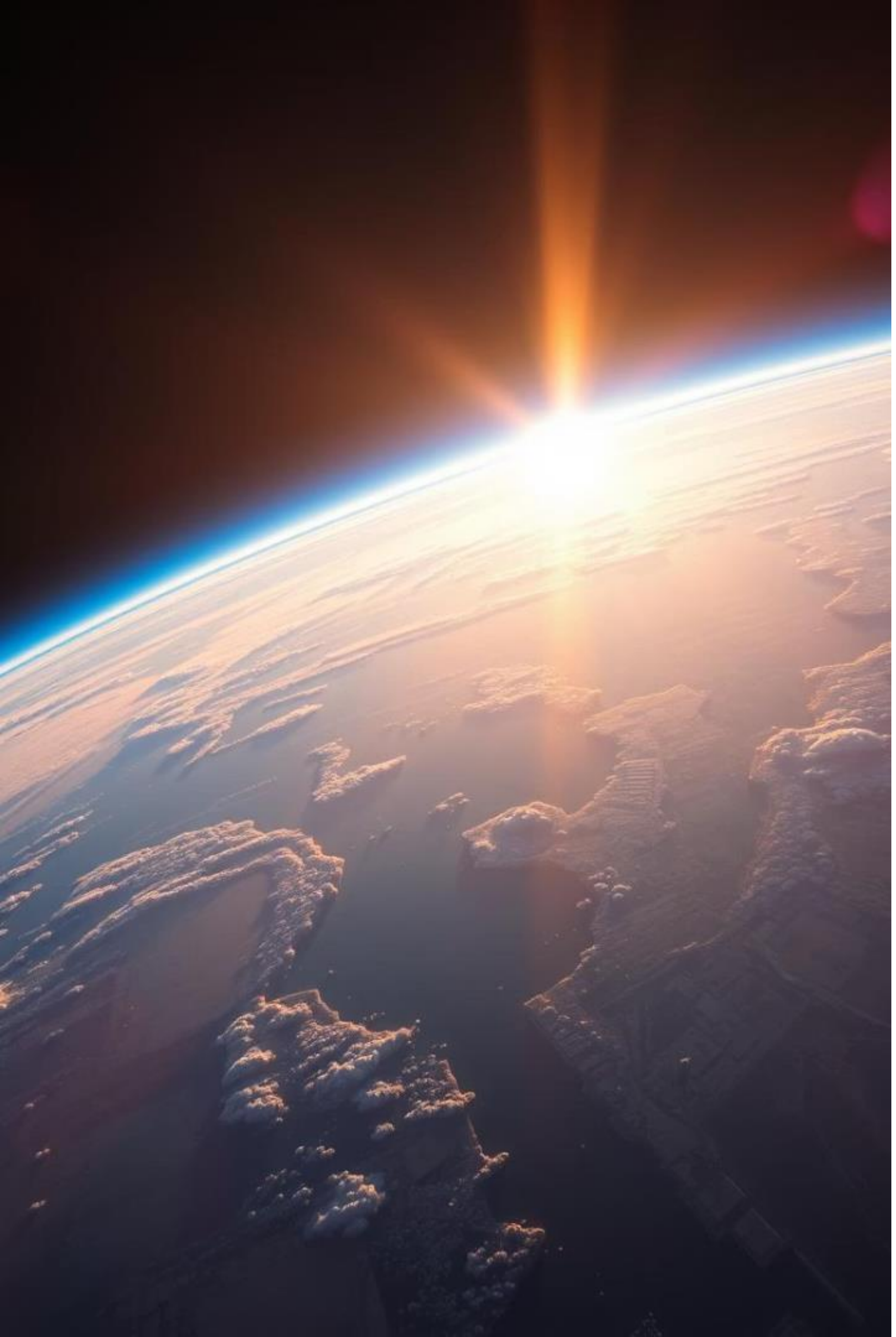


# Satellite Image Classifier & Horizon Detection

This presentation introduces our final project for the Introduction to Space Engineering course at Ariel University. Our team developed an autonomous onboard software system designed for nanosatellites (CubeSats) to classify satellite images and detect the horizon. The project aims to optimize image transmission by filtering and compressing imagery based on quality, ensuring only valuable data reaches ground stations. This innovative approach addresses resource constraints unique to nanosatellites, enhancing mission efficiency and data utility.

# Project Overview

## Challenge

CubeSats face severe limitations in processing power, memory, and communication bandwidth. These constraints require efficient onboard image selection and transmission strategies.

## Objective & Approach

We implemented a Python-based rule system to automatically classify images, images, detect black, blurry, noisy, or sunburned frames, identify images containing Earth and horizon, and compress compress selected images to conserve conserve bandwidth.

