AML 2025 Group 42 - Challenge 1: Aerial Imagery - Cactus Classification Challenge

AHAMAN ULLAH Efaz
EURECOM
Efaz.Ahaman-Ullah@eurecom.fr

CHAUVIER Tom
EURECOM
Tom.Chauvier@eurecom.fr

MEKKI Romain
EURECOM
Romain.Mekki@eurecom.fr

SARR Serigne
EURECOM
Serigne.Sarr@eurecom.fr

Abstract

In this report, we address the Cactus Classification Challenge using 3 different solutions. The dataset presents an imbalanced binary classification task, for which we adopted the F1-score as the main evaluation metric. Having identified the linear shape of columnar cactus, we started by implementing a simple line detection algorithm using Canny edge detector and hough transform. After that, we trained a simple CNN that achieved a test F1-score of 0,993. Finally, we took a more Bayesian approach by applying logistic regression with Hamiltonian Monte Carlo on the latent features produced by a VAE, in hope that we could both classify images and generate new ones with similar distribution to correct the class imbalance.

1 Data Preparation

1.1 Dataset analysis

The training dataset consists of 17,500 labeled images, each either containing a cactus (`has_cactus = 1`) or not (`has_cactus = 0`). An initial class distribution check revealed a significant imbalance:

- `has_cactus = 1`: 75.06%
- `has_cactus = 0`: 24.94%

Given this imbalance, accuracy alone would be misleading. Thus, we opted to use the **F1-score**, a harmonic mean of precision and recall, which is more appropriate for imbalanced binary classification problems.

We observed that cacti often appear as thin, straight vertical lines, making them distinguishable from the background. In contrast, images without cacti typically depict ground textures, usually brown or gray, with no prominent or structured shapes.

Figure 1: Comparison between image with and without cacti

1.2 Dataset construction

Splitting

We performed two different stratified splits of the data to preserve label distribution. The first one is a classic split we used for our first approaches :

- 70% for training
- 15% for validation
- 15% for testing

The second one was destined to be used to train and test the VAE and the Bayesian logistic regressor with HMC.

- 80% for training (including calibration)
- 20% for testing

Features

All images in the dataset are 32x32 pixels and in RGB format. Appropriate transformations were made following the model we used. For instance, when we first decided to train the Bayesian classifier we chose to train directly on flattened images, thus having vectors with a fairly large dimension (1024). HMC struggled to explore the posterior distribution, giving very little samples, thus encouraging us to use 20 latent features of the images instead (given by a VAE) for the Bayesian approach. Note that those considerations don't apply to CNNs and the classic computer vision approach.

2 Models and Performances

2.1 Solution 1: line detection

We observed that, in the dataset, cacti often appear a straight thin lines. This motivated us to first try an old school computer vision solution: line detection using Canny edge detection and Hough line transforms.

Image preprocessing

We performed the following preprocessing on each image:

- Convert the image to grayscale.
- Apply Canny edge detection with variable thresholds.
- Apply Hough Line Transform to detect linear features.

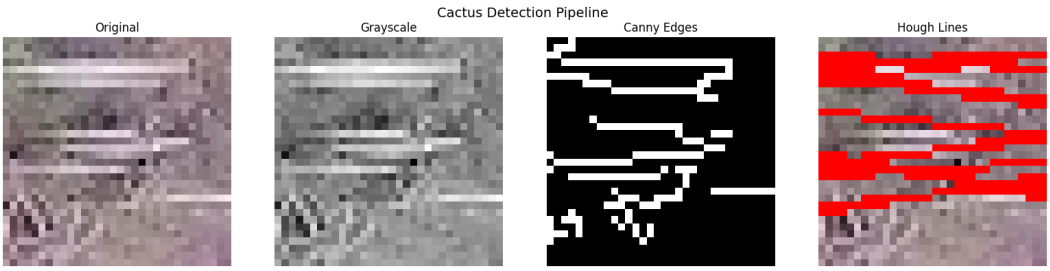

Figure 2: Cactus detection pipeline

Classification Rule

After the preprocessing, this solution is very straightforward: we classified each image with at least one line as a cactus and images with zero lines as no cactus.

