Zero-Shot Leaf Health Classification

1. Introduction

At Hexafarms, an agricultural technology company based in Berlin, during my internship, I was tasked to implement automation in image annotations related to plant health monitoring for greenhouses. The main goal was to come up with a classifier capable of distinguishing the sections of images that might come from greenhouses into classes such as healthy, unhealthy, and background, which would be able to take some workload off the plant experts, scaling the process faster.

Following are a few of the key challenges which the project had aimed to address:

- Processing images from multiple cameras across different greenhouses
- Handling complex scenes with overlapping leaves and varying environmental conditions
- Reducing the need for manual annotation of large datasets
- Providing a scalable solution for rapid plant health assessment

2. Methodology

2.1 Data Processing

It cuts the input images into smaller sections. That means each image can be analyzed at a much finer scale for the health of a plant.

The main components of the data processing pipeline:

- Image slicing in specified slice size and overlap
- Slice size adaptation according to image resolution-384x384 pixels for Full HD and lower; for larger images, 256x256 pixels
- Overlap of 20% between slices to ensure comprehensive coverage

2.2 Model Ensemble

We implemented an ensemble of five state-of-the-art zero-shot learning models:

1. ViT-SO400M-14-SigLIP-384
2. ViT-H-14-378-quickgelu
3. ViTamin-XL-384
4. ViT-bigG-14-CLIPA-336
5. EVA02-E-14-plus

These models are chosen for their strengths and generalization in zero-shot image classification, as well as data the model has never seen. Zero-shot learning models in this regard are quite efficient because they do not need to be trained on plant disease data, which is in general limited or unavailable Xian et al. (2018).

2.3 Classification Process

For each image slice, the following steps are performed:

1. Slice preprocessing to the format required by each model
2. Encoding healthy, unhealthy, and background classes as pre-defined textual prompts
3. Calculation of the similarity between image and textual features
4. Determination of class based on maximum similarity

2.4 Majority Voting

To gain robustness and reduce biases from individual models, one can implement majority voting in which:

- Each model votes for a class for each slice
- The final classification is the majority vote
- A minimum threshold of 3 votes is required for a definitive classification; otherwise, the slice is classified as "undecided"

2.5 Visualization

The output of the system will visually include:

- The individual model prediction per slice.
- The final majority vote result
- Colored annotations: Green = Healthy, Red = Unhealthy, Blue = Background, Yellow = Undecided.

3. Implementation

The project was implemented in Python, leveraging the following key libraries:

- OpenCLIP: For zero-shot image classification models
- PyTorch: For efficient GPU-accelerated computations
- OpenCV: For image processing tasks
- Matplotlib: For visualization of results

Key components of the implementation:

```
# Configuration
SLICE_SIZE_FHD = 384
SLICE_SIZE_LARGE = 256
OVERLAP = 0.2
VOTE_THRESHOLD = 3

# List of models
MODELS = [
    ('ViT-SO400M-14-SigLIP-384', 'webli'),
    ('ViT-H-14-378-quickgelu', 'dfn5b'),
    ('ViTamin-XL-384', 'datacomp1b'),
    ('ViT-bigG-14-CLIPA-336', 'datacomp1b'),
    ('EVA02-E-14-plus', 'laion2b_s9b_b144k'),
]

# Prompts for each class (Prompt Engineering)
PROMPTS = {
    'healthy': [
        "Green, normal-looking plant leaf or leaves",
        "Plant foliage appearing healthy and undamaged",
        "Leaf surface with natural color and texture",
        "Vibrant leaf or leaves without visible issues",
    ],
    'unhealthy': [
        "Plant leaf or leaves showing discoloration or spots",
        "Foliage with visible signs of disease or stress",
        "Leaf area with unusual texture or damage",
        "Plant leaves displaying abnormal colors or patterns",
    ],
    'background': [
        "Non-leaf elements such as soil, sky, or structures",
        "Area without any visible plant leaves",
        "Greenhouse materials or equipment",
        "Plant parts other than leaves, like fruits or stems",
```

```
        "Blurred or indistinct background elements",
    ]
}

def process_image(image_path, output_dir):
    # Implementation details...

def process_directory(input_dir, output_dir):
    # Implementation details...
```

Prompt Engineering

This implementation has a critical aspect in terms of the zero-shot prompt engineering. As seen above, the `PROMPTS` dictionary contains examples of text for each class (healthy, unhealthy and background) that are constructed in a specific way. The aim of these prompts is to help the zero-shot models probe into and learn the necessary attributes within any of the classes.