## Impact

The system maximizes scientific data sent sent to Earth, reduces unnecessary transmission, and supports realistic CubeSat mission conditions, improving improving operational efficiency.

# Motivation

## Context

Nanosatellites operate with strict limits on size, power, and and data transmission, facing facing dynamic environmental environmental challenges like like motion and lighting changes.

## Problem

A large majority of images captured are low quality or irrelevant, consuming limited limited bandwidth and requiring laborious ground-ground-based filtering.

## Goal

Enable autonomous onboard filtering and compression to prioritize prioritize meaningful images containing Earth and horizon, streamlining streamlining data handling and increasing mission value.

# System Architecture

1

## Phase 1: Image Classification

Sequential rule-based checks detect image defects and Earth/horizon presence to classify images as good or bad automatically.

2

## Phase 2: Image Compression

Good images are compressed adaptively using JPEG quality quality adjustments based on file size for efficient transmission. transmission.

# Image Quality Classification

## Key Quality Checks

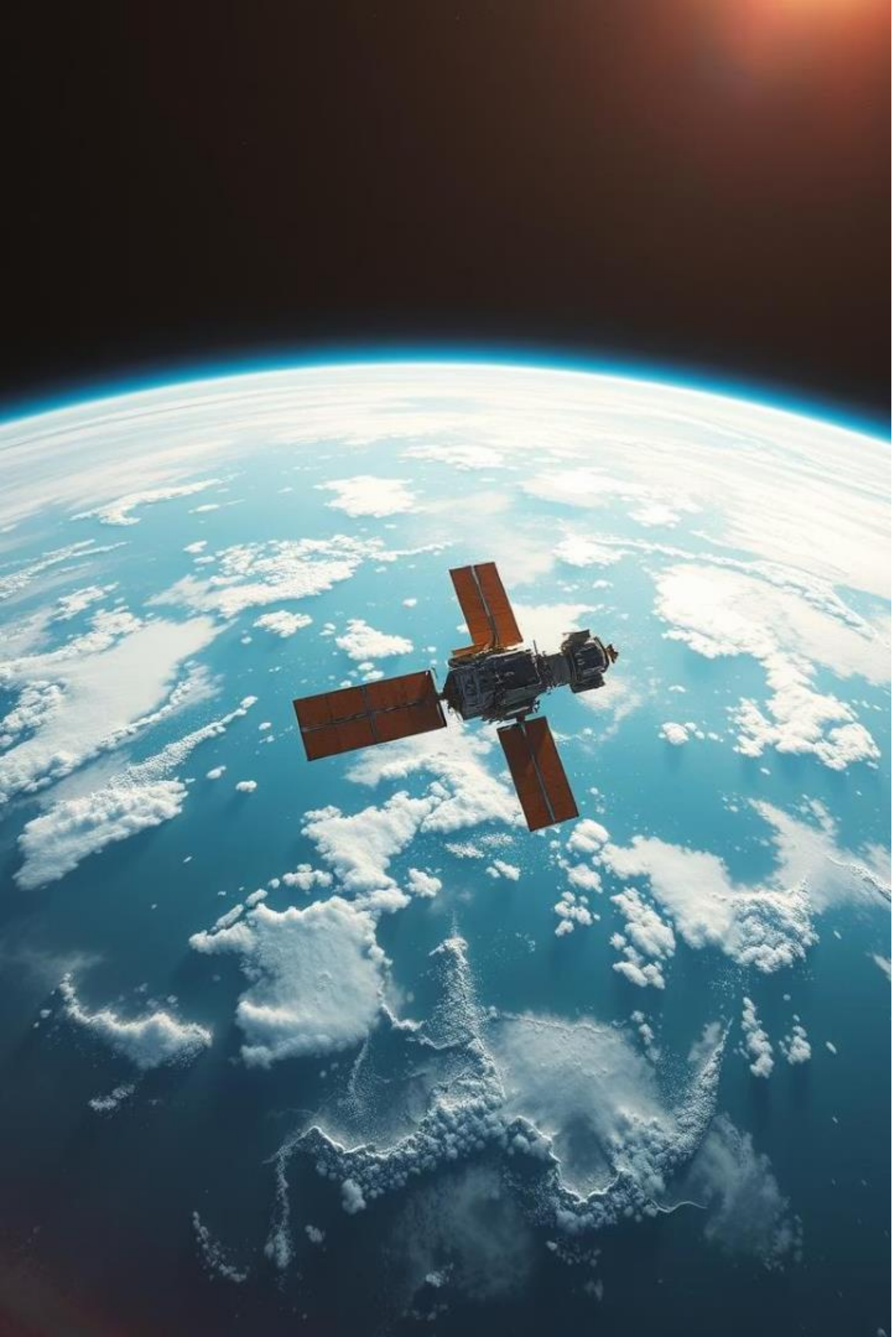
- Black Image Detection (≥98% dark pixels)
- Blur Detection via Laplacian variance
- Noise Detection comparing original & denoised images
- Sunburn Detection (≥60% bright pixels)

