Universiteit Leiden

Master Computer Science

Automatic Nutrient Deficiency Symptom Detection in Vertical Farming

Name:	Bas van Aalst
Student ID:	2031868
Date:	[21/08/2025]
Specialisation:	Data Science: Computer Science
1st supervisor:	Daan Pelt
2nd supervisor:	Niki van Stein

Master's Thesis in Computer Science

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Acknowledgements

I want to thank all the individuals that have contributed to the completion of my thesis. Thank you to my supervisors Daan Pelt and Niki van Stein from Leiden University for guiding and supervising me for 6 months on this project. Both Daan and Niki sparked new fresh ideas and criticisms to challenge and improve my academic research. I also want to thank my colleagues Mert İmre, Jochem Meuwese, Luca Jäger, Sascha Bussian and Harinath Bijjala at Growy for offering a warm welcome and a helping hand in my six months at the company. I want to thank my co-intern Mojde for all the brainstorm sessions, nice walks and encouragement. My thesis would not have been possible without everyone supporting me.

Abstract

Vertical farming introduces modern challenges in monitoring plant health. Manual stress detection is time-consuming for biologists at Growy and results in inaccuracies. To avoid inaccurate manual labeling of stress, the idea of automatic stress detection is introduced. Images of dwarf blue kale, paksoi, salad rocket and thyme are collected, cleaned and segmented using a combination of depth estimation, thresholding and training a U-Net model. Background segmentation achieved a F1-score of 0.875 while an unstable validation loss of the U-Net model suggests a lack of generalisability. Feature extraction and piecewise-segmented regression are utilised to extract various nutrient deficiency features from images over time and pinpoint the onset of stress. The dataset is split up into two classes: healthy and nutrient-deficient. Due to minimal differences in mean feature values, change point t_c cannot be detected and the dataset is split using visual markers of stress. Image classification models ResNet50, EfficientNet-B0, and DenseNet-121 are trained on the dataset. DenseNet-121 consistently outperformed ResNet50 and EfficientNet-B0 and achieved over 90% accuracy. GRAD-CAM analysis suggests that models trained on background-removed images primarily focus on leaf area when predicting classes instead of relying on background information. The code is available at https://github.com/growx-tech/growy-data-intern-nutrient-stress for the growy-data-users team.

Keywords: machine learning, computer vision, nutrient deficiency, stress detection, image classification, image segmentation, feature extraction

CONTENTS

I Introduction 5
- I-A Research Approach . . . 5
- I-B Outline . . . 6

II Background 7
- II-A Growy . . . 7
- II-B Hardware . . . 8
- II-C Datastructures . . . 8

III Related Work 10
- III-A Nutrient Deficiency Detection . . . 10
- III-B Depth Estimation . . . 11
- III-C Image Segmentation . . . 11
- III-D Image Classification . . . 12

IV Methodology 14
- IV-A Dataset Collection . . . 14
- IV-B Data Cleaning . . . 15
- IV-C Image Preprocessing . . . 16
- IV-D Feature Extraction . . . 16
  - IV-D1 TPCA . . . 18
  - IV-D2 Entropy . . . 19
  - IV-D3 Energy . . . 19
  - IV-D4 Homogeneity . . . 19
  - IV-D5 Contrast . . . 19
- IV-E Piecewise-segmented Regression Analysis . . . 20
- IV-F Image Classification Models . . . 20
- IV-G Training Configuration . . . 21
- IV-H Evaluation Metrics . . . 22

V Results 24
- V-A Background Segmentation . . . 24
- V-B Detection of Stress Initiation . . . 26
- V-C Image Classification Models . . . 27
- V-D GRAD-CAM Analysis . . . 28

VI Discussion 30
- VI-A Limitations . . . 31
- VI-B Future Work . . . 31

VII Conclusion 32

References 33

Appendix A: Nutrient deficiency in cultivars over time 37

Appendix B: ROC-AUC curves 38

Appendix C: Confusion matrices 39

I Introduction

Agriculture plays an important role in food production. Climate change, population growth and limited water resources form challenging problems for traditional farms [1]. Pests, insects, diseases and harsh climates create sub-optimal growth conditions for crops. Traditionally, crops are monitored by humans who increase the well-being of crops with pesticides, fertilizer and regular inspections.