The prompts were made as such that they were quite descriptive but were also a bit diverse to encompass different scenarios:

- Healthy leaf prompts indicate that leaves are green, healthy, and there are no problems.
- Prompts for unhealthy types show yellowing leaves, wilting, and dying plants among other features of plants in distress.
- Where background elements , which are background or sections of images, excludes leaves but is common in greenhouse images materials of plants and their surroundings to help the model differentiate the plant material from background.

In this way, the possibility of making such decisions is provided for the developed models and, which is important in our case, the models are not trained specifically for a particular dataset of plant diseases, which is a big plus considering the nature of the greenhouse that we operate in.

Script Workflow

1. Image Processing:
    - The `process_image` function loads an input image and slices it into smaller sections using the defined `SLICE_SIZE` and `OVERLAP` parameters.
    - Adaptive slice sizing is used based on the image resolution to ensure consistent processing across various image sizes.
2. Model Application:
    - Each slice is processed through all five models listed in the `MODELS` array.
    - For each model, the image features are extracted and compared against the text features generated from the prompts.
    - The class with the highest similarity score is assigned to each slice for each model.
3. Majority Voting:
    - After all models have classified a slice, a majority voting mechanism is applied.
    - A class is only assigned if it receives at least `VOTE_THRESHOLD` (3) votes; otherwise, the slice is marked as "undecided".
4. Visualization:
    - The results from each model and the final majority vote are visualized using color-coded annotations on the original image.
    - Green represents healthy areas, red for unhealthy, blue for background, and yellow for undecided regions.
5. Batch Processing:
    - The `process_directory` function allows for batch processing of multiple images in a given directory, making it scalable for large-scale greenhouse monitoring.

4. Results and Analysis

The system was tested on a sample image of hydroponic strawberries in a greenhouse environment. Before diving into the specific results, it's important to address some key challenges and limitations of this project.

4.1 Limitations and Challenges

1. Lack of Quantitative Metrics:
   The absence of quantitative metrics is a significant limitation in this project. This is due to the lack of annotated production data from Hexafarms' greenhouses. The primary reason for not having these metrics is that the project's main goal was to avoid manual annotation of images or slices, which is time-consuming and expensive. Without a ground truth dataset, it's challenging to calculate standard evaluation metrics such as accuracy, precision, or recall.

   The results were primarily evaluated through visual inspection by myself, the CTO, and a plant expert. While we all agreed that the results were promising, this method of evaluation is subjective and doesn't provide quantifiable performance measures.

2. Previous Attempts with Traditional Image Classifiers:
   Before implementing the zero-shot learning approach, we attempted to train a traditional image classifier using publicly available plant disease datasets. However, this approach failed to perform adequately on our production data. The main reasons for this failure were:

   a. Domain Shift: The publicly available datasets typically contain images of single leaves in controlled environments with clear backgrounds. In contrast, our production data features complex scenes with overlapping leaves, inconsistent lighting, and crowded greenhouse environments.

   b. Uncontrolled Environment: Our greenhouse images contain leaves in various orientations, partially obscured by other plants or greenhouse structures, and under varying lighting conditions. This real-world complexity was not well represented in the public datasets.

   c. Lack of Specificity: The public datasets didn't necessarily include the specific plant varieties or conditions present in Hexafarms' greenhouses.

3. Advantages of the Zero-Shot Learning Approach:
   The zero-shot learning technique implemented in this project performed better than the traditional image classifier for several reasons:

   a. Flexibility: Zero-shot models can adapt to new classes or conditions without retraining, which is crucial given the variability in greenhouse environments.

   b. Generalization: These models can leverage general visual understanding to classify images even in unfamiliar contexts.

   c. No Need for Large Annotated Datasets: Zero-shot learning doesn't require a large, annotated dataset specific to our greenhouse conditions.

4.2 Model Performance Analysis

The output images demonstrate the classification results for each of the five models and the final majority vote.