## Scene Validation

- Earth Detection (≥3% pixels bright enough)
- Space Detection for dark empty frames
- Horizon Detection using Canny edge + circle fitting

# Horizon Detection

- Detects Earth's curved horizon using edge and contour analysis.
- Applies Canny edge detection and fits a circle to contours using least squares.
- Validates the arc based on shape, size, and image position.
- Only accepts arcs that resemble the horizon (not full Earth or noise).
- If detected, the image is tagged as HZ_Horizon; otherwise, NO_HORIZON.



# Image Compression Module

## Compression Logic

JPEG compression quality is dynamically set based on image size, balancing file size reduction and visual fidelity.

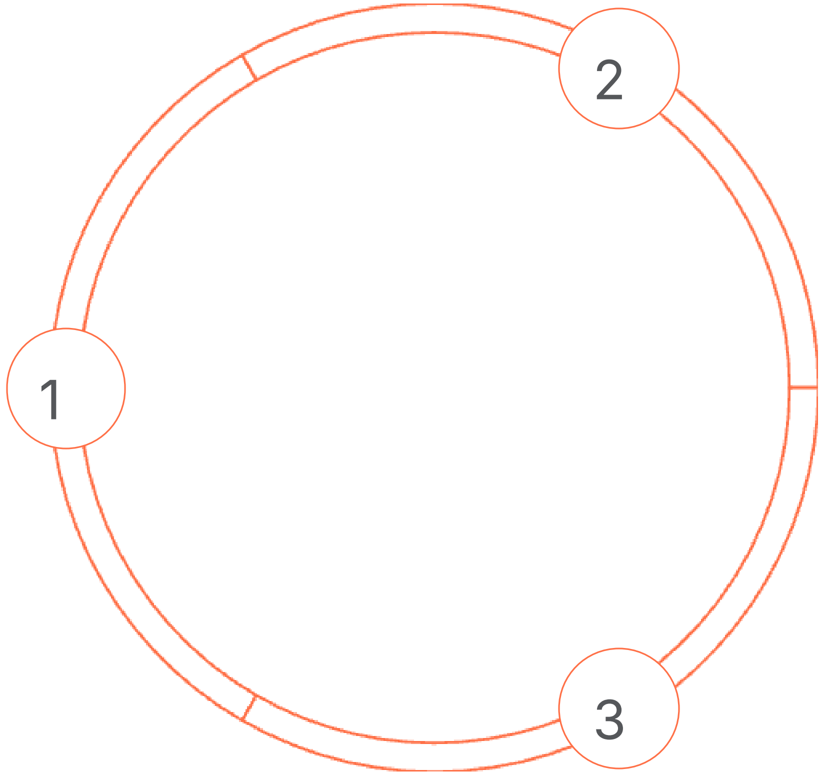
## Quality Levels

- > 3000 KB: Quality 55
- 2000–3000 KB: Quality 65
- 1000–2000 KB: Quality 75
- < 1000 KB: Quality 85

## Output

Compressed images saved to Final_Output, with detailed metadata logging for logging for mission monitoring.

# Results & Examples




## Good Images

34% of images met quality criteria including horizon and Earth visibility.

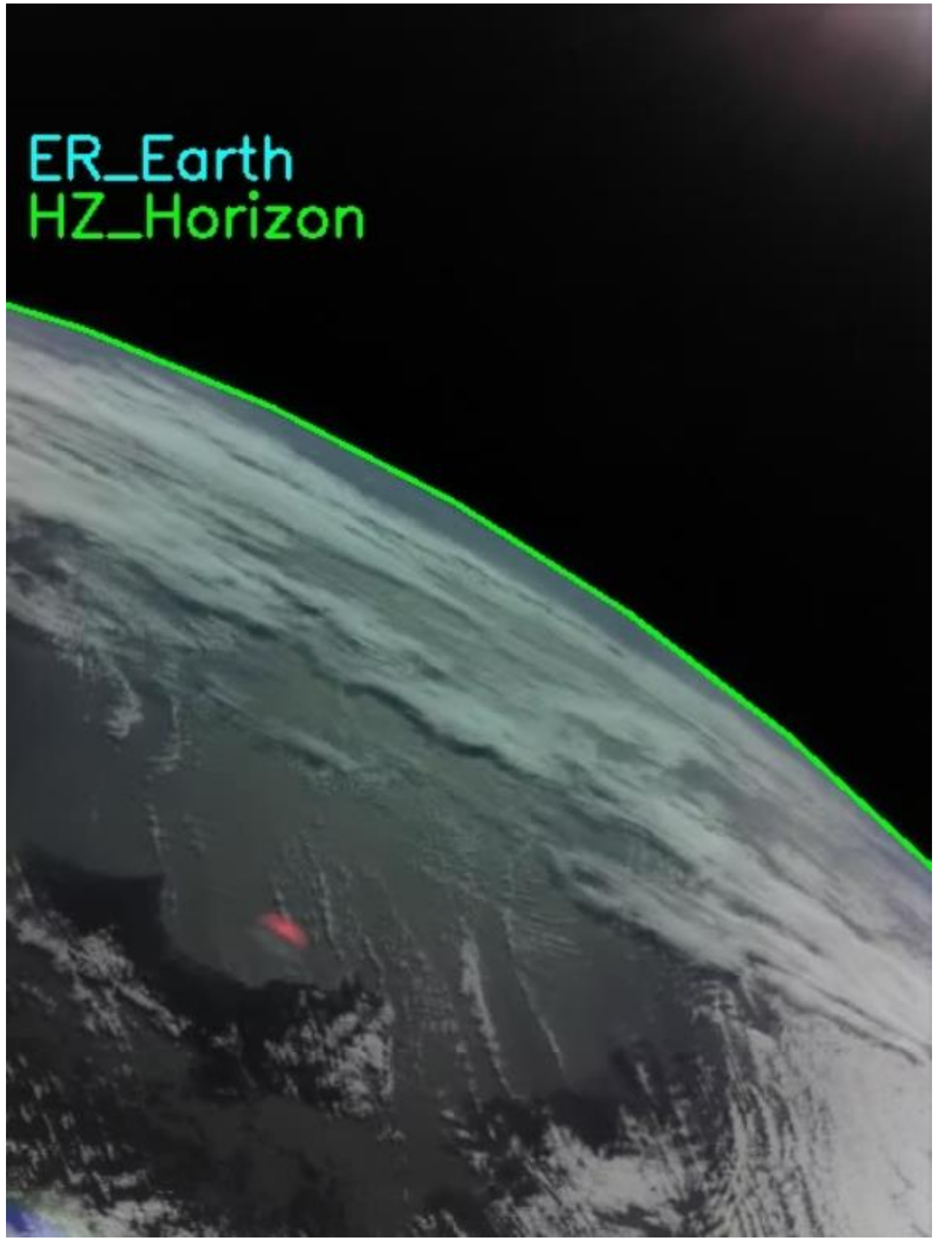
## Rejected Images

66% were discarded due to defects like sunburn glare, blur, black frames, or lack of relevant content.

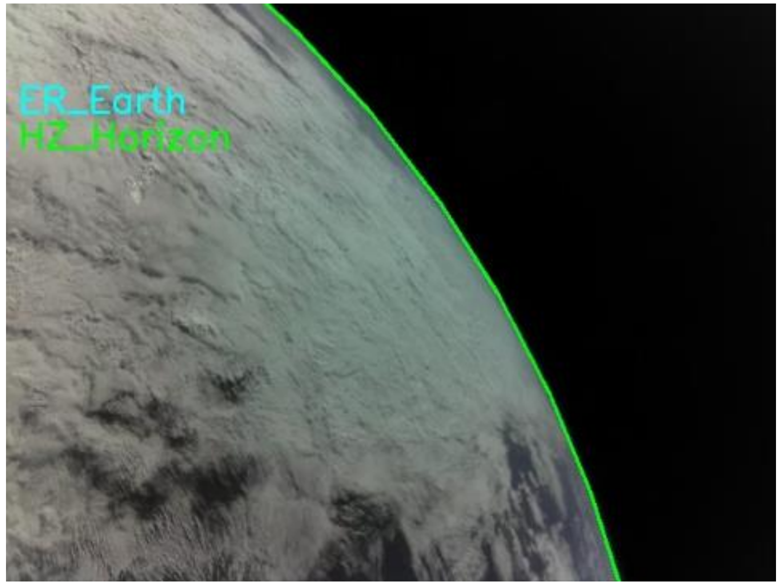
## Compression

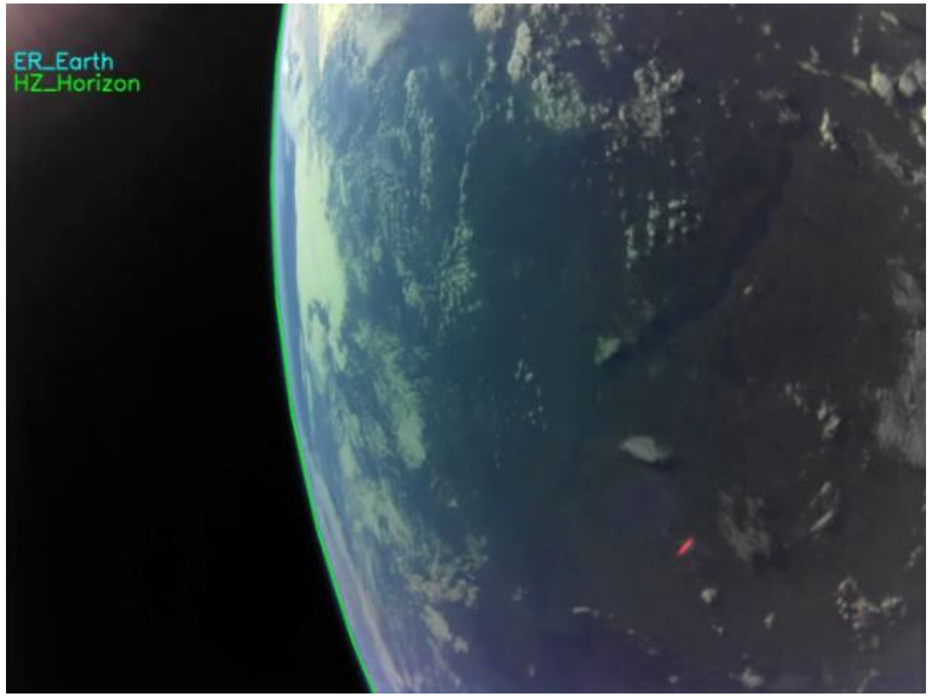
File size reduced by 40–70%, with preservation preservation of essential visual quality for transmission optimization.

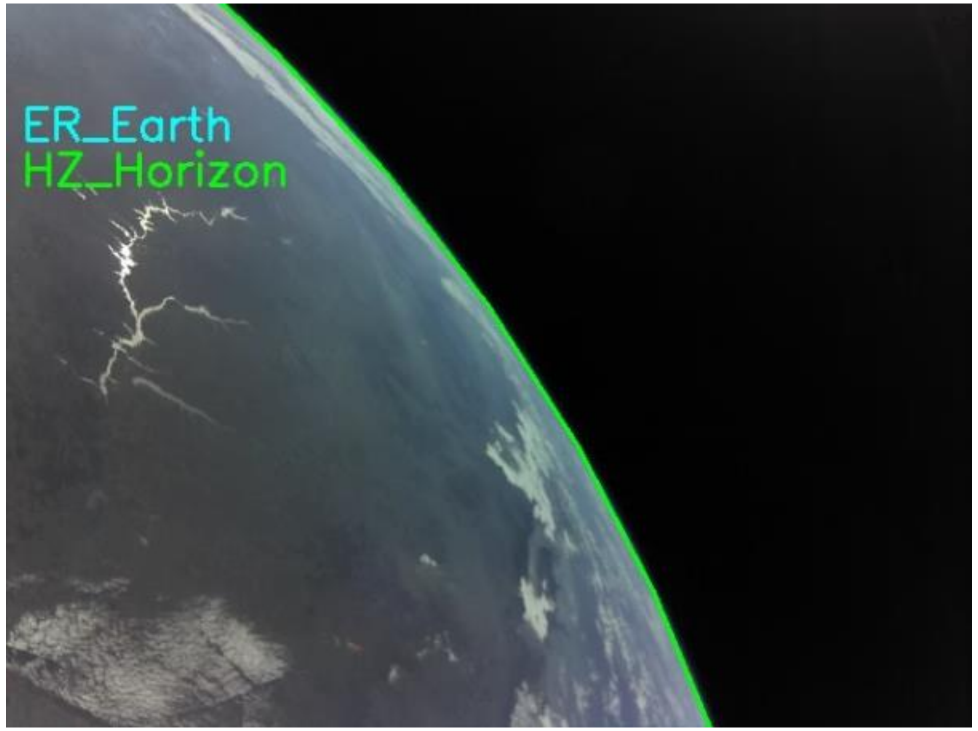
# result as good images:



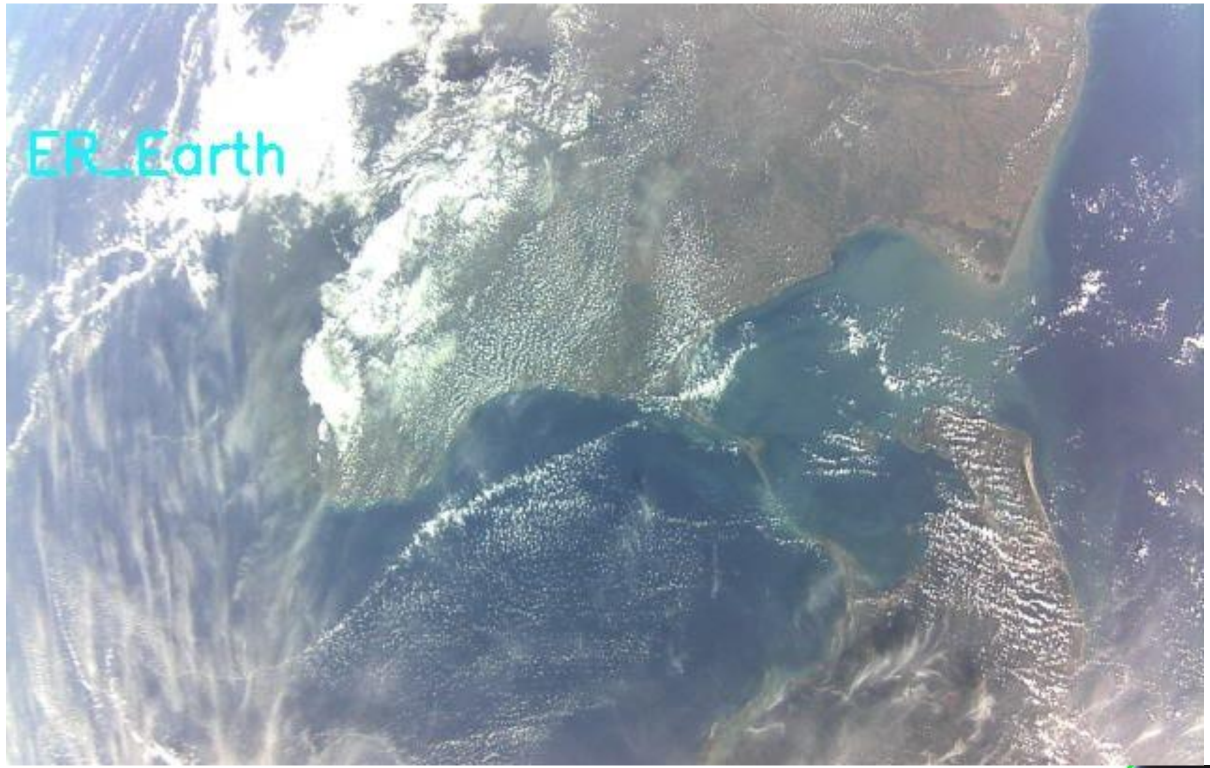

# another good images:




# Bad images:

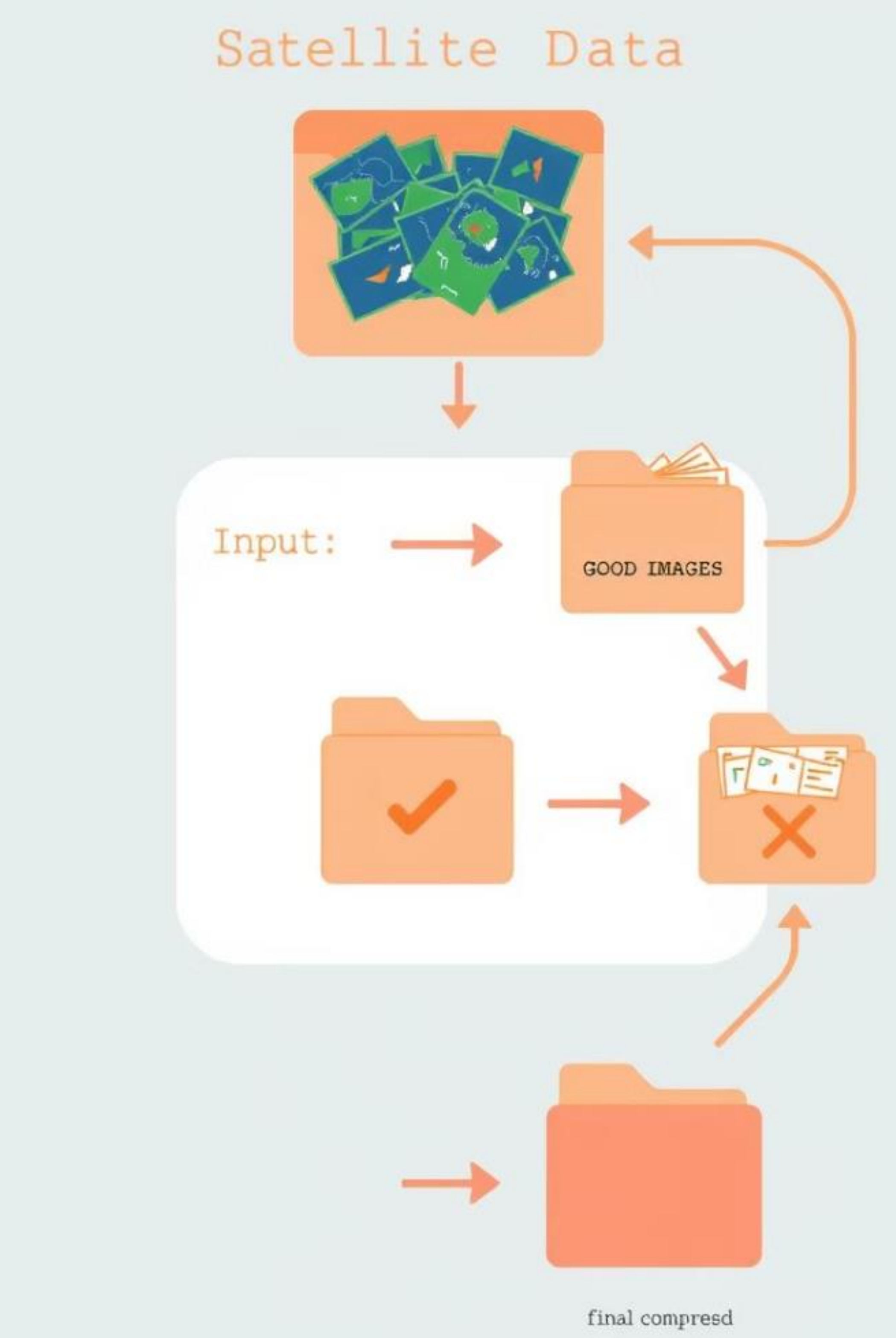




# Compression & Output Analysis

🔧 Starting image compression...
📁 Input folder : DataSet/Output_Image/Good_Image
📁 Output folder: DataSet/Output_Image/Final_Output