Hyperparameters choice

We needed to set three hyperparameters:

- the Canny low threshold
- the Canny high threshold
- Hough line threshold

Those values need to be balanced so that cactus are detected, but smaller background shapes are not. To evaluate this, we balanced the dataset so that our model is not favored to choose cactus.

We conducted a grid search to set them, and found the best result with (150, 400, 15), with a balanced f1-score of 0,76.

Result

On the test set, the model achieved the following result when using the balanced set:

F1-score: 0.7691
Precision: 0.7564
Recall: 0.7824

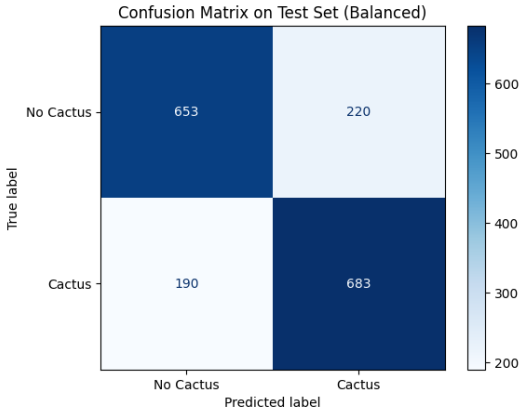

Figure 3: Confusion Matrix on Test Set

We obtained interesting results considering the simplicity of this solution. It does not require any training phase and is very computationally efficient for images of this size.

Unfortunately, we rapidly saw the limits of this solution: Many images without cactus contain edges in their background or other structures. To exclude those, we need to increase the thresholds, and exclude many images containing small cactus, increasing false negatives.

We need to explore more complex solution to have a truly accurate classifier.

2.2 Solution 2: CNN

CNNs are well-suited for image classification because they automatically learn spatial features like edges, textures, and shapes. For this challenge, where cactus detection relies on visual patterns in small RGB images, CNNs can effectively capture relevant cues without manual feature engineering. Their ability to share parameters and preserve spatial hierarchies makes them both efficient and accurate. Given the simplicity of the input data and the relatively modest dataset size, we opted for a lightweight CNN architecture to balance performance and training efficiency, avoiding overfitting while maintaining high classification accuracy.

Architecture

We implemented a lightweight CNN:

- **Conv2D (3, 32)** → ReLU → MaxPool
- **Conv2D (32, 64)** → ReLU → MaxPool
- Fully connected layers: 64*6*6 → 64 → 1

A `Sigmoid` activation was used to output probabilities between 0 and 1.

Loss and Optimizer

Binary Cross-Entropy Loss was used, optimized with Adam (`lr=0.001`). The model was trained for 10 epochs.

Results

Training and validation showed good convergence. Validation F1-scores improved significantly over epochs, reaching a peak of **0.9916**. On the test set, the model achieved strong performance with the following metrics:

F1-score: 0.9916
Precision: 0.9944
Recall: 0.9888

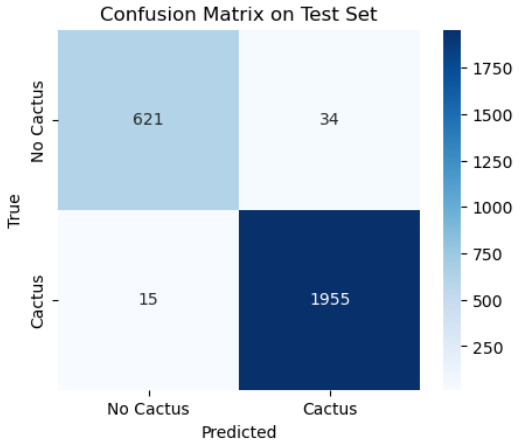

Figure 4: Confusion Matrix on Test Set

Metrics Justification

The dataset is imbalanced. Accuracy could be misleading if the model always predicted the dominant class. We therefore used the **F1-score**, which better reflects performance on both classes, especially the minority class.

2.3 Bayesian approach

Motivation

Despite the very good performances of our CNN model, models based on frequentist approaches lack clear expression of uncertainty in their output. For the final purpose of the task, the goal was to be able to recognize vegetation over protected areas, but we would like to know to what extend our prediction is uncertain. Moreover, the promise of generating new images to perhaps better balance our dataset to build more robust classic classifiers was attractive.

