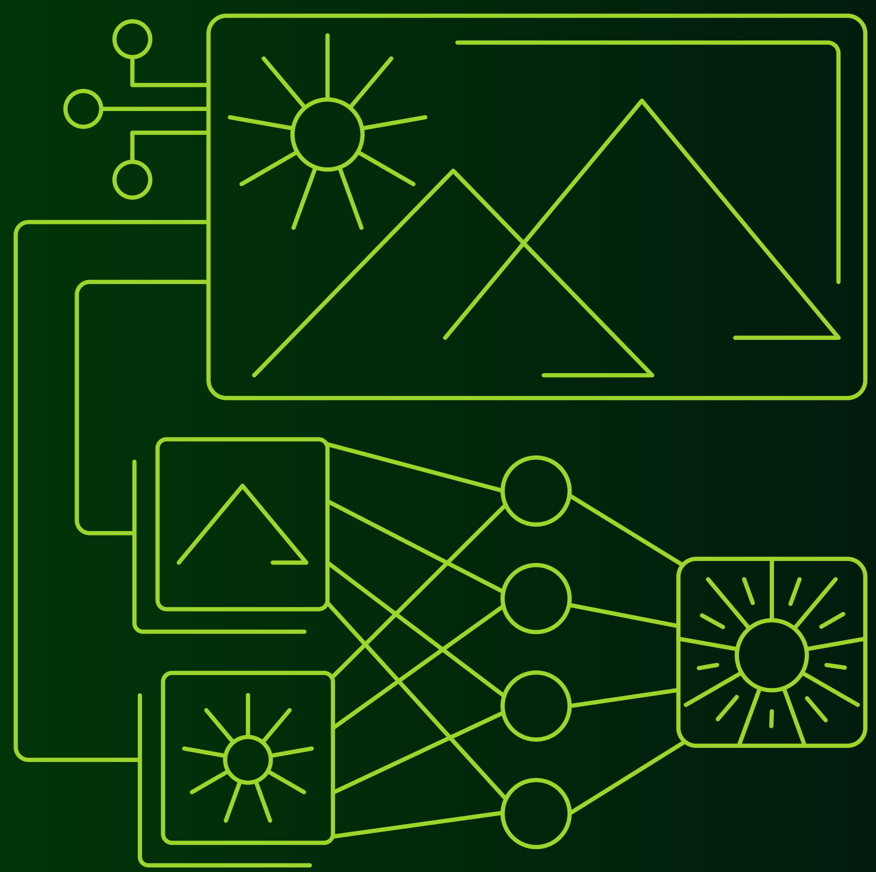
MACHINE LEARNING

IMAGE CLASSIFICATION WITH CONVULOTIONAL NEURAL NETWORKS



Renato Vivar Orellana

Data Science Engineer

SLIDE 1: PROBLEM & APPROACH OVERVIEW

PERFORMANCE ON TEST SET

- Binary image classification task using 48×48 grayscale images of Mars.
- Goal: Automatically classify images as with craters (1) or without craters (0).
- Evaluation metric: F1 Score (accounts for imbalance in the dataset).

CHALLENGES & STRATEGY

- Imbalanced data: High risk of bias toward majority class
- Limited labeled data, especially for minority class
- Semi-supervised learning strategy adopted:
- Train an initial CNN model on labeled data
- Use its predictions on extra unlabeled data
- Select high-confidence predictions (low threshold for class 0)
- Add pseudo-labeled samples to expand the dataset
- Retrain a final CNN model with combined data

DATASET DETAILS

- Labeled dataset: ~3,000 images
- Imbalance: ~64% class 0, 36% class 1
- Unlabeled extra dataset provided for potential improvement
- Test set without labels provided for submission

Mars images with (top row) and without (bottom row) craters.

MODELING HIGHLIGHTS

- Use of Convolutional Neural Networks (CNNs) for image representation
- Architecture tuning, dropout regularization, Adam optimizer
- Performance tracked using:
- Validation loss
- F1 Score, Precision, Recall, Confusion Matrix, PR Curve