| 📄 Filename | Original (KB) | Compressed (KB) | Ratio | Quality | Status |
|---|---|---|---|---|---|
| B3_ER_640x480_119.jpg | 97.9 | 57.2 | 58% | 85 | ✅ Compressed |
| B3_ER_640x480_121.jpg | 106.0 | 66.1 | 62% | 85 | ✅ Compressed |
| B3_ER_640x480_131.jpg | 66.6 | 39.5 | 59% | 85 | ✅ Compressed |
| B3_ER_640x480_133.jpg | 119.9 | 69.8 | 58% | 85 | ✅ Compressed |
| B3_ER_640x480_143.jpg | 75.4 | 46.9 | 62% | 85 | ✅ Compressed |
| B3_ER_640x480_150.jpg | 106.0 | 66.1 | 62% | 85 | ✅ Compressed |
| B3_ER_640x480_155.jpg | 76.9 | 48.3 | 63% | 85 | ✅ Compressed |
| B3_ER_640x480_2.jpg | 63.9 | 37.3 | 58% | 85 | ✅ Compressed |
| B3_ER_640x480_37.jpg | 111.1 | 64.6 | 58% | 85 | ✅ Compressed |
| B3_ER_640x480_44.jpg | 72.3 | 42.6 | 59% | 85 | ✅ Compressed |
| B3_ER_640x480_51.jpg | 66.8 | 36.8 | 55% | 85 | ✅ Compressed |
| B3_ER_640x480_53.jpg | 80.4 | 49.3 | 61% | 85 | ✅ Compressed |
| B3_ER_640x480_62.jpg | 102.1 | 60.0 | 59% | 85 | ✅ Compressed |
| B3_ER_640x480_78.jpg | 96.4 | 60.1 | 62% | 85 | ✅ Compressed |
| B3_ER_640x480_83.jpg | 77.7 | 47.9 | 62% | 85 | ✅ Compressed |
| B3_ER_640x480_85.jpg | 75.4 | 43.8 | 58% | 85 | ✅ Compressed |
| B3_ER_640x480_92.jpg | 101.5 | 63.2 | 62% | 85 | ✅ Compressed |
| B3_ER_640x480_99.jpg | 63.9 | 34.3 | 54% | 85 | ✅ Compressed |
| B3_HZ_320x240_127.jpg | 24.5 | - | - | - | ⚠️Skipped (too small) |
| B3_HZ_640x480_100.jpg | 62.4 | 37.0 | 59% | 85 | ✅ Compressed |
| B3_HZ_640x480_104.jpg | 62.1 | 33.7 | 54% | 85 | ✅ Compressed |
| B3_HZ_640x480_109.jpg | 69.7 | 43.7 | 63% | 85 | ✅ Compressed |
| B3_HZ_640x480_120.jpg | 60.0 | 36.2 | 60% | 85 | ✅ Compressed |
| B3_HZ_640x480_124.jpg | 73.9 | 44.1 | 60% | 85 | ✅ Compressed |
| B3_HZ_640x480_13.jpg | 90.2 | 52.3 | 58% | 85 | ✅ Compressed |
| B3_HZ_640x480_152.jpg | 47.6 | 29.1 | 61% | 85 | ✅ Compressed |
| B3_HZ_640x480_29.jpg | 54.3 | 32.4 | 60% | 85 | ✅ Compressed |
| B3_HZ_640x480_35.jpg | 93.6 | 54.7 | 59% | 85 | ✅ Compressed |
| B3_HZ_640x480_40.jpg | 66.4 | 40.2 | 60% | 85 | ✅ Compressed |
| B3_HZ_640x480_47.jpg | 60.9 | 36.3 | 60% | 85 | ✅ Compressed |
| B3_HZ_640x480_48.jpg | 52.4 | 31.0 | 59% | 85 | ✅ Compressed |
| B3_HZ_640x480_50.jpg | 77.7 | 48.6 | 63% | 85 | ✅ Compressed |
| B3_HZ_640x480_52.jpg | 66.5 | 36.7 | 55% | 85 | ✅ Compressed |
| B3_HZ_640x480_54.jpg | 55.8 | 34.0 | 61% | 85 | ✅ Compressed |
| B3_HZ_640x480_55.jpg | 60.6 | 33.7 | 56% | 85 | ✅ Compressed |
| B3_HZ_640x480_59.jpg | 86.2 | 51.6 | 60% | 85 | ✅ Compressed |
| B3_HZ_640x480_91.jpg | 53.1 | 31.9 | 60% | 85 | ✅ Compressed |
| B3_HZ_640x480_96.jpg | 63.8 | 39.2 | 61% | 85 | ✅ Compressed |
| B4_ER_320x240_15_complete.jpg | 25.8 | - | - | - | ⚠️Skipped (too small) |
| B4_ER_320x240_19_complete.jpg | 26.5 | - | - | - | ⚠️Skipped (too small) |
| B4_ER_320x240_1_complete.jpg | 22.2 | - | - | - | ⚠️Skipped (too small) |
| B4_ER_320x240_6_complete.jpg | 25.2 | - | - | - | ⚠️Skipped (too small) |
| B4_ER_640x480_13_complete.jpg | 23.6 | - | - | - | ⚠️Skipped (too small) |
| B4_ER_640x480_36_complete.jpg | 53.8 | 29.5 | 55% | 85 | ✅ Compressed |
| B4_ER_640x480_46_incomplete.jpg | 64.2 | 36.9 | 57% | 85 | ✅ Compressed |
| B4_ER_640x480_49_incomplete.jpg | 63.7 | 37.6 | 59% | 85 | ✅ Compressed |
| B4_ER_640x480_63_complete.jpg | 65.2 | 37.4 | 57% | 85 | ✅ Compressed |
| B4_ER_640x480_87_complete.jpg | 70.7 | 40.1 | 57% | 85 | ✅ Compressed |
| B4_HZ_320x240_89_complete.jpg | 17.6 | - | - | - | ⚠️Skipped (too small) |
| B4_HZ_640x480_75_complete.jpg | 44.0 | 27.0 | 61% | 85 | ✅ Compressed |

✅ Compression process complete.



# Conclusion & Future Work

## Achievements

Developed autonomous pipeline for efficient image filtering and compression optimizing CubeSat constraints with minimal quality loss.

## Limitations

Current rule-based approach limits adaptability and can miss borderline cases due to cases due to static thresholds.

## Next Steps

- Integrate machine learning classifiers like CNNs for improved accuracy. accuracy.
- Deploy on real satellite hardware such as FPGA or AI accelerators.
- Expand analysis with multimodal data and advanced scene understanding. understanding.

Our vision is to enable intelligent autonomous image selection onboard future satellites, satellites, boosting the value and efficiency of space data transmission.