Those considerations led us to consider making a probabilistic classifier based on Bayesian logistic regression, using Hamiltonian Monte Carlo for its reliability despite the relatively slow convergence.

Architecture

We trained a Variational autoencoder (VAE) on the second training set with 20 latent dimensions for the future hamiltonian sampling to be fast enough :

Encoder (input: $1 \times 32 \times 32$) :

- **Conv2D (1, 16, k=3, s=2, p=1)** $\rightarrow$ ReLU $\rightarrow$ Output: $16 \times 16 \times 16$
- **Conv2D (16, 32, k=3, s=2, p=1)** $\rightarrow$ ReLU $\rightarrow$ Output: $32 \times 8 \times 8$
- **Conv2D (32, 64, k=3, s=2, p=1)** $\rightarrow$ ReLU $\rightarrow$ Output: $64 \times 4 \times 4 \rightarrow$ Flatten
- Two linear layers:
  - **fc_mu** : 1024 $\rightarrow$ latent_dim
  - **fc_logvar** : 1024 $\rightarrow$ latent_dim

Decoder (reconstructing image) :

- Linear layer: latent_dim $\rightarrow$ 1024 then unflatten to $64 \times 4 \times 4$

- **ConvTranspose2D (64, 32, k=3, s=2, p=1, op=1)** → ReLU
- **ConvTranspose2D (32, 16, k=3, s=2, p=1, op=1)** → ReLU
- **ConvTranspose2D (16, 1, k=3, s=2, p=1, op=1)** → Sigmoid

The VAE served to map the features of the training images to a latent representation in 20 dimensions. It was trained for 20 epochs. The resulting data was used to train a logistic regressor based on HMC.

Hamiltonian Monte Carlo

We simulated tree traces with an NUTS sampler with $target\_accept = 0.9$ using 500 burnt samples per chain and sampling approximately 2000 samples per chain, as well as giving normal distributions to all weights/bias. The R Hat metric was overall approximately 1 for all weights signifying close convergence.

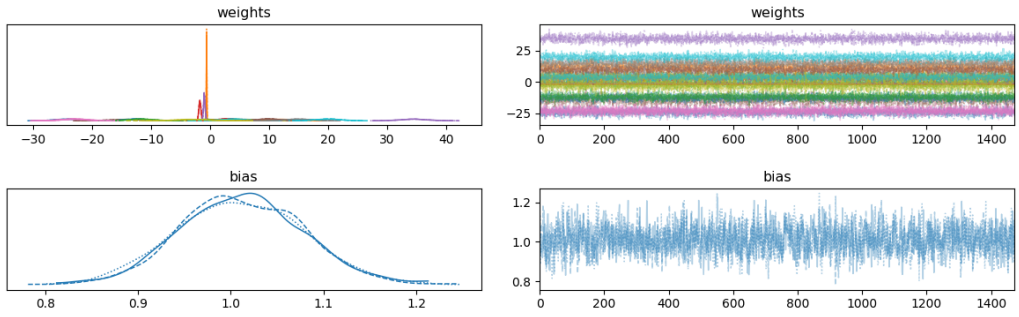

Figure 5: Trace plot of the weights and bias during HMC

Logistic regression

With the posterior distribution we could then classify an image by applying dot product to the weights and latent image, scale the logits (by a temperature) then applying the sigmoid function. The advantage of this method is that it yields a probability score that allows to tell at which confidence an image is being classified. To figure out what was the threshold to chose for separating the probability of getting each class we plotted the various probabilities obtained with images of a given class :