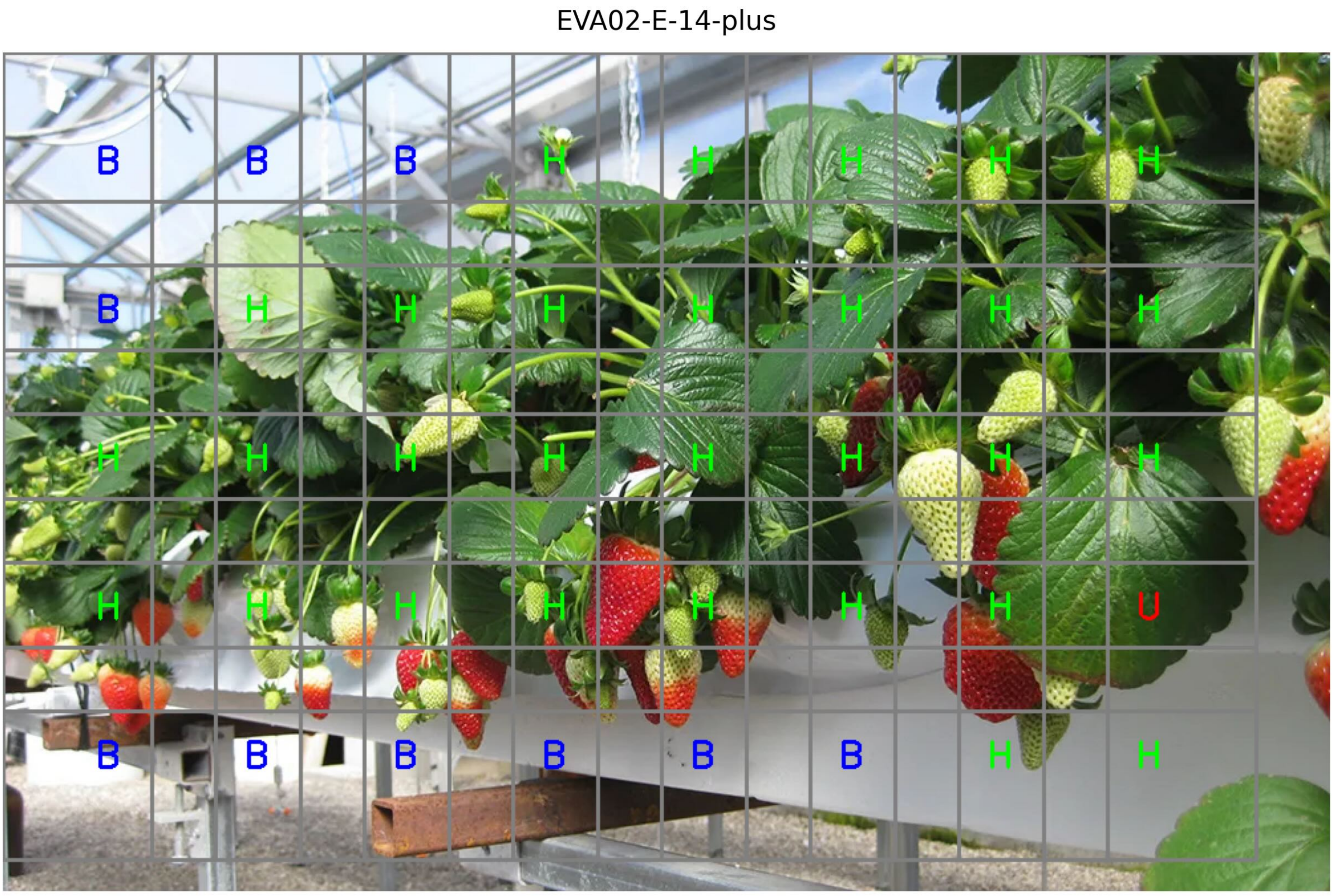

Figure 1: Classification results from the EVA02-E-14-plus model

1. EVA02-E-14-plus (Figure 1):
    - Shows a balanced classification, identifying most leaf areas as healthy (H).
    - Correctly identifies the greenhouse structure as background (B).
    - Detects one unhealthy (U) slice in the lower right corner, which appears to show some leaf discoloration.

