🌿 Ecobloom Plant-Stress Detection Challenge – Technical Report

Objective

The main aim of this project was to build a robust yet simple pipeline that can highlight stress regions on plant leaves using RGB images. The pipeline should work across three plant species—basil, strawberry, and cucumber—while keeping things lightweight, explainable, and fast (under 2 minutes per image).

Dataset & Visual Analysis

The dataset contains roughly 1,200 RGB images:

- 400+ for each of basil, strawberry, and cucumber
- A few demo images that show clear examples of healthy vs. stressed leaves

What We Observed:

- **Healthy leaves** were uniformly bright green
- **Early stress** showed yellowing, especially toward the center
- **Medium stress** introduced orange to light brown hues
- **Severe stress** areas turned dark red to nearly black

This pattern—a gradient from green edges to reddish centers—was consistent across all plant types. That insight drove the design of our color-based detection logic.

Pipeline Design (Classical CV Approach)

I chose a classical approach for its transparency and ease of tuning under time constraints. The core idea was to segment healthy green regions and treat everything else as potentially stressed.

Step-by-Step Overview:

1. **Convert to HSV color space** for better color segmentation
2. **Segment healthy green regions** using a broad green HSV range
3. **Invert the mask** to get candidate stress regions
4. **Apply morphological cleaning** to reduce noise and refine regions
5. **Extract visual patches** from healthy and stressed masks
6. **Overlay labels** ("Healthy Leafs" and "Severe") for interpretability

```
lower_green = np.array([25, 20, 20])
upper_green = np.array([90, 255, 255])
```

These thresholds performed well across most examples. The mask was cleaned using a simple 5x5 kernel for opening and closing.

Result Presentation:

- Healthy regions were extracted and overlaid with a red label
- Inverse regions were shown as potential stress zones

One Failure Case

On some strawberry leaves with **medium stress**, the orange region in the center was not well-captured. Since orange is closer to green than deep red in HSV, it was excluded by the threshold.

Why this matters:

Medium stress is critical—it's when intervention is most effective. Missing these regions could delay timely decisions.

Improvements & Future Work

To improve this:

- **Switch to LAB color space**: It's more perceptually uniform than HSV, making subtle changes easier to detect
- **Profile central gradients**: Stress often starts at the center, so modeling that spatially can help
- **Use instance segmentation (Mask R-CNN)** if more time is available—it's more precise, but requires detailed annotations

For now, color clustering (like K-Means) or basic classifiers on patch-level data could give better granularity.

Scaling to Other Plants

To generalize the solution:

- **Adjust HSV/LAB ranges** by profiling new species
- **Use few-shot learning**: Annotate a handful of leaves and fine-tune a simple model

- **Self-supervised learning**: Methods like BYOL can help learn the concept of "healthy" vs. "stressed" with very little annotation
- **Separate leaves before analysis**: Crop the leaf, then apply stress detection

---

✍️ Final Thoughts

This solution strikes a good balance between clarity, speed, and flexibility. It’s interpretable by design, adaptable to different species, and easily extensible. Given more time, I’d explore hybrid models that use color logic for rough segmentation and ML models for refinement.

Still, in its current form, the pipeline demonstrates the potential for low-cost, real-time plant monitoring systems that can run on edge devices or in field conditions.