Renato Vivar Orellana

Data Science Engineer

CNN ARCHITECTURE & ROLE IN CLASSIFICATION

CNN ARCHITECTURE OVERVIEW

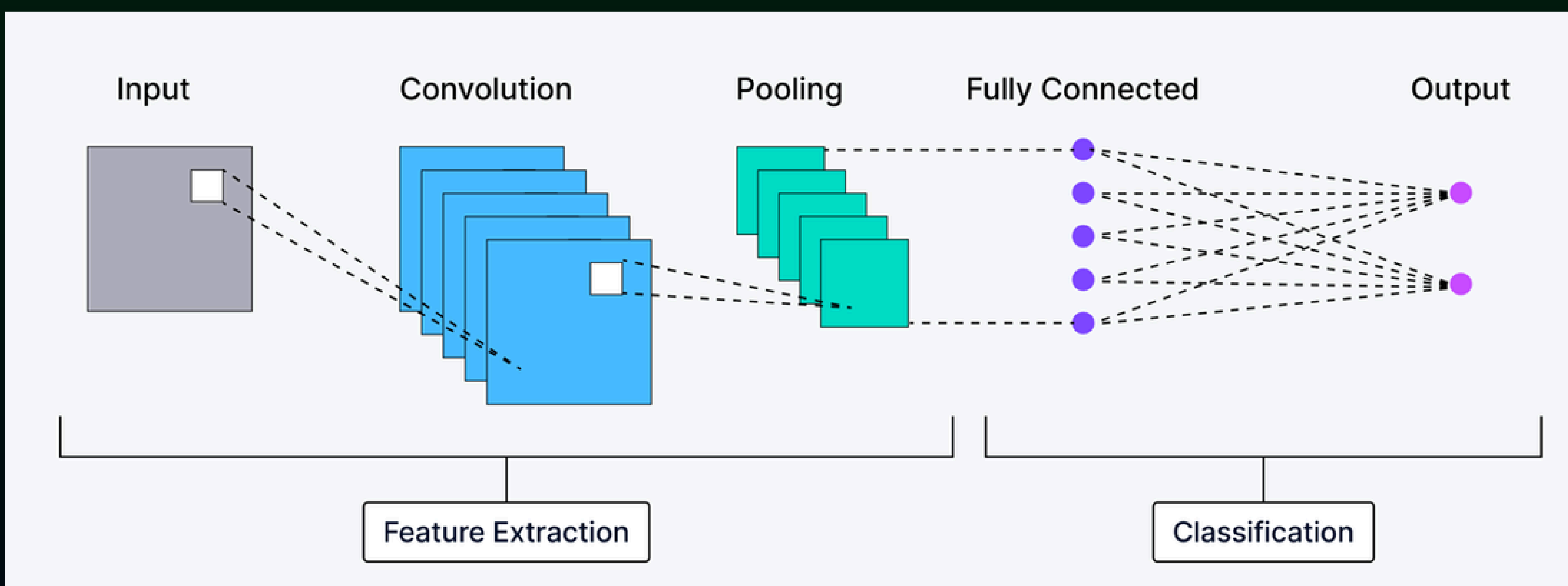
Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 46, 46, 16)	160
max_pooling2d (MaxPooling2D)	(None, 15, 15, 16)	0
conv2d_1 (Conv2D)	(None, 13, 13, 32)	4,640
max_pooling2d_1 (MaxPooling2D)	(None, 6, 6, 32)	0
flatten (Flatten)	(None, 1152)	0
dense (Dense)	(None, 8)	9,224
dropout (Dropout)	(None, 8)	0
dense_1 (Dense)	(None, 1)	9

Model Code Architecture - CNN

- Input: 48×48 grayscale image
- Convolution + Pooling: Detect crater-like features
- Flatten + Dense: Combine features into decision
- Sigmoid output: Predicts crater presence (1) or absence (0)
- Threshold: Applied to output probability (e.g., 0.5 or 0.25)
- Used again in semi-supervised loop to expand dataset

WHY CNNS FOR THIS TASK?

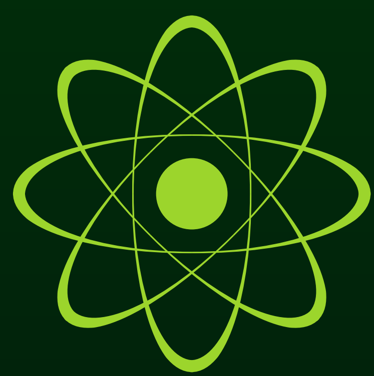
- Automatically learn features (no manual pixel engineering)
- Capture spatial patterns (edges, textures, crater shapes)
- Efficient for image data with shared weights and local connectivity
- Well-suited for grayscale 2D input like Mars crater images



CNN PARAMETERS & EVALUATION CONCEPTS

KEY PARAMETERS

- **Epoch:** One full pass through the training data.
  → We tested 50–75 epochs to balance learning and overfitting.
- **Learning Rate** (LR): Controls how fast weights are updated.
  → Tuned between 0.0003–0.0004 for stable convergence.
- **ReLU Activation**: ReLU(x) = max(0, x)
  → Speeds up training by avoiding vanishing gradients.
- **Threshold** (e.g., 0.25 / 0.5):
  → Applied to model output to decide final class (0 or 1).
  → Lower thresholds help recover more class 0 in imbalance.