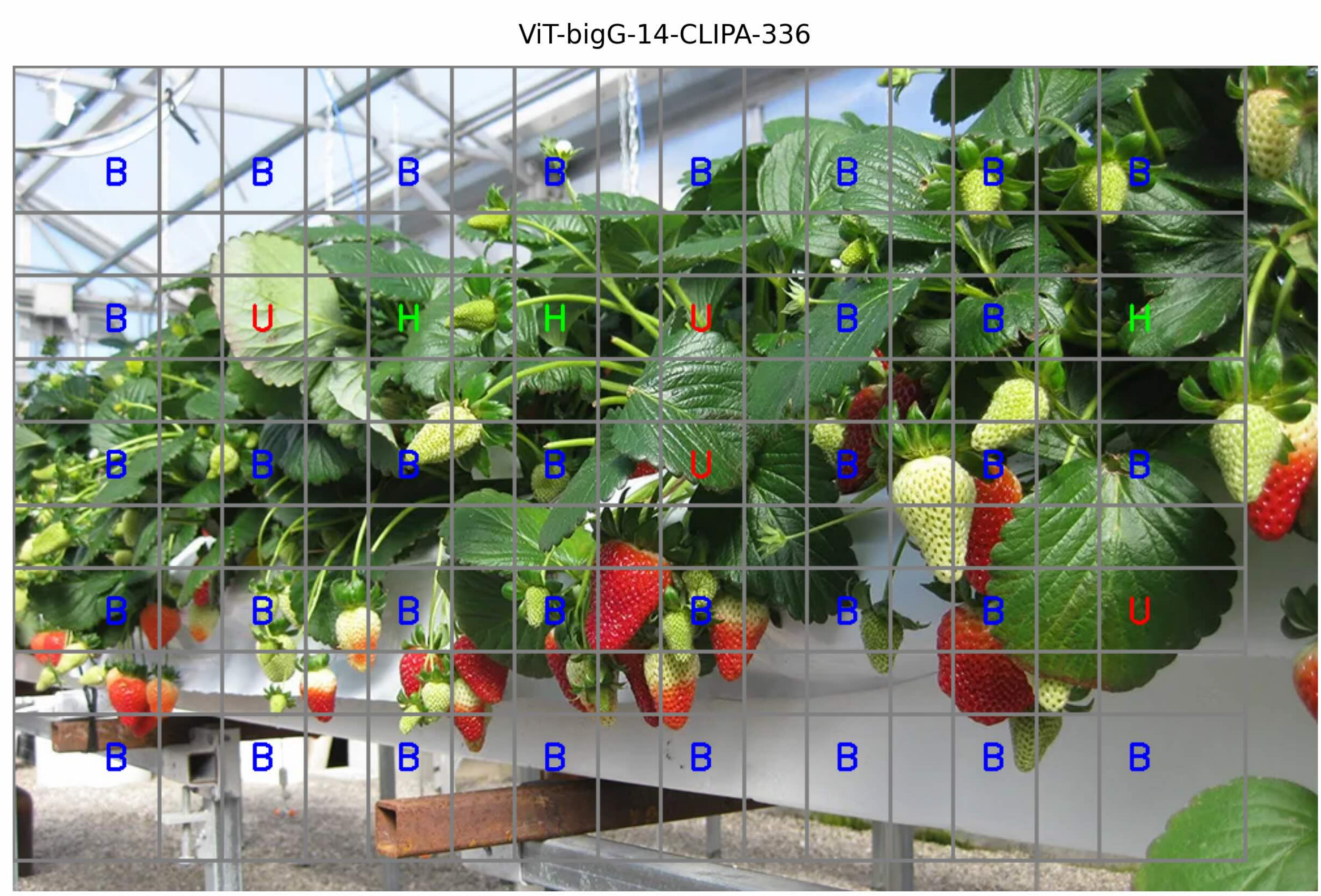

Figure 2: Classification results from the ViT-bigG-14-CLIPA-336 model

2. ViT-bigG-14-CLIPA-336 (Figure 2):
    - More conservative in classifying healthy areas, labeling many leaf regions as background (B).
    - Identifies several unhealthy (U) spots, including areas that other models classify as healthy.
    - Shows higher sensitivity to potential plant stress, but may overclassify background elements.