Vertical farming offers a practical and durable approach to the problems of traditional farms [2]. Vertical farms grow crops in vertically stacked layers and incorporate soilless farming techniques within a controlled environment. By creating a separate ecosystem for crops to grow in, vertical farms protect against external factors such as climate, disease and insects. Vertical farms focus on crops which require little space to increase efficiency and optimise limited space. Vertical farms may grow cultivars to optimize their growth process and taste. Cultivars are varieties of crops produced by breeding.

Vertical farming faces unique challenges in the agricultural sector such as temperature regulation, nutrient mix optimization and system design for efficient water use [3]. Cultivars grow in volatile environments while vertical farms face these unique challenges. Consequently, vertical farms require quality assurance inspections during the growth process. To perform quality assurance inspections throughout the farm, robots are often deployed to automatically capture images of cultivars. Cultivar images are analyzed by biologists to optimize the growth process and the vertical farming system.

Biologists spend time manually analyzing cultivar images. Sifting through heaps of images is time-consuming and reduces time spent on effective analysis. Manual stress detection is also prone to human errors. To help biologists save time and reduce human errors in detecting stress, the idea of automatic stress detection in cultivar images is introduced.

I-A Research Approach

The research goal is reducing stressful living conditions for cultivars in the farm. Automatic stress detection can focus on abiotic stress such as drought and nutrient deficiency or biotic stress such as pests and diseases. Vertical farming hardly ever suffers from biotic stress due to a controlled environment, so the focus lies on abiotic stress. This thesis will investigate nutrient deficiency symptom detection in cultivar images. A complex image background may influence the detection of nutrient deficiency symptoms. For this reason, finding a method to effectively detect and remove the background of cultivar images is explored. A test group of cultivars will artificially receive stress and a control group of cultivars will not. To determine the time of stress initiation of the test group and whether images contain nutrient deficient or healthy cultivars, pixel-level feature extraction and regression are

explored to achieve this. To automatically detect nutrient deficiency symptoms in images, three models will be trained on a dataset with labeled images containing nutrient deficient or healthy cultivars. The trained models will learn to predict the class labels of unseen images and therefore detect nutrient deficiency symptoms. In summary, five research questions are investigated:

- **RQ1:** Is it possible to effectively detect and remove the background in cultivar images?
- **RQ2:** Which textural features predict nutrient deficiency symptoms in cultivar images?
- **RQ3:** Is it possible to separate images of healthy cultivars from images of nutrient deficient cultivars through pixel-level feature extraction and regression?
- **RQ4:** Which image classification model achieves a higher accuracy than other state-of-the-art out-of-the-box models detecting nutrient deficiency symptoms?
- **RQ5:** Does background segmentation of cultivar images improve the accuracy of image classification models detecting nutrient deficiency symptoms?

I-B Outline

The structure of the remainder of the thesis is as follows: Chapter II gives background information to understand the problem setting, Chapter III describes important related work, and Chapter IV gives a detailed overview of the methodology. Chapter V details the results. Chapter VI addresses the discussion of our research questions, the limitations of our research and introduces ideas for future work. Finally, Chapter VII draws conclusions.

II Background

II-A Growy

Growy is a vertical farming company that produces a large selection of cultivars consisting of herbs, salads, and microgreens. Cultivars are grown in four phases consisting of germination, pre-growth, growth and pre-harvest. Phases differentiate between optimal environments to optimise the life cycle of cultivars. Cultivars are grown in soilless gutters that contain growfoam as substrate. Growy uses an active hydroponic system called nutrient film technique. The system circulates nutrient solution through the gutters allowing the plant roots to directly absorb nutrients and reuse water from the reservoir. The gutters are placed on a small slope which allows nutrient solution to automatically flow through the gutters. Growy uses a specific nutrient mix which is being refined to optimize growth. The pH, electrical conductivity and temperature of the nutrient solution are consistently measured multiple times per day.

Figure 1a showcases Growy farm operations. All products are grown in cells containing multiple vertical layers on which cultivars grow in gutters. Carbon dioxide is blown through the cell to facilitate photosynthesis. Lighting is controlled by white, blue and red LEDs. In Figure 1b, a sample image of basil is captured in white LED light. Visually, green algae grow on the growfoam substrate which blends in with green cultivars.

