areas considered hazardous, which could harm probes or rovers that may enter the planet's surface.

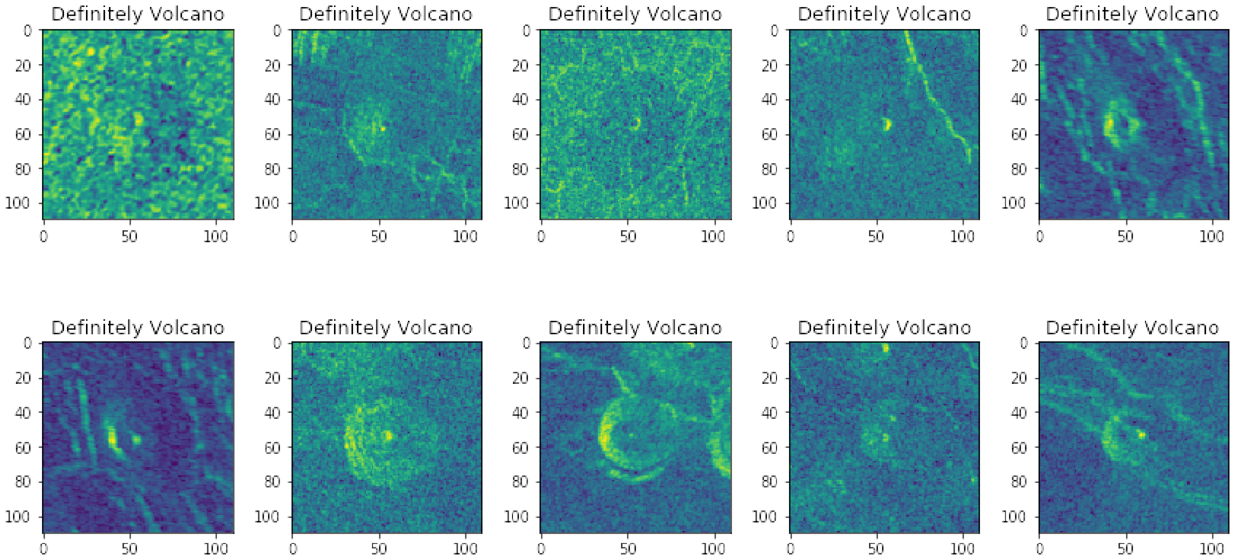

Figure 1: Examples from the dataset that contain volcanoes.

2 Data Analysis

The official data can be found in the NASA's official repository[1]. However, for this work, a processed version of the dataset was used, available on Kaggle: `https://www.kaggle.com/datasets/fmena14/volcanoesvenus`. The original images, with a resolution of 75 meters per pixel, were cropped into smaller sections and converted to grayscale, with a resolution of 120×120 pixels. This processing facilitates a better interpretation of each image by reducing the amount of encoded information and increasing the volume of data available for machine learning models.

2.1 Description

In this section, a detailed analysis of the provided data is presented, which is available in training and test sets. The input data is stored in two `csv` files. These files contain flattened monochromatic images with dimensions of 110×110 pixels, where each pixel has a value ranging from 0 to 255. Each flattened image is represented in a single row of data, with 12,100 columns representing the pixels. These images may show the presence of multiple volcanoes or, in some cases, no evidence of volcanoes. Figure 1 shows an example of data samples that contain volcanoes.

The files containing the target values are also provided for both training and validation.

[1] `https://pds-geosciences.wustl.edu/missions/magellan/`

These files include four columns that describe the characteristics of the identified volcanoes, as detailed below:

- **Volcano**: This column indicates the presence of volcanoes in the image, with a value of 1 if volcanoes are present and 0 if they are absent. For images labeled with 0, the following three characteristics are filled with NaN values.
- **Type**: This column classifies the probability of an object in the image being a volcano according to the following types:
  - 1 = Definetly a volcanoe
  - 2 = Probably a volcanoe
  - 3 = Possibly a volcanoe
  - 4 = Only a visible depression
- **Radius**: Refers to the radius of the volcano located at the center of the image, measured in pixels.
- **Number Volcanoes**: Indicates the number of volcanoes present in the image.

For images that contain volcanoes, it is generally observed that the volcano is centered in the image. It is important to note that the ”ground truth” associated with this data is not absolute. Since the Magellan mission does not allow for the physical presence of researchers on Venus and due to the limited quality of the images, the identification of volcanoes is not entirely precise and may not be completely unambiguous, even for experts.

Additionally, the data exhibit a significant imbalance, with fewer images containing volcanoes compared to those that do not (in the training set, 6000 images without volcanoes and 1000 images with volcanoes). This imbalance should be considered in subsequent analyses and models. Also, some images contain areas with spacecraft artifacts (pixels with a value of 0), which result from gaps in the acquisition or communication processes of the Magellan spacecraft. These regions can generally be disregarded in the analysis. Figure 2 shows examples of these regions.