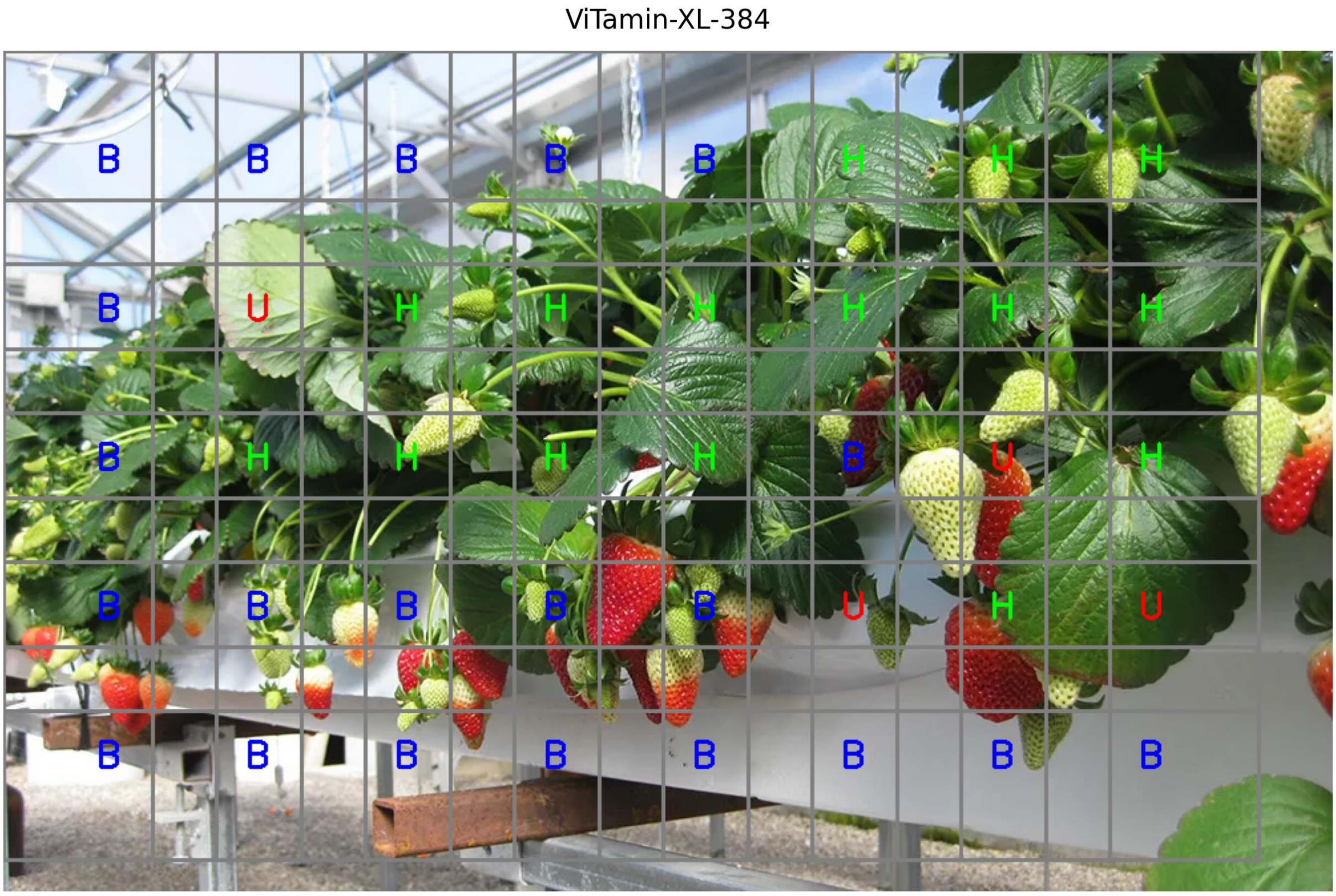

Figure 3: Classification results from the ViTamin-XL-384 model

3. ViTamin-XL-384 (Figure 3):
    - Provides a balanced classification similar to EVA02-E-14-plus.
    - Detects multiple unhealthy (U) areas, including the discolored leaf in the upper left and lower right.
    - Shows good discrimination between plant material and background elements.

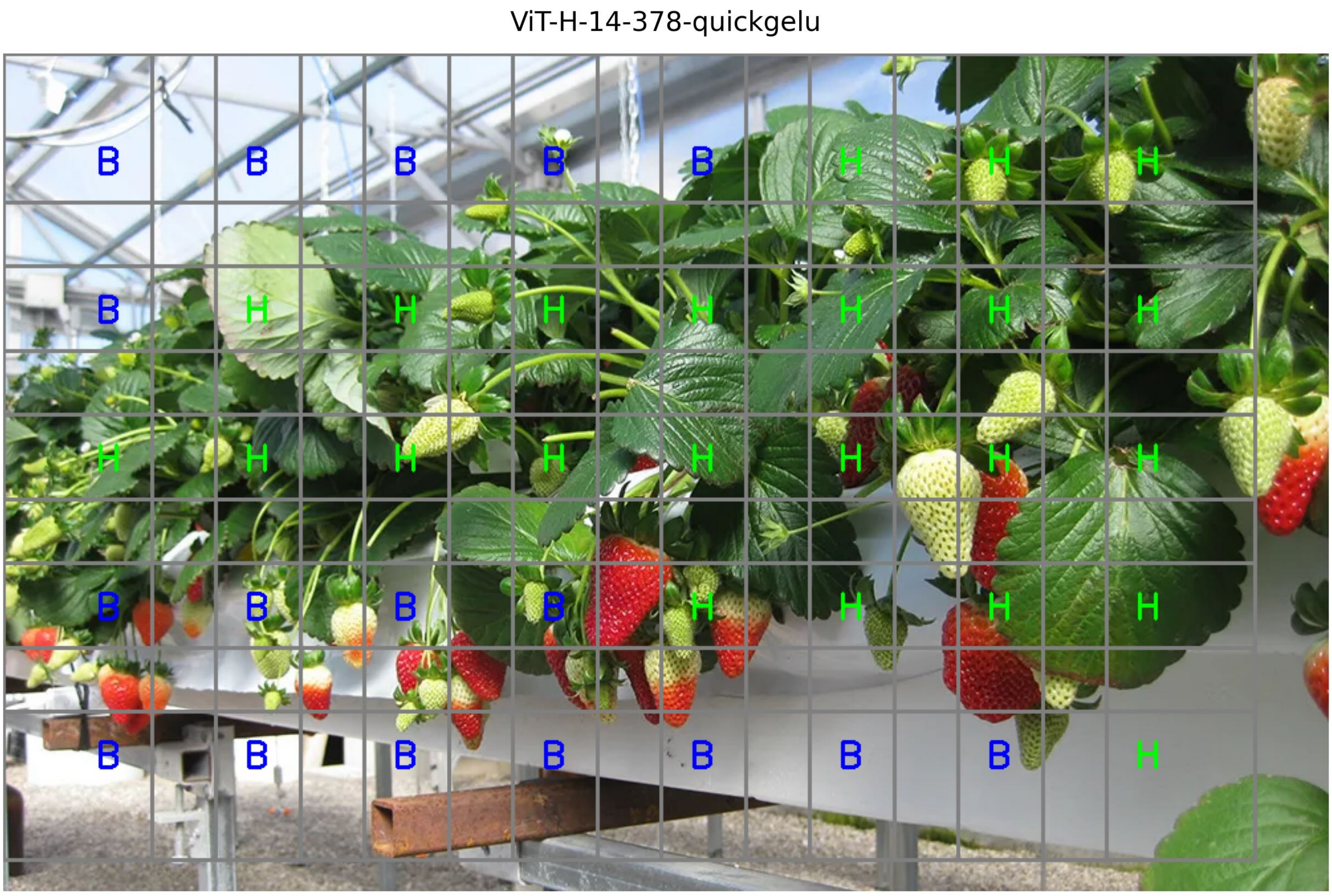

Figure 4: Classification results from the ViT-H-14-378-quickgelu model

4. ViT-H-14-378-quickgelu (Figure 4):
    - Most conservative in identifying unhealthy areas, classifying almost all plant material as healthy (H).
    - Accurately identifies background elements.
    - May be less sensitive to subtle signs of plant stress.

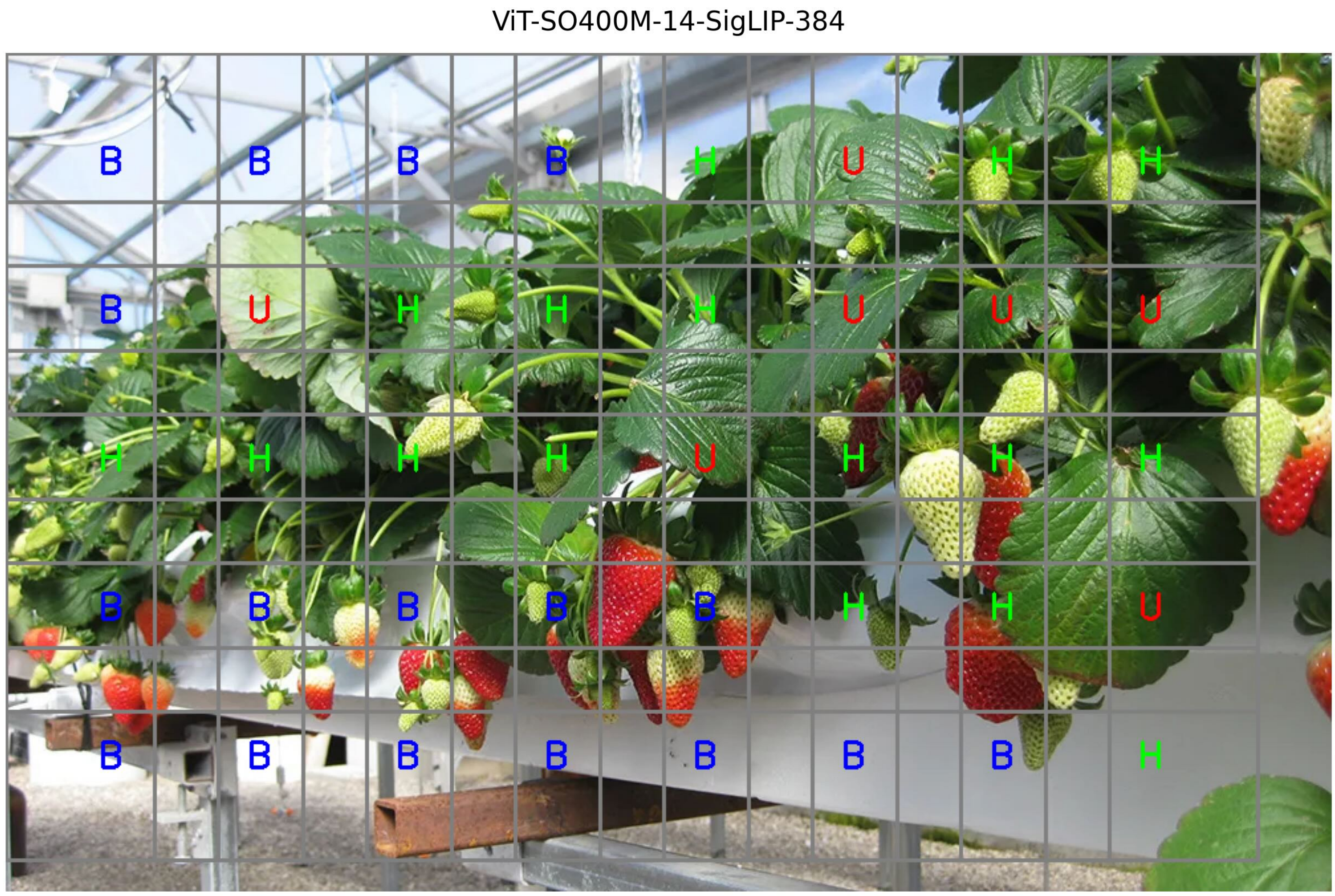

Figure 5: Classification results from the ViT-SO400M-14-SigLIP-384 model

5. ViT-SO400M-14-SigLIP-384 (Figure 5):
    - Shows the highest sensitivity to potential plant stress, identifying numerous unhealthy (U) areas.
    - Correctly identifies background elements.
    - May be oversensitive, potentially leading to false positives for unhealthy classification.

4.3 Majority Vote Analysis

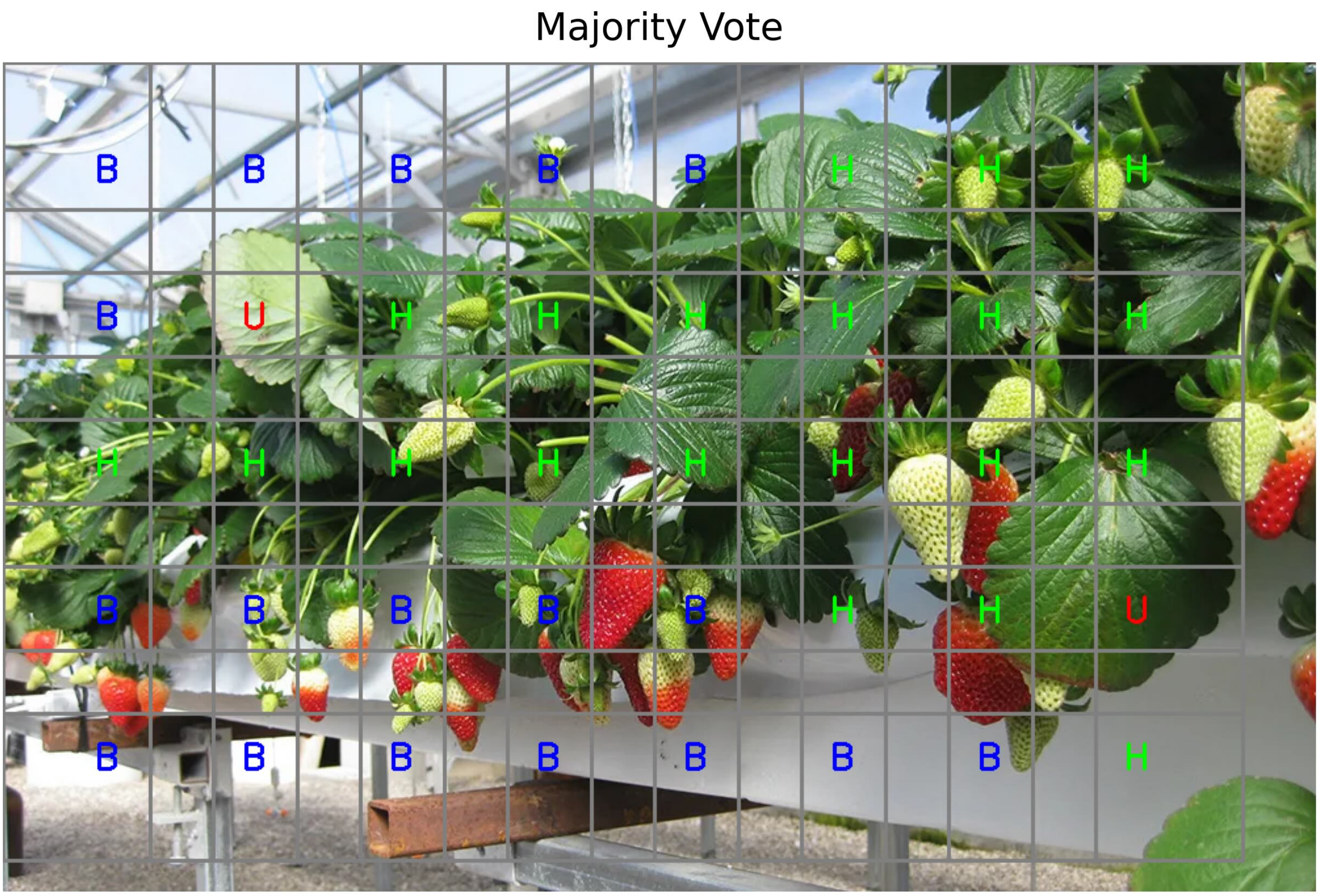

Figure 6: Final classification results after majority voting

The majority vote result (Figure 6) effectively combines the strengths of each model:

- Correctly identifies the greenhouse structure and non-plant elements as background (B).
- Classifies most leaf areas as healthy (H), which aligns with the overall appearance of the plants.
- Identifies two key unhealthy (U) areas:
    1. A leaf in the upper left corner, which shows visible discoloration.
    2. A leaf in the lower right corner, which also exhibits signs of stress or disease.

4.4 Key Observations

1. Consensus on Background: All models consistently identify the greenhouse structure and non-plant elements as background, demonstrating the system's ability to distinguish between plant material and the surrounding environment.
2. Variation in Healthy/Unhealthy Classification: There is some disagreement among models in classifying healthy versus unhealthy plant material. This variation highlights the challenge of detecting subtle signs of plant stress.
3. Effectiveness of Majority Voting: The majority vote successfully filters out potential false positives from oversensitive models (like ViT-SO400M-14-SigLIP-384) while still capturing significant unhealthy areas identified by multiple models.
4. Detection of Subtle Issues: The system successfully identifies areas of potential concern (the two unhealthy leaves) that might require closer inspection by a plant expert.
5. Handling of Complex Scenes: The models demonstrate an ability to handle the complex greenhouse environment, with overlapping leaves, varying light conditions, and the presence of fruits (strawberries) in the image.

4.5 Additional Limitations

1. Potential for False Classifications: While the system shows promise, we observed some false classifications during visual inspections. This highlights the need for further refinement and possibly the integration of domain-specific knowledge.

2. Challenges with Production Data: The uncontrolled nature of the greenhouse environment, with its overlapping leaves, inconsistent lighting, and crowded scenes, presents ongoing challenges for accurate classification.
3. Sensitivity Calibration: The variation in sensitivity across models suggests that further calibration might be necessary to optimize the balance between detecting genuine issues and avoiding false alarms.

5. Conclusion

The automated leaf health classification system developed during this internship project at Hexafarms demonstrates the potential of zero-shot learning models in agricultural monitoring applications. By leveraging an ensemble of advanced models and a majority voting mechanism, the system can provide rapid assessments of plant health without the need for extensive manual annotation.

Key achievements:

- Successful implementation of a multi-model ensemble for zero-shot classification
- Development of an adaptive image processing pipeline for various image resolutions
- Creation of a visualization system for easy interpretation of results
- Improved performance compared to traditional image classifiers on complex greenhouse data

Despite the limitations, the zero-shot learning approach has shown potential in addressing the unique challenges of Hexafarms' greenhouse monitoring needs. The system's ability to provide useful classifications without requiring a large, annotated dataset makes it a promising solution for rapid deployment and scalability.

Future work and improvements:

- Collection and annotation of a small set of production data for quantitative evaluation
- Fine-tuning of models on domain-specific data to improve accuracy
- Exploration of semi-supervised learning techniques to leverage unlabeled production data
- Integration of temporal analysis to track plant health over time
- Further calibration of model sensitivities to reduce false positives/negatives

This project lays the groundwork for more efficient and scalable plant health monitoring in greenhouse environments, potentially reducing labor costs and improving crop management practices. The use of zero-shot learning models offers a promising approach to overcome the limitations of traditional supervised learning in agricultural applications where labeled data is scarce.

References

Barbedo, J.G.A., 2018. Impact of dataset size and variety on the effectiveness of deep learning and transfer learning for plant disease classification. Computers and Electronics in Agriculture, 153, pp.46-53.

Chao, X., Zhang, L., Li, Y. and Zhou, B., 2016. Recognition of greenhouse plant diseases and pests based on image processing technology. Journal of Sensors, 2016.

Kuncheva, L.I. and Rodríguez, J.J., 2014. A weighted voting framework for classifiers ensembles. Knowledge and Information Systems, 38(2), pp.259-275.

Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J. and Krueger, G., 2021. Learning transferable visual models from natural language supervision. In International Conference on Machine Learning (pp. 8748-8763). PMLR.

Xian, Y., Lampert, C.H., Schiele, B. and Akata, Z., 2018. Zero-shot learning—A comprehensive evaluation of the good, the bad and the ugly. IEEE transactions on pattern analysis and machine intelligence, 41(9), pp.2251-2265.

GitHub Repository

https://github.com/30-A/Zero-Shot-Leaf-Health-Classification