(a) (b)

Fig. 1: (**a**) Farm operations in a cell showing vertically grown cultivars under a combination of red and blue LED light at Growy (photo: Ramon van Flymen) and (**b**) a top view image taken of basil growing in a gutter with green algae growing on the growfoam substrate.

Growy sells to supermarkets and restaurants, so the product cannot contain discolored or wilted plants. At the moment, a substantial amount of product cannot be sold due to poor quality. Growy wants to prevent discoloration and wilting through custom plant profiles, nutrient mixes or environment variables. To address this problem, stress detection and

analysis are needed. In Figure 2, the workflow for stress detection and analysis is shown. Automatic stress symptom detection aims to help biologists fully focus on analysis of stressed cultivars and omitting images of healthy cultivars. An added benefit is that stress symptoms may be detected before biologists observe stress with the naked eye.

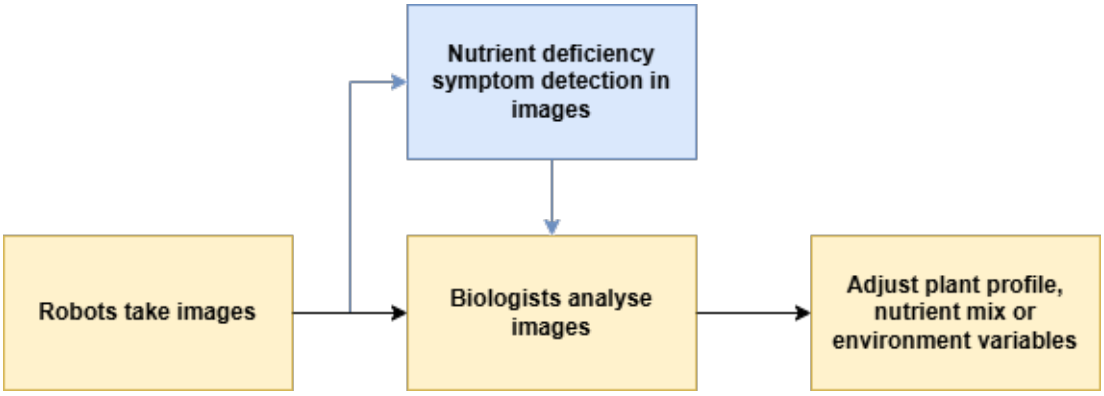


Fig. 2: Workflow for stress detection and analysis. The original workflow is described in yellow and an addition to the workflow is described in blue. Images are captured and biologists analyse them to adjust the plant profiles or environmental conditions.