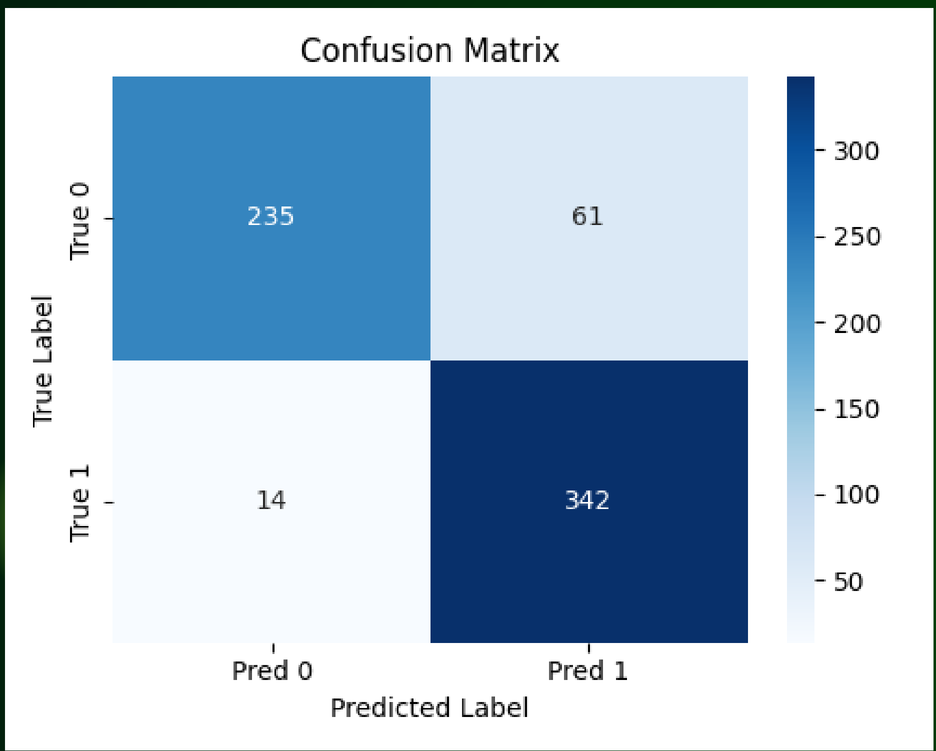
EVALUATION METRICS

- F1 Score: Harmonic mean of Precision and Recall.
  → Good for imbalanced data.
  → Formula:
- Validation Loss: Measures how well the model generalizes to unseen data.
  → Used to select the best model and detect overfitting.

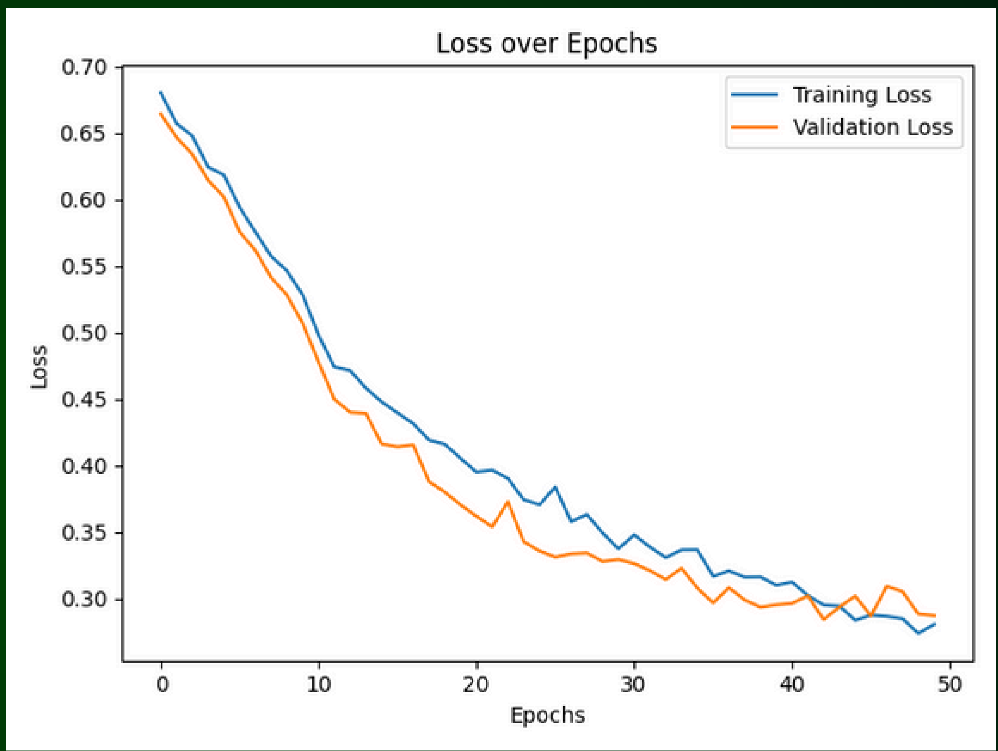
Renato Vivar Orellana
Data Science Engineer