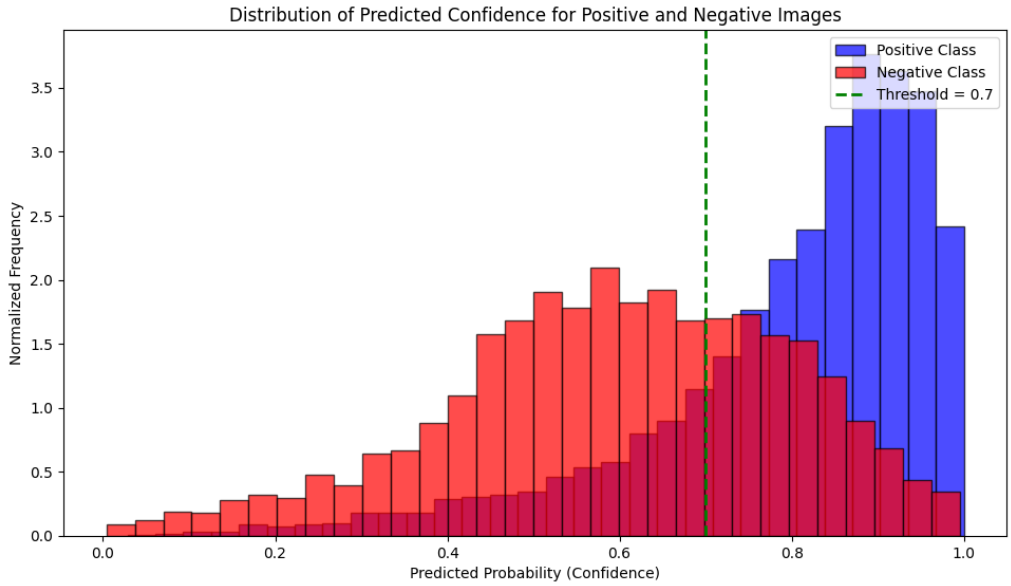

Figure 6: Confidence distribution for each class

With the rather significant overlap between the two distributions - which can be explained by the fact that latent features for cactus images tend to resemble those of non-cactus images with a latent dimension of 20 - we chose a threshold that is roughly composed of the mean of the mean confidences for each distribution. That way, a latent image would be classified as a cactus image if its probability exceeds the threshold, and not otherwise. We could have increased the latent dimension, but it would have resulted to very slow convergence of the Hamiltonian Monte Carlo in our setup due to the posterior difficulty.

F1-score: 0.8388
Precision: 0.8836
Recall: 0.7982

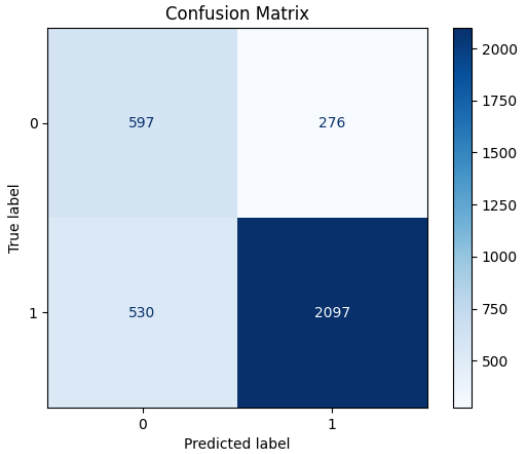

Figure 7: Confusion matrix on test set for logistic HMC approach

Additionnaly, calibration testing on the training set gave an optimal temperature - one that minimizes the ECE error - of 0.9.

Bonus : Non-optimal Hamiltonian Cacti

The bonus of using a VAE was being able to sample new images from the learned latent distribution. We did so by rejection sampling using the previously learned latent space. Though rejection sampling is not the very best way of doing so - as one may want to optimize each feature to maximize the acceptance probability - we obtained a couple of cactus images that achieved "good" probability (> 0.7) of being classified as cactus images. Disclaimer : they don't look like cacti.

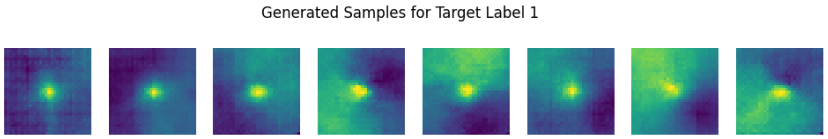

Figure 8: Hamiltonian cacti

3 Conclusion

This challenge demonstrated the effectiveness of simple solutions, namely using computer vision algorithms, as well as the robust ability of CNNs to capture appropriate features of an image. A Bayesian approach is always welcome but rather incomplete in our case, as the VAE seemed to not have captured enough the features of cacti images, explaining the overlap in the two classes despite the satisfying convergence for HMC. Finally, although the cactus images generated via Hamiltonian Monte Carlo were visually intriguing, their fidelity was insufficient for reliable inclusion in our dataset. Perhaps experimenting with GANs could allow for better fidelity.