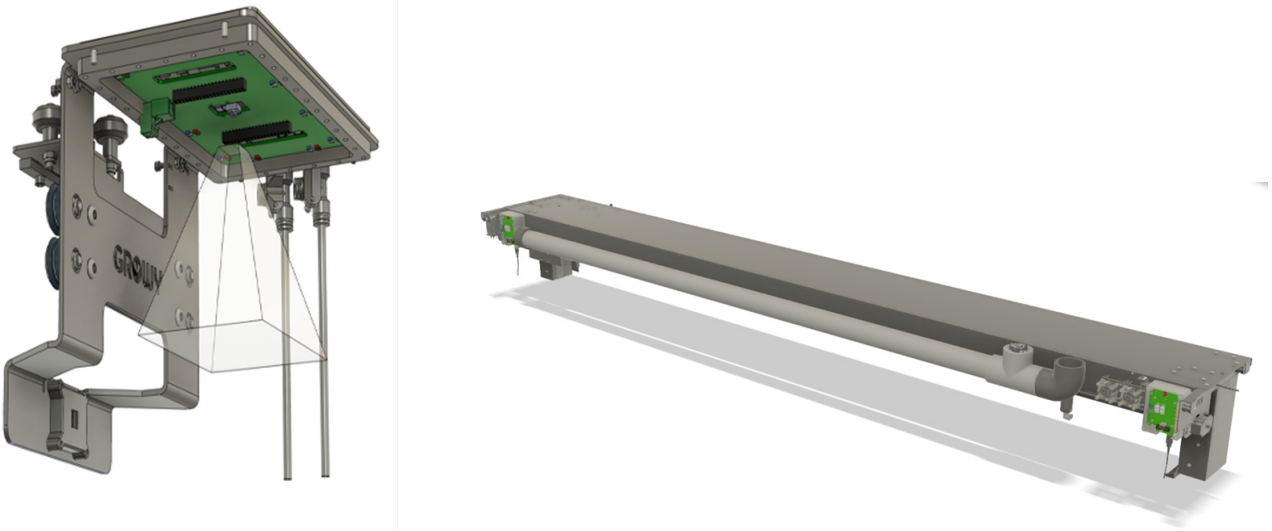
II-B Hardware

Remote-controlled farm robots are deployed to routinely water and capture images of cultivars. The images were taken with a camera board containing a Raspberry Pi Zero 2 W and Raspberry Pi Camera Module 3 NoIR (using the *picamera2* library) illustrated in Figure 3a. The camera boards are mounted on layer robots that receive commands from a Raspberry Pi 4 shown in Figure 3b. The farm robots experience occasional difficulties such as movement during image capture, loss of camera focus, synchronization of robot commands and improper lighting settings.

II-C Datastructures

A pipeline has been developed to store captured images in the cloud on Amazon Web Services. AWS Simple Storage Service (S3), Sagemaker AI and Athena make up an integral part of this Extract, Transform, Load pipeline. Cloud storage provides a landing area for images to be accessed. Sagemaker AI provides access to notebooks with CPU and GPU compute for model training and access to labeling jobs. Athena retrieves structured data from the database based on SQL queries.

PlantManager is a platform which allows users to plan, visualise and manage the processes happening in the farm. The key functionalities of PlantManager are split into four catagories:



(a) (b)

Fig. 3: (**a**) Camera board containing a Raspberry Pi Zero 2 W and Raspberry Pi Camera Module 3 NoIR to capture images below and (**b**) a layer robot that moves horizontally across a layer. It is equipped with a mounted camera board and a watering system. Images taken from Growy.

- Planning: Organises scheduling of growth cycles.
- Lifecycle: Visualises what gutters are being seeded and harvested per day. Also, displays information about growth cycle IDs such as plant profile, location history and harvesting date.
- Manage: Consists of creating new plant profiles, managing layer robots, viewing details of inventory and displaying incidents of the past 24 hours.
- Grafana: Displays exploratory data analysis (EDA) dashboards.

Grafana consists of multiple dashboards which display company metrics guiding decision-making processes. For instance, the air quality and water system are monitored using various sensors which are displayed with bar charts and graphs. Various camera and plant metadata is accessed through the *CameraMetaData* and *PlantMetaData* classes in Sagemaker AI notebooks provided by Growy.

III Related Work

It is essential to understand the current state of research surrounding early stress detection. Plant stress detection includes research in many domains such as nutrient deficiency detection, depth estimation, image segmentation and image classification. The section Related Work provides an overview of prior research on these topics and highlights techniques that help implement a stress detection pipeline.

III-A Nutrient Deficiency Detection

Nutrient deficiencies in plants can manifest visually and can be detected by the naked eye. Yellowing or browning of leaves, stunted growth and abnormal leaf patterns are all indicative signs of nutrient deficiency [4]. Nevertheless, visual inspection can be deceptive or symptoms can be misinterpreted. Relying solely on visual inspection may result in misdiagnosis, because symptoms of nutrient deficiency may resemble symptoms caused by pests, diseases or environmental stress. Also, Growy produces differently colored cultivars which may express varying stress symptoms and hinder nutrient deficiency detection. When nutrient deficiency is not yet visible to the naked eye, stress already affects plant growth [5]. This makes visual inspection a sub-optimal approach.

Soil testing is a reliable method to detect nutrient deficiencies. Soil acidity and alkalinity affect nutrient uptake due to variance in pH [6]. Soil salinity also impacts the rate at which nutrients are absorbed by plants which is measured using electrical conductivity (EC). Growy uses growfoam substrate instead of soil which cannot be evaluated using traditional soil pH meters or EC probes. However, the nutrient solution added to the gutters can be measured in terms of pH and EC.