2.2 Data Statistics

Before starting data preprocessing, it is important to find the main statistics associated with the dataset so that it is possible to guide the preprocessing step in the best possible

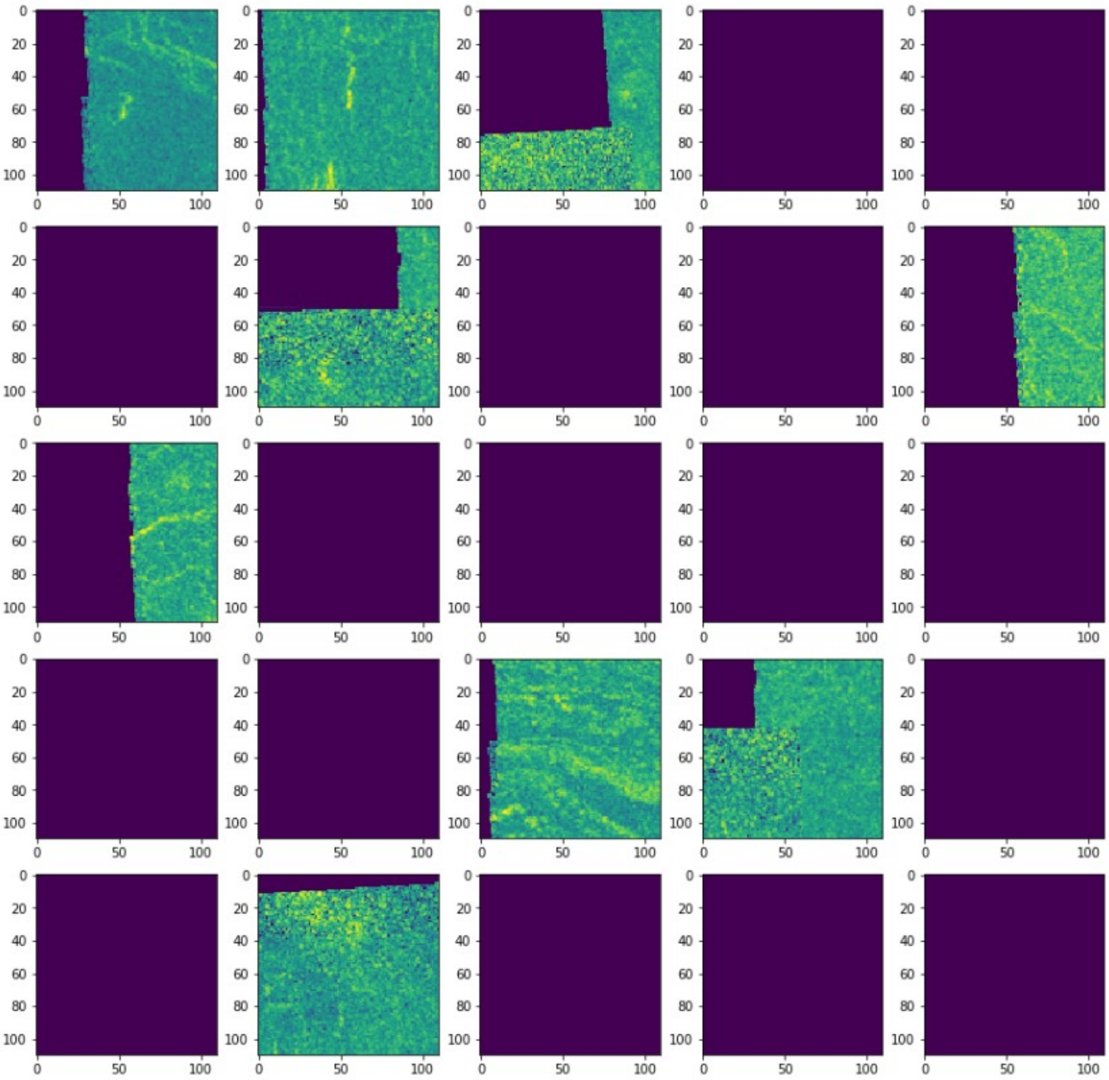

Figure 2: Examples of images with black regions.

way and check for any inconsistencies in the data that should be investigated.

Table 1 shows that, for the training set, the average pixel value is approximately 100 ± 26. Figure 3 presents the pixel histogram, showing that the pixel values are well distributed with few outliers. Also, for the 1000 images that contain volcanoes (represening $14,29\%$ of the whole dataset), approximately $10,50\%$ are definetly comproved volcanos.

Statistic	Value
Mean	100
Standard Deviation (Std)	26

Table 1: Mean and Standard Deviation values for the pixels in the training set.