RESULTS – MODEL EVALUATION

The final CNN model demonstrates strong generalization and reliable classification performance. As shown in the learning curve, both training and validation loss steadily decrease and converge around epoch 45, indicating stable learning without overfitting. The precision-recall curve maintains high precision (>0.9) across most recall values, reflecting excellent class discrimination even under class imbalance. The confusion matrix confirms balanced predictive power: 260 true negatives, 335 true positives, and relatively few misclassifications (43 false positives and 21 false negatives). This leads to a high F1 score and validates the use of threshold tuning and semi-supervised augmentation to boost crater detection performance.

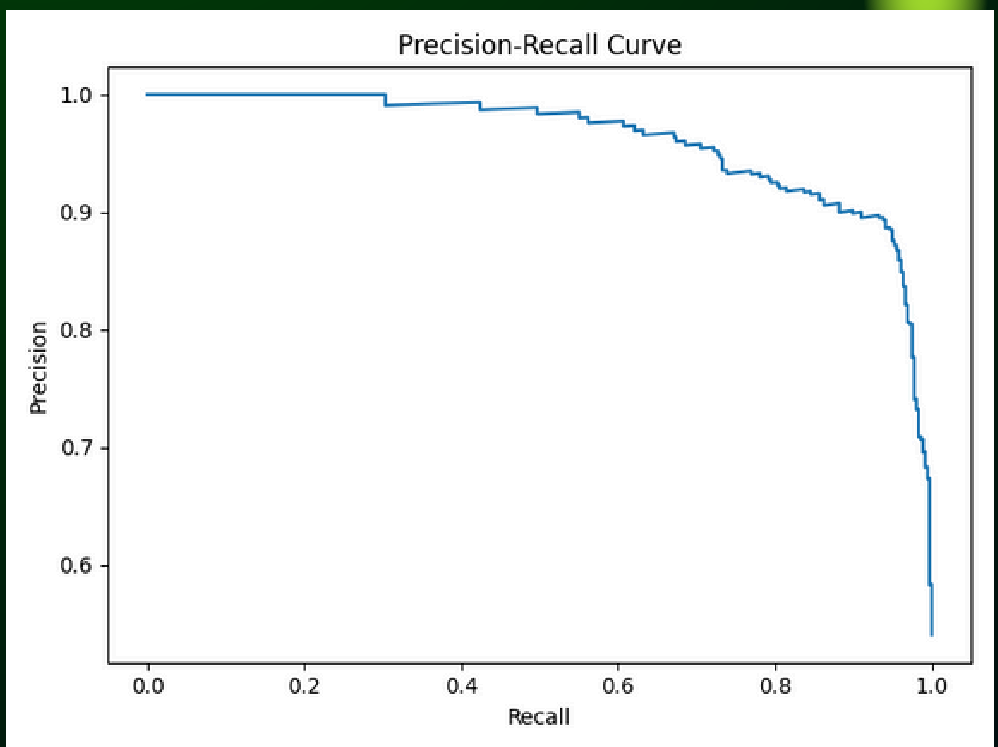


Confusion Matrix



Learning Curve

FINAL METRICS	
VAL LOSS	0.2989
VAL ACCURACY	0.8953
F1 SCORE	0.9016
RECALL	0.941



Precision-Recall Curve

Renato Vivar Orellana
Data Science Engineer

KEY TAKEAWAYS & APPLICATIONS

- Threshold Tuning Improves Recall on Minority Class
- Adjusting the output threshold (e.g., lowering to 0.25) allowed us to recover more crater (class 0) samples from the unlabeled data, helping mitigate the dataset imbalance and improve model robustness.
- Careful Architecture & Hyperparameter Tuning Is Key
- We achieved the best results using a small CNN with two Conv2D + Pooling layers, a dropout layer to prevent overfitting, and tuning the learning rate and number of epochs. These adjustments improved both validation loss and generalization.
- Not All Standard Techniques Work in Every Scenario
- Although techniques like data augmentation, input reduction, and batch normalization often help with imbalance, in this case they led to overfitting or loss of meaningful crater features due to the small size (48x48) and grayscale nature of the images.
- Semi-Supervised Learning Helped Leverage Unlabeled Data
- Using the initial model to pseudo-label high-confidence examples from the extra dataset helped expand the training set without additional human labeling effort, improving overall F1-score and model stability

Renato Vivar Orellana
Data Science Engineer