Nutrient deficiency can also be detected using laboratory testing. This method can confirm or deny the intuition that a plant is nutrient deficient from plant and soil analysis [7]. Although laboratory testing is a thorough method, our problem requires both quick and accurate nutrient deficiency detection.

Textural feature analysis methods such as local binary patterns (LBP) [8] and grey level co-occurence matrices (GLCM) [9] analyse the spatial relationship of pixel values between neighbouring pixels. LBP and GLCM calculate various textural features of leaves which are predictors for nutrient deficiency. Waghmare et al. [10] uses fractal-based features for disease detection and classification in grape leaves and achieve an accuracy of 96.6%. Fractal-based features are locally invariant in nature and therefore patterns of diseases can be distinguished. In another paper by Sabri et al. [11] magnesium, nitrogen and potassium deficiencies are classified using GLCM, hu-histogram and color histogram as parameters. The random forest classifier achieved an accuracy of 78.35%. Sulastri et al. [12] performs

feature extraction with a GLCM to predict nutrient deficiency using the Learning Vector Quantization (LVQ) method. This method assigns data points called prototypes to represent a class and move them during training to better represent the class. After training, a data point is assigned the class of the closest prototype. LVQ achieved an accuracy of 87.5% for nutrient deficiency detection of nitrogen, phosphorus and potassium.

III-B Depth Estimation

Metric and relative depth estimation techniques perform a per-pixel regression task to estimate the absolute or relative depth of objects in images. State-of-the-art depth estimation models have been trained on large datasets of depth maps. Time-of-Flight and Light Detection and Ranging (LiDAR) are two popular depth sensing technologies to create depth maps.

MiDaS [13], [14] was one of the first robust monocular depth estimation models. MiDaS v3.1 uses vision transformers as image encoders to improve its quality and runtime. Bhat et al. [15] created ZoeDepth which has excellent generalization capabilities while maintaining metric scale, combining both metric and relative depth estimation. Marigold by Ke et al. [16] uses a diffusion model to perform depth estimation. Marigold relies on the rich latent space of diffusion models to gain knowledge about the intrinsic structure of images. Bochkovskii et al. [17] created Depth Pro which outputs high frequency details and high resolution results compared to earlier models.

III-C Image Segmentation

Traditional image segmentation consists of region- and edge-based segmentation methods. Region-based methods include thresholding [18] and split-and-merge [19]–[21]. Thresholding splits images into foreground and background based on a threshold value. Local thresholding utilizes a tile-based approach to find thresholds locally. Global thresholding finds one global threshold for the whole image. Split-and-merge splits images into smaller regions based on a homogeneity criterion and similarly homogeneous regions are merged to create the segmented result.

Edge-based methods include the Canny edge detector [22], Sobel edge detector [23] and Marr-Hildreth edge detector [24]. Edges are places in an image where the intensity rapidly changes. Edge detectors use first derivatives or second derivatives of abrupt changes in pixel intensity to detect edges. First derivatives measure the rate of change of pixel intensities throughout an image. Second derivatives measure the rate of change of the first derivative which often indicates an edge.

Modern image segmentation consists of neural network-based segmentation methods. Vision transformers (ViT) [25] are large neural network-based models which specialize in object detection, segmentation, classification or pose estimation, or all of them. For instance,

the Segment Anything Model (SAM) and SAM 2 [26], [27] use vision transformers as image encoders. In the context of image segmentation, Kirilllov et al. perform object segmentation in images using a text prompt to decide which objects to segment. SAM 2 is trained on both image and video data and introduces real-time video processing. Grounded-SAM [28] uses GroundingDINO [29] and SAM to combine object segmentation with bounding box regression. GroudingDINO predicts bounding boxes for objects using one or multiple text prompts. Non-max suppression [30] is utilized to remove overlapping, redundant bounding boxes in Grounded-SAM. FastSAM [31] aims to solve the huge computation costs of SAM. FastSAM achieves comparable results to SAM at 50 times higher run-time speed.

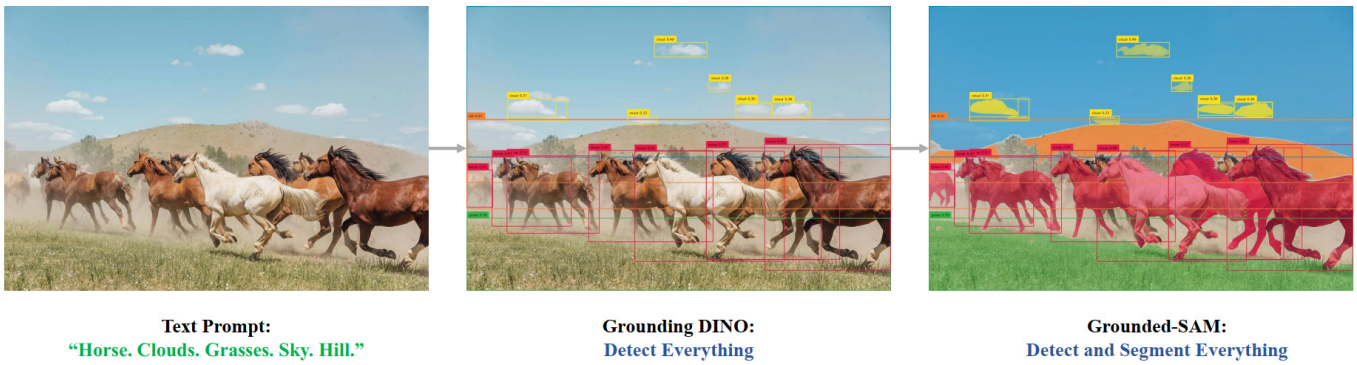


Fig. 4: Overview of Grounded-SAM. The input "Horse. Clouds. Grasses. Sky. Hill." instructs GroundingDINO to find instances of all mentioned objects and draw bounding boxes around them. SAM generates segmentation masks by segmenting the objects within bounding boxes. Image taken from Ren et al. [28]

III-D Image Classification

Convolutional neural networks (CNNs) [32] perform state-of-the-art image classification in a variety of tasks. CNNs use convolutional layers to perform convolution operations on an image and output a feature map. Convolution operations slide a window, called a kernel or filter, across the image. The dot product between the values of the kernel and the input of each position in the image is computed and stored in a fixed-size feature map. The outputted feature map contains detected features of the image. Detected features consist of areas of interest such as edges, textures or shapes. CNNs are mostly useful for image classification, because spatial information is lost after the dense layers of the network. As opposed to CNNs, fully convolutional networks (FCNs) [33] preserve spatial information of the input and produce per-pixel predictions which is ideal for image segmentation tasks. Image classification models predict the class of a whole image. Models are trained using large datasets of labeled images such as the MNIST [34], CIFAR-10/100 [35] and ImageNet [36] datasets. Transfer learning is a technique where a pre-trained model which has already learned general features is further trained using task-specific data. Consequently, a model does not have to be trained from scratch anymore.

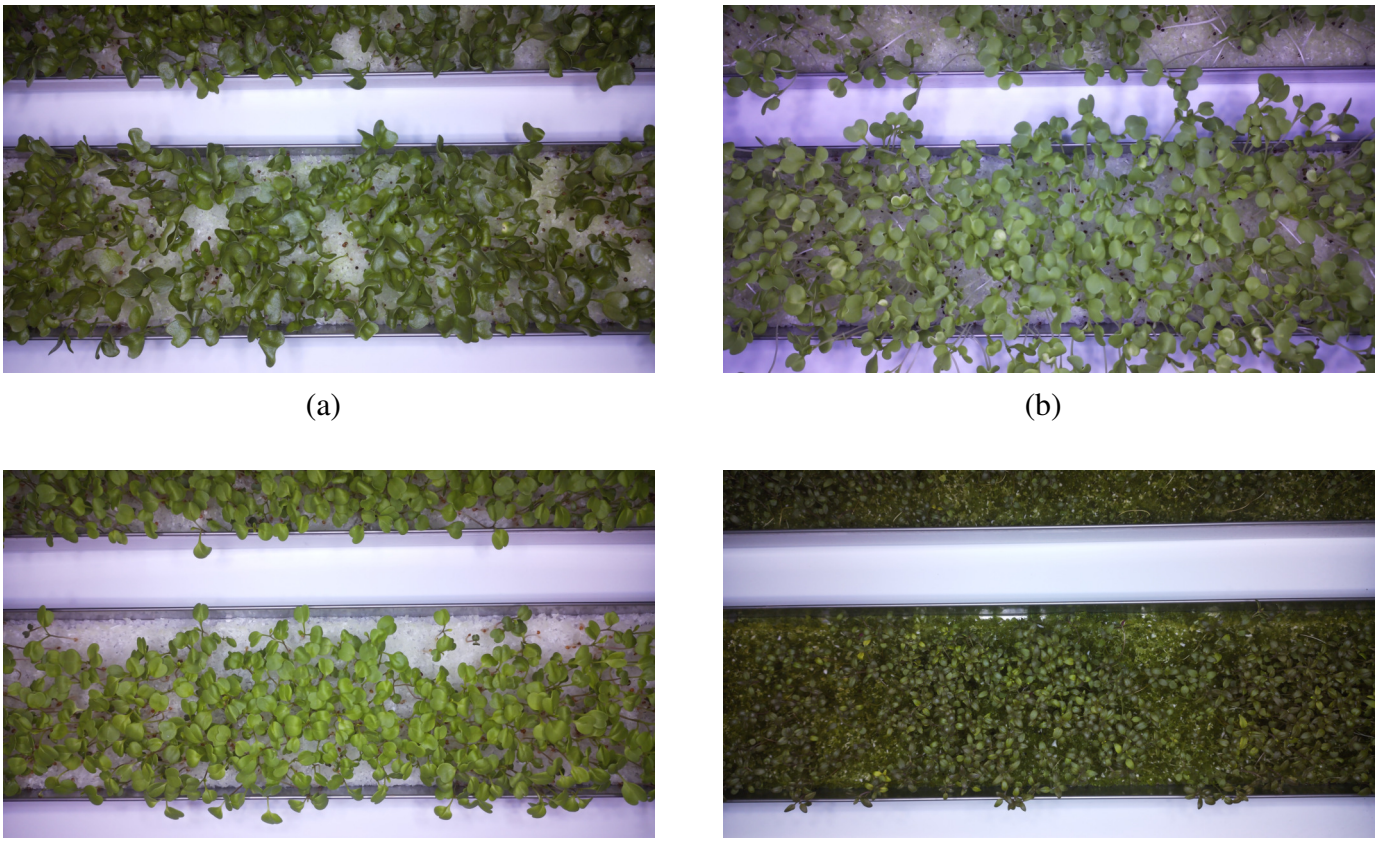
Several models that started the rise of deep learning are called AlexNet [37], GoogLeNet [38] and VGG16 [39]. AlexNet adopted ReLU activations and large fully connected layers which resulted in a large amount of parameters. AlexNet reduced the top-5 error in the ImageNet dataset from $\tilde{2}6\%$ to 15%. GoogLeNet adopted the Inception module, a combined block of parallel convolutions and max pooling, and was much more computationally effective than other state-of-the-art models. Global average pooling was used instead of the standard fully connected layers. VGG16 showed that deeper layered models outperformed shallow models. VGG16 had 16 layers and a massive amount of parameters which makes training the model from scratch computationally heavy. However, VGG16 is commonly used in transfer learning tasks which require only small-scale training.

IV METHODOLOGY

IV-A Dataset Collection

Farm robots collect images of four cultivars: dwarf blue kale (*brassica oleracea*), paksoi (*brassica rapa* var. *chinensis*), salad rocket (*eruca sativa*) and thyme (*thymus vulgaris*). Biologists at Growy selected cultivars based on their plant profiles and their weakness to nutrient deficiency. Plant profiles are defined as the number of days needed in each phase to complete a lifecycle. For the selected four cultivars shown in Figure 5, plant profiles are defined as:

- Dwarf Blue Kale - 3x germination, 5x pre-growth, 10x growth, 2x pre-harvest
- Paksoi - 4x germination, 4x pre-growth, 10x growth, 2x pre-harvest
- Salad Rocket - 3x germination, 3x pre-growth, 10x growth, 2x pre-harvest
- Thyme - 5x germination, 8x pre-growth, 10x growth, 2x pre-harvest



(a) (b) (c) (d)

Fig. 5: Selected cultivars: (**a**) dwarf blue kