Plant growth analysis : Leaf counting algorithm

Overview

This project implements an image processing pipeline using OpenCV to enhance images of leaves, segment them, and count the number of leaves in each image. The results are visualized using Matplotlib. The code is designed to process a series of images taken over a period of 14 days, tracking the growth of plants.

Requirements

To run this code, you need to have the following Python packages installed:

- opencv-python
- numpy
- matplotlib
- notebook (if not already installed)

You can install these packages using pip:

1. **pip install opencv-python numpy matplotlib**

Directory Structure

The code expects the following directory structure:

text

```
project_root/
│
├── images/
│   ├── input/        # Input images should be placed here
│   └── output/        # Output images will be saved here
│
└── script.py         # Python script containing the code
```

Make sure to place your input images in the images/input/ directory. Rename images as Day{day_number}_T_06.jpg.

The out will be saved in images/output/enhanced_images/ and images/output/

How to Run the Notebook

1. Open the Jupyter Notebook:
    - Navigate to the directory containing your notebook file and run jupyter notebook
    - Open Leaf_count.ipynb from the browser interface.
2. Run the cell:
    - The notebook will process the images, save outputs, and display a plot at the end.

Code Description

Functions

1. image_enhance(image, output_path, filename)

This function enhances the input image by isolating green regions (leaves) using HSV color space, applying morphological operations to clean up noise, and utilizing the watershed algorithm for segmentation.

- **Parameters**:
    - image: The input image in BGR format.
    - output_path: The path where enhanced images will be saved.
    - filename: The name of the input file for saving purposes.
- **Returns**: A binary image representing the segmented leaves.

2. contour_draw(image, enhanced_image, output_path, filename)

This function finds contours in the enhanced image and draws bounding rectangles around detected leaves.

- **Parameters**:
    - image: The original input image.
    - enhanced_image: The binary image obtained from the segmentation process.
    - output_path: The path where contour images will be saved.
    - filename: The name of the input file for saving purposes.
- **Returns**: The count of detected leaves.

Main Loop

The main loop processes a series of images named Day01_T_06.jpg to Day14_T_06.jpg. For each day:

1. It loads the image.
2. It enhances the image to segment leaves.
3. It counts and draws contours around detected leaves.
4. It stores the leaf counts for visualization.

Visualization

At the end of processing all images, a line plot is generated using Matplotlib to visualize leaf counts over the 14-day period.

Usage

1. Place your leaf images in the images/input/ directory.
2. Run the script using notebook.
3. After execution, check the output directories for enhanced images and contour images.
3. A plot displaying leaf counts over time will be displayed.

Example Output

- Enhanced segmented images will be saved in: images/output/enhanced_images/
- Contour-drawn images will be saved in: images/output/image_contours/
- A plot showing "Plant Growth: Leaf Count Over 14 Days" will be displayed.

Troubleshooting

- If an image is not found, ensure that it exists in the specified input directory and that its filename matches what is expected by the script.
- If you encounter issues with package installations, verify your Python environment and package manager settings.