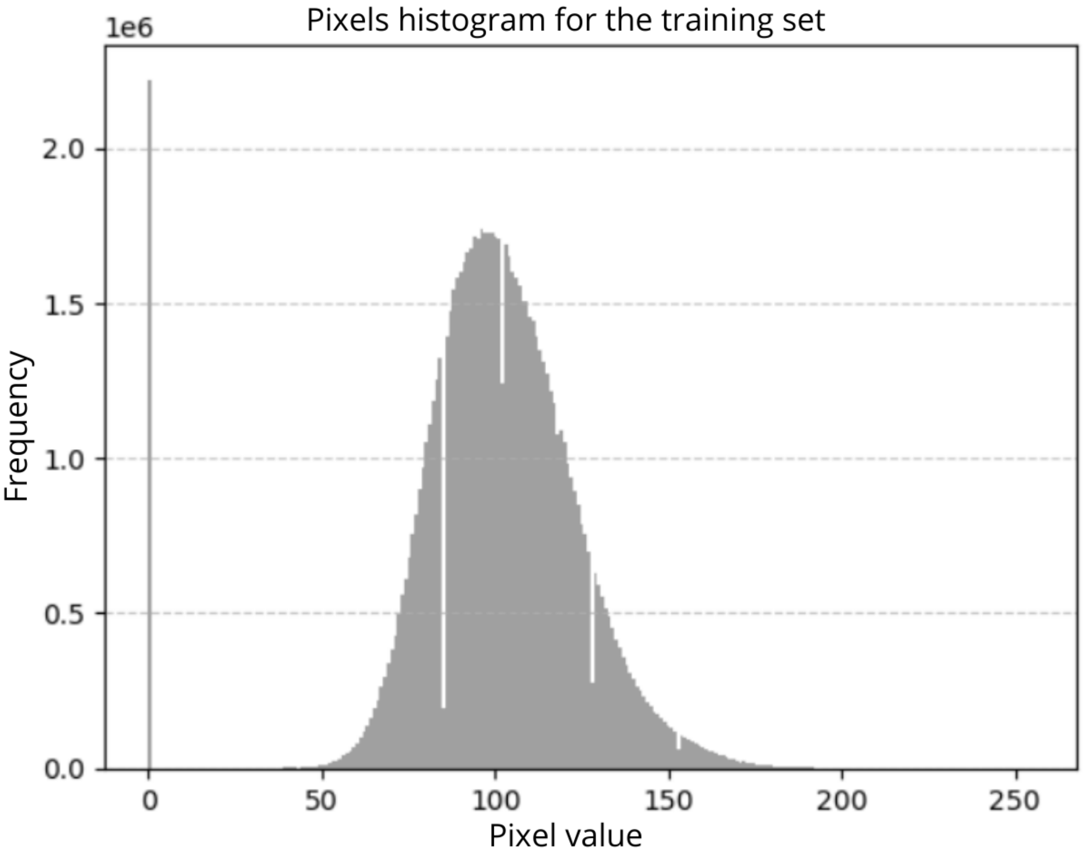

Figure 3: Pixel histogram for the training set.

2.3 Data Preprocessing

Data preprocessing is an essential step when using classical machine learning algorithms. These algorithms do not perform automatic feature extraction as in the case of convolutional networks, thus requiring that the images first undergo processing to extract the most important features for the task, reducing the complexity of the networks.

In this project, the input images are very noisy and contain an excessive amount of details that may hinder classifier performance. Therefore, a preprocessing strategy focused on removing excess details or noise was chosen. For this, there are two main proposed transformations on the data: the use of filtering with Gaussian blur or wavelets for denoising.

It is important to mention that in both preprocessing proposals, the images first go through a data normalization step, and after processing (Gaussian blur or wavelet), the Histogram of Oriented Gradients (HOG) method is applied to generate relevant features. The goal is to train classifiers for each of these two processing proposals and verify which one improves the final performance on the test set.

2.3.1 Normalization

Normalization is an important preprocessing step in image processing that scales pixel values to a standardized range. In this project, normalization is achieved by dividing all pixel values by 255.0. This operation transforms the pixel values from the range $[0, 255]$ to $[0, 1]$, making the data more suitable for machine learning algorithms.

The formula for normalization is:

$$I_{\text{norm}} = \frac{I}{255.0} \tag{1}$$

where I represents the original pixel value, and I_{norm} is the normalized pixel value. By scaling the pixel values to a range between 0 and 1, normalization helps in stabilizing the training process, improving convergence, and making the model less sensitive to variations in input intensity.

2.3.2 Gaussian Blur

Gaussian blur is a widely used image processing technique that applies a Gaussian function to an image to smooth it and reduce noise. The Gaussian function distributes the pixel values in a bell-shaped curve, giving more weight to the central pixel and less to those further away. This results in a blurring effect that helps reduce high-frequency noise and detail, making the image less sharp. The degree of blur is controlled by the size of the Gaussian kernel and its standard deviation. In machine learning, Gaussian blur is often used as a preprocessing step to eliminate unnecessary details and enhance the performance of classifiers by focusing on more prominent image features. Equation 2 shows the formula for Gaussian blur function.

$$G(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \tag{2}$$

$G(x, y)$ represents the Gaussian function applied at the point (x, y), where x and y are the pixel coordinates relative to the center of the kernel. The parameter σ is the standard deviation, which controls the amount of blur; a higher σ leads to greater smoothing. The term $\frac{1}{2\pi\sigma^2}$ normalizes the function, ensuring that the total area under the Gaussian curve is equal to 1. The exponential function $\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$ determines how much influence each surrounding pixel has on the central pixel, with closer pixels having a greater impact than those further away. Figure 4 shows an example of Gaussian blur of kernel 7×7 in our original data.

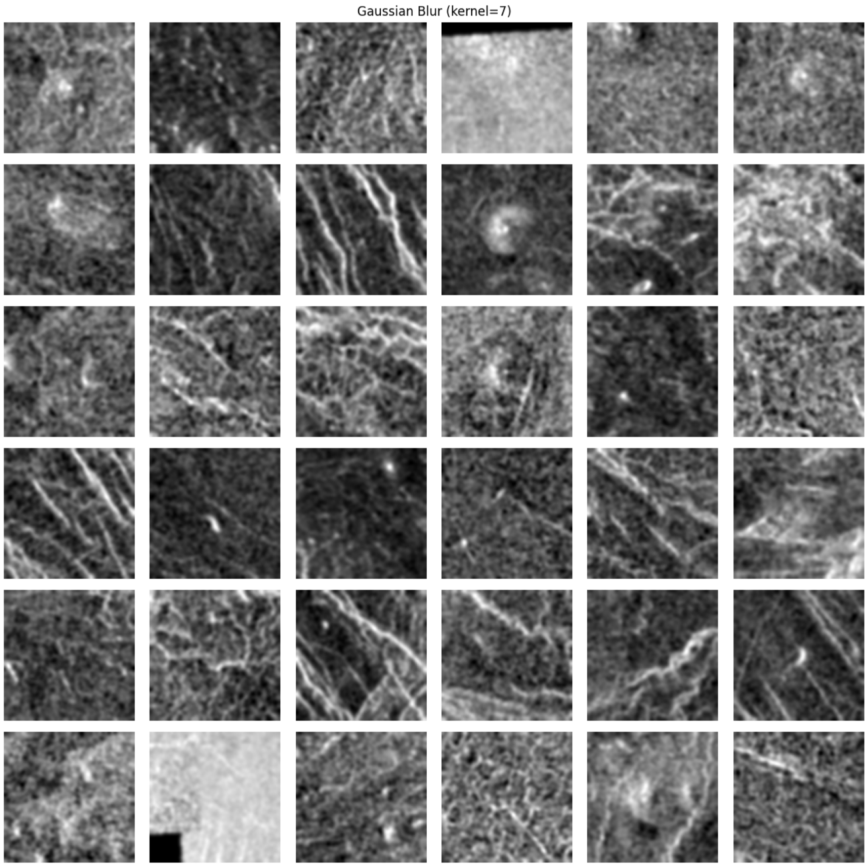

Figure 4: 7×7 Gaussian blur preprocessing.

2.3.3 Wavelet Denoising

Wavelet denoising is a technique used to remove noise from images by leveraging wavelet transforms. The process involves decomposing an image into its wavelet coefficients, modifying these coefficients to reduce noise, and then reconstructing the image. The wavelet transform allows for both spatial and frequency domain analysis, making it effective for noise reduction.

A commonly used wavelet in denoising is the Daubechies wavelet (denoted as ‘db1‘). The choice of wavelet affects how well the denoising process can capture and reduce noise while preserving important features in the image. In addition to selecting the wavelet type, the level of decomposition is also crucial. The ‘level‘ parameter controls how many times the wavelet transform is applied, affecting the granularity of the noise reduction.

The thresholding method applied to the wavelet coefficients determines how aggressively noise is removed. Two common types of thresholding are soft and hard thresholding. In soft thresholding, coefficients below a certain threshold are set to zero, and larger coefficients are shrunk by the threshold amount. Hard thresholding simply sets coefficients below the threshold to zero, leaving larger coefficients unchanged. Figure 5 shows an example of wavelet denoising application on the dataset.

2.3.4 Histogram of Oriented Gradiets

Histogram of Oriented Gradients (HOG) is a feature descriptor used in computer vision and image processing for object detection and recognition. It focuses on capturing the gradient information of an image, which helps in identifying local object shapes and structures.

The HOG descriptor works by dividing the image into small, spatially connected regions called cells. For each cell, the gradient directions and magnitudes are computed. These gradients are then quantized into a histogram, where each bin represents a range of gradient orientations. The histograms from individual cells are then concatenated to form a feature vector that represents the entire image or object. Figure 8 shows an example of images after Gaussian blur + HOG processing,

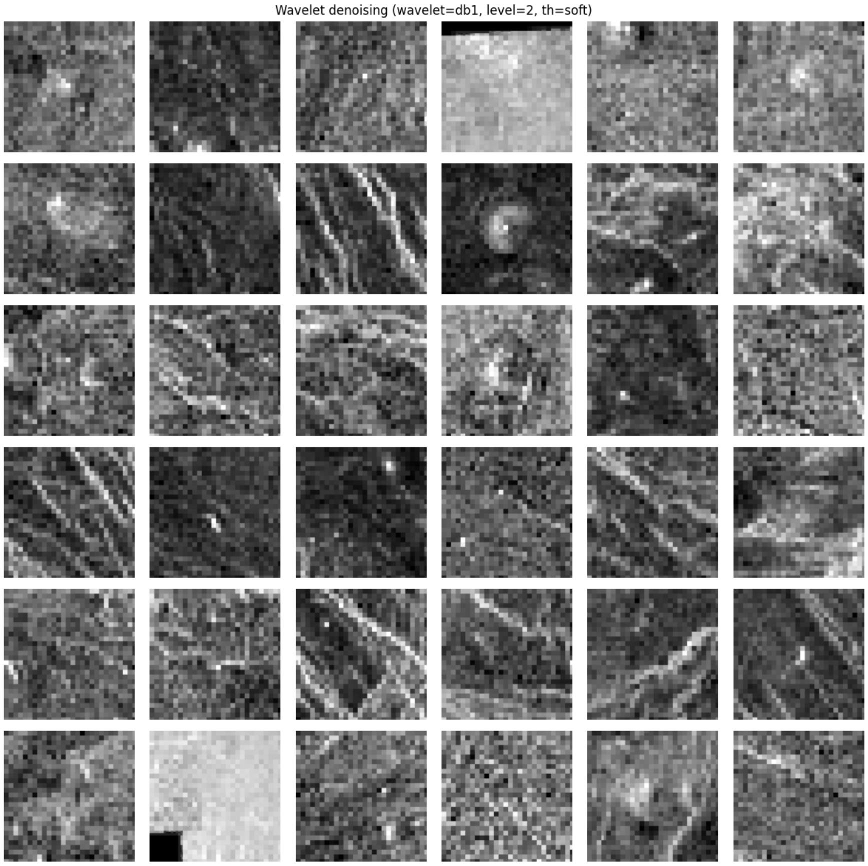

Figure 5: Wavelet denoising preprocessing.

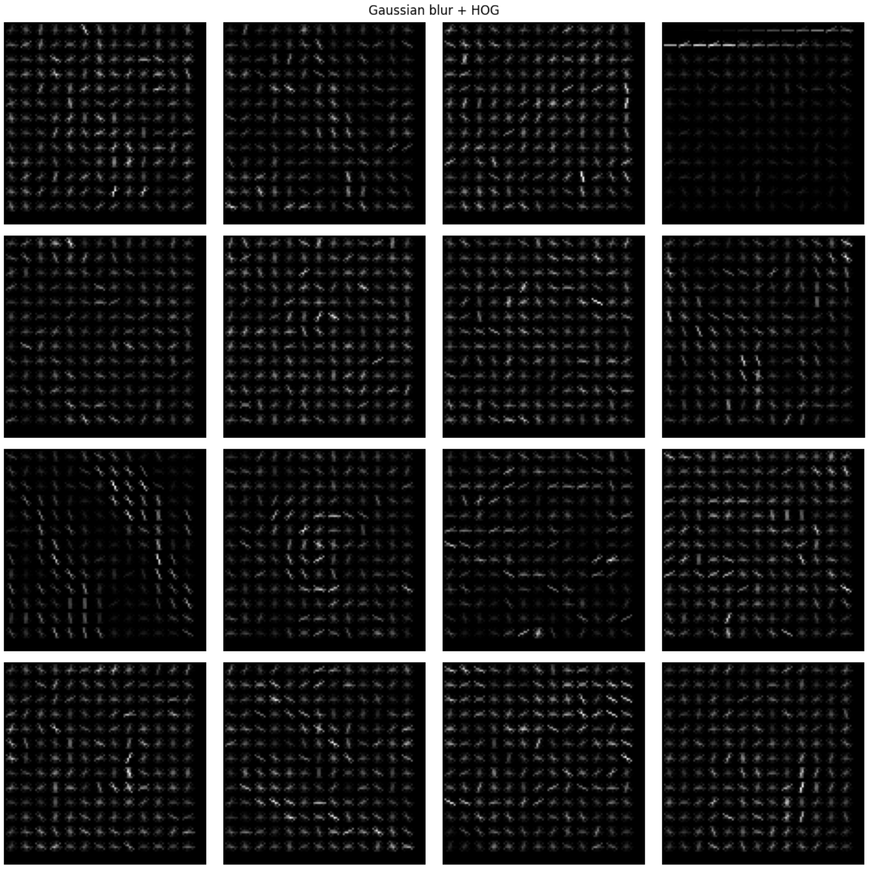

Figure 6: HOG preprocessing.

2.4 Data Augmentation

Since the training dataset is highly imbalanced, as mentioned earlier, just a good preprocessing applied to the images might not be sufficient to achieve good classification performance. To address this issue, intrinsic data augmentation was performed by creating five copies of the training images that contained volcanoes. For each of these five copies, additional processing was applied. The methods used were CLAHE (Contrast Limited Adaptive Histogram Equalization), random horizontal flip, random vertical flip, random diagonal flip (D1), and random anti-diagonal flip (D2). After applying the transformations, the targets for these images were created (as they are all '1'), added to the original dataset, and shuffled to avoid bias. The final results show a balanced batased comprising of 6000 images that contain volcanoes and 6000 images that do not contain volcanoes.

3 Methodolody

The following sections will describe the methodology steps used in this work.

3.1 Workflow Control

As mentioned in the previous sections, the objective of this research is to use classical machine learning algorithms to classify the presence of possible volcanoes in images from a probe on Venus to avoid potential damage in low-visibility or surface approach scenarios. For this purpose, a data augmentation and preprocessing phase was applied, followed by the modeling steps.

A structured workflow was created and made publicly available at `https://github.com/misabellerv/Magellan/tree/main` to optimize all project stages and allow for future modifications. The structure was based on OOP, and a JSON file was created for flow control, storing all project configurations, from data paths to model parameters. This allowed better management of each stage in a single file. Figure 7 shows the workflow diagram.

The data preprocessing phase includes additional parallelization trhough Joblib to speed up the process. Additionally, a pipeline is built that automatically adds each transformation sequentially using scikit-learn Pipelines. Since the data repository already provides training and test data, the data is split only between training and validation to perform cross-validation with 5 folds using GridSearchCV to search for the best hyperparameters

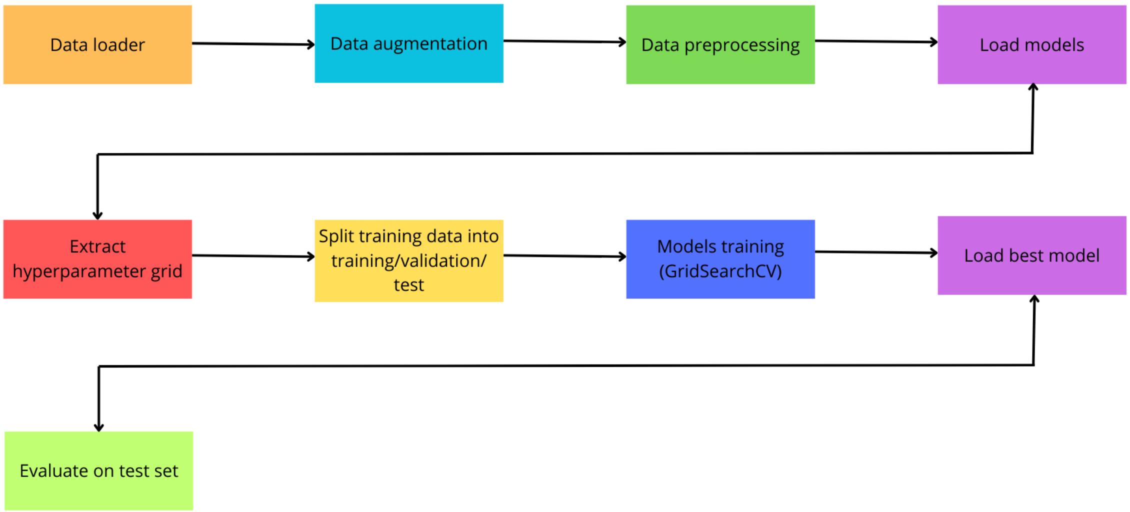
Data loader
Data augmentation
Data preprocessing
Load models
Extract hyperparameter grid
Split training data into training/validation/ test
Models training (GridSearchCV)
Load best model
Evaluate on test set

Figure 7: Workflow scheme.

based on the recall score. This approach was chosen because it was considered more important to correctly classify all cases where volcanoes are present in the image, even if it means mistakenly classifying some images without volcanoes as potentially dangerous.

3.2 Preprocessing Parameters

Although grid search was used to find model hyperparameters during training, the preprocessing parameters were defined visually due to computational and time constraints. For the Gaussian blur, different kernel sizes were tested, and the 7x7 kernel was ultimately chosen. In the case of wavelets, the db1 type, level 2, and soft threshold were selected. Figures 8 and 9 show experiments performed with different values for preprocessing.

3.3 Machine Learning Models

Several different machine learning models were tested to determine which one achieves the highest performance and process optimization. Each of these models will be briefly described below.

- **Support Vector Machine (SVM):** SVM aims to find the optimal hyperplane that best separates the classes in the feature space. It is particularly effective in high-dimensional spaces and for problems where the decision boundary is complex but still requires the data to be linearly separable or close to it through kernel tricks.

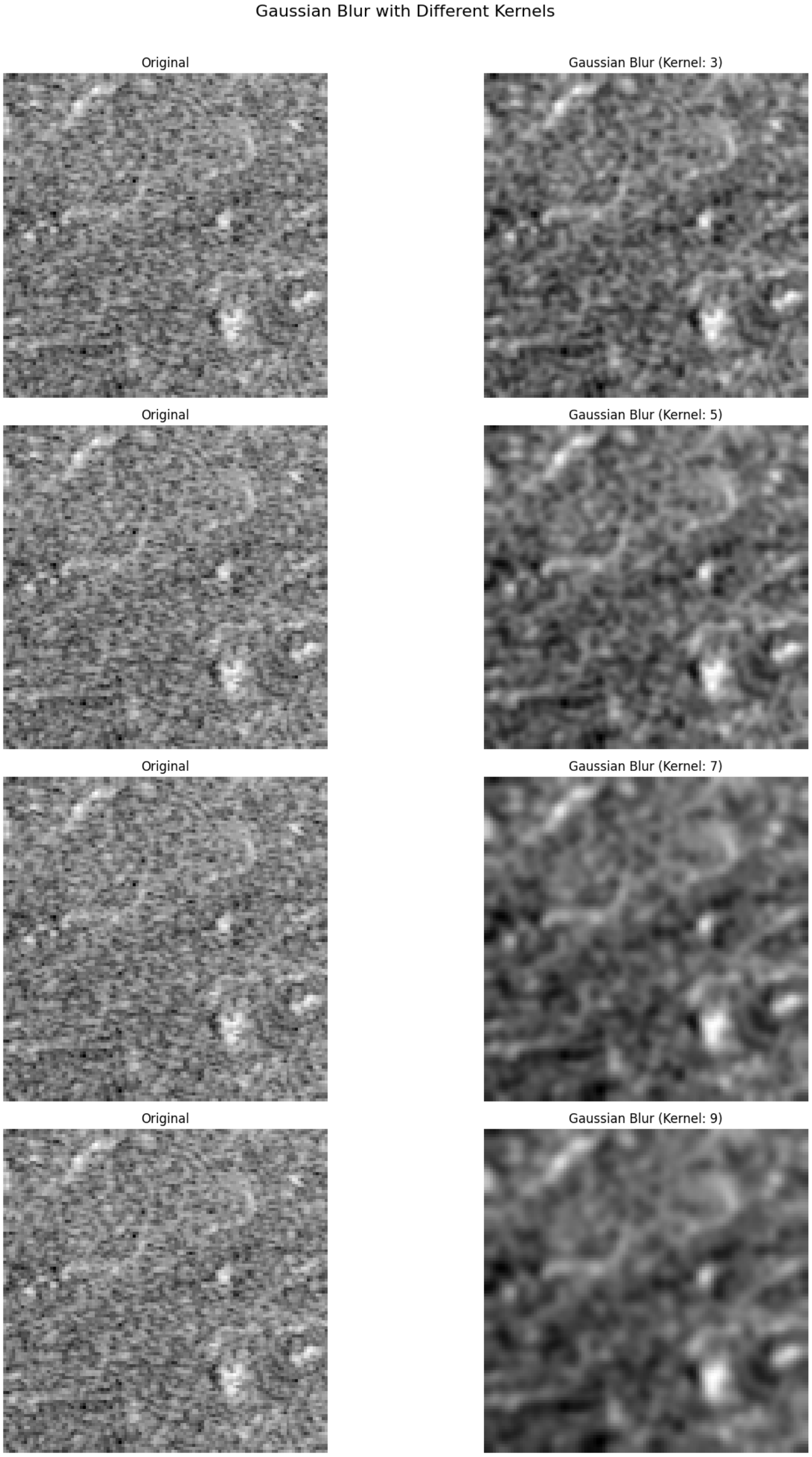
Gaussian Blur with Different Kernels
Original
Gaussian Blur (Kernel: 3)
Original
Gaussian Blur (Kernel: 5)
Original
Gaussian Blur (Kernel: 7)
Original
Gaussian Blur (Kernel: 9)

Figure 8: Gaussian blur preprocessing experiments.

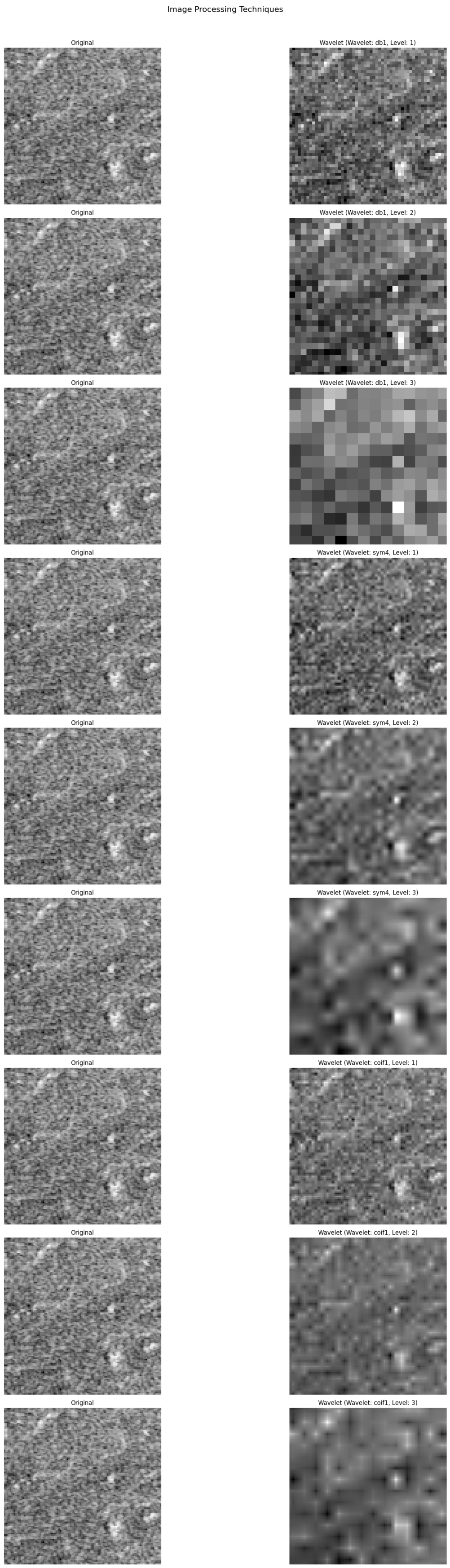
Image Processing Techniques
Original
Wavelet (Wavelet: db1, Level: 1)
Original
Wavelet (Wavelet: db1, Level: 2)
Original
Wavelet (Wavelet: db1, Level: 3)
Original
Wavelet (Wavelet: sym4, Level: 1)
Original
Wavelet (Wavelet: sym4, Level: 2)
Original
Wavelet (Wavelet: sym4, Level: 3)
Original
Wavelet (Wavelet: coif1, Level: 1)
Original
Wavelet (Wavelet: coif1, Level: 2)
Original
Wavelet (Wavelet: coif1, Level: 3)

Figure 9: Wavelet preprocessing experiments.

- **XGBoost:** XGBoost is an implementation of gradient-boosting decision trees. It builds an ensemble of weak learners (trees) iteratively, where each tree corrects the errors of the previous one. XGBoost is known for its scalability and ability to handle large datasets and complex patterns.
- **Decision Trees:** Decision Trees split the data based on feature values to create branches that represent decision rules. Each internal node represents a feature test, and each leaf node represents a class label. Easy to interpret but prone to overfitting if not properly regularized.
- **K-Nearest Neighbors (KNN):** KNN is an instance-based learning algorithm that classifies data points based on the majority class of their nearest neighbors. It can be computationally expensive as it requires storing the entire training set and recalculating distances for each prediction.

3.4 Models Trainined

The strategy chosen for model training was k-fold cross-validation with grid search. This was selected to optimize the search for hyperparameters that could improve the model's performance and also to find the best model on data subsets without specific biases. The number of folds chosen was 5, with an 80% training and 20% validation split. The best model was selected based on the highest recall score on the validation set, and the winning model and its parameters were saved for later evaluation.

It is important to note that two different preprocessing approaches were proposed, and thus, the best models found for each case may differ. In the end, we aim to evaluate which preprocessing method is the best, and which model is the best from the optimal preprocessing method. Tables 2, 3, 4 and 5 show the range of hyperparameters investigated in the grid search.

Table 2: SVM hyperparameters investigated.

C	0.1, 1, 10, 100
Kernel	linear, rbf, poly

Table 3: Decision trees hyperparameters investigated.

Max depth	3, 5, 10, 50
Min samples split	2, 5, 10, 50
Criterion	gini, entropy

Table 4: XGBoost hyperparameters investigated.

N estimators	10, 50, 100, 150
Learning rate	0.001, 0.01, 0.1, 0.5
Eval metric	logloss

Table 5: KNN hyperparameters investigated.

N neighbors	2, 3, 5, 10
Metric	minkowski, euclidian, manhattan

3.5 Results

The results section will be divided by preprocessing type for better organization of the discussions.

3.5.1 Gaussian blur experiments

Tables 6, 7, 8, and 9 show the best hyperparameters found for each model after training with Gaussian blur preprocessing. Table 10 shows the metrics results after evaluating the best models chosen on the test set. It can be observed that there is a tie in recall between XGBoost and SVM. The final choice for performing inferences was XGBoost, as its processing is faster and more optimized compared to SVM. Figure 10 shows the confusion matrix for the best classifier (XGBoost), being possible to observe that the model is predicting most of the figures with volcanoes correctly.

Table 6: SVM hyperparameters tuned.

C	10
Kernel	rbf

3.5.2 Wavelet experiments

Tables 11, 12, 13, and 14 show the best hyperparameters found for each model after training with wavelet preprocessing. Table 15 shows the metrics results after evaluating the best models chosen on the test set. This time, XGBoost has the best overall recall value. Figure 11 shows the confusion matrix for the best classifier (XGBoost), being possible to observe that this model is also predicting most of the figures with volcanoes correctly.

Although both preprocessing methods yielded very good results with average recalls above 90%, it is evident that the final winner is the Gaussian blur preprocessing method (which is also less costly than wavelets) combined with XGBoost for model training and

Table 7: Decision trees hyperparameters tuned.

Max depth	50
Min samples split	10
Criterion	entropy

Table 8: XGBoost hyperparameters tuned.

N estimators	150
Learning rate	0.5
Eval metric	logloss

Table 9: KNN hyperparameters tuned.

N neighbors	2
Metric	manhattan

Table 10: Metrics for the test set (gaussian blur preprocessing).

Model	Accuracy	Precision	Recall	F1
KNN	0.83	0.86	0.83	0.84
Decision trees	0.85	0.89	0.85	0.86
XGBoost	0.95	0.95	0.95	0.95
SVM	0.95	0.96	0.95	0.96

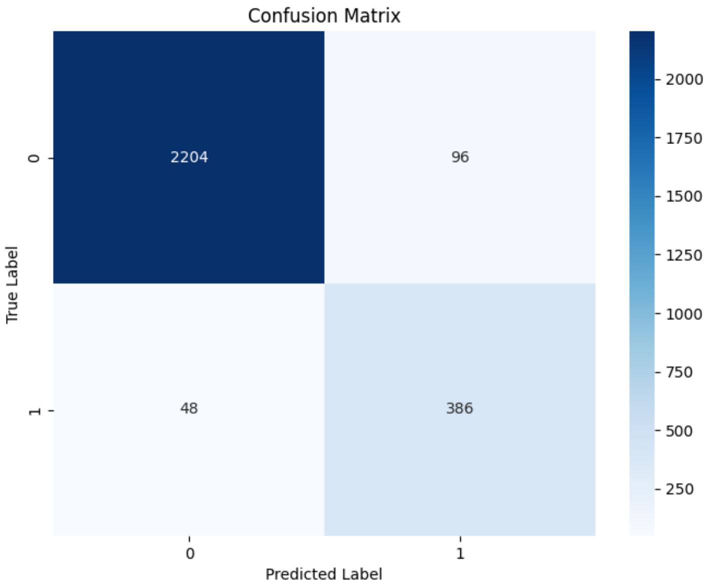

Figure 10: Confusion matrix for XGBoost classifier on the test set.

evaluation, using the hyperparameters found in Table ??. These results demonstrate that it is possible to achieve excellent results, and even better than those obtained with convolutional neural networks (which are costly algorithms), using much simpler methods and without the need for GPU.

Table 11: SVM hyperparameters tuned.

C	100
Kernel	rbf

Table 12: Decision trees hyperparameters tuned.

Max depth	50
Min samples split	5
Criterion	entropy

Table 13: XGBoost hyperparameters tuned.

N estimators	150
Learning rate	0.5
Eval metric	logloss

4 Conclusions

In this work, we explore some of the data obtained from the Magellan spacecraft mission, which represented a significant advancement in the exploration of Venus by providing a detailed mapping of the planet's surface. In this context, we discuss the growing importance of artificial intelligence (AI) and computer vision in enhancing modern space missions, especially in identifying suitable landing areas and analyzing extraterrestrial terrains.

The methodology proposed in this study involved a robust workflow for the analysis and processing of space data, available in a Kaggle dataset. Using tools such as Python, we conducted data analysis, preprocessing, data augmentation, and investigated various machine learning models, including SVM, XGBoost, Decision Trees, and KNN. These models were trained and validated using cross-validation techniques, and their performances were evaluated based on the recall metric.

The results indicated that the XGBoost model, in conjunction with preprocessing using Gaussian blur, achieved the best performance, with 95% recall on the test set, standing

Table 14: KNN hyperparameters tuned.

N neighbors	2
Metric	manhattan

Table 15: Metrics for the test set (wavelet preprocessing).

Model	Accuracy	Precision	Recall	F1
KNN	0.82	0.87	0.82	0.83
Decision trees	0.82	0.87	0.82	0.83
XGBoost	0.93	0.93	0.93	0.93
SVM	0.92	0.92	0.92	0.92

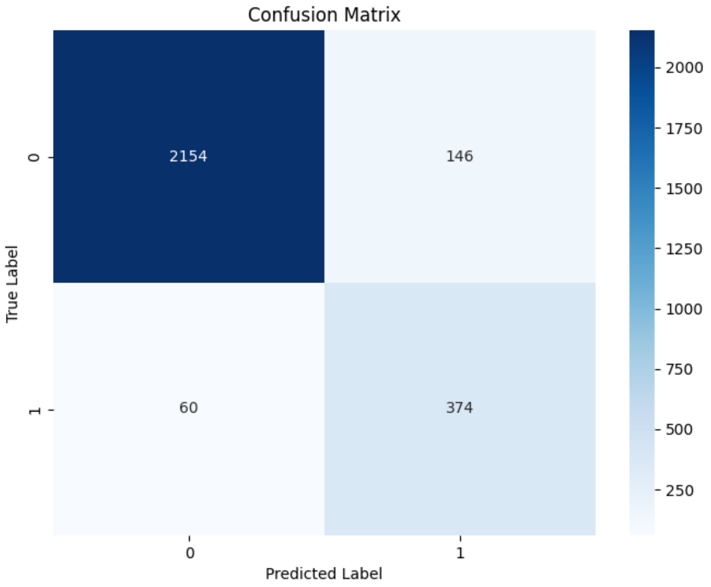

Figure 11: Confusion matrix for XGBoost classifier on the test set.

out for its high sensitivity and robustness in inference. XGBoost has higher processing speed compared to SVM, making it a better option for classifying potential volcanic regions.

As next steps, we suggest exploring new hyperparameters and introducing additional models, such as LDM, Naive Bayes, and Convolutional Neural Networks, to deepen the investigation. Expanding to these methods may offer new insights and further enhance the accuracy and applicability of the developed models. The ongoing application of AI and computer vision promises to transform space exploration, enabling increasingly autonomous and secure missions capable of uncovering the secrets of the cosmos with greater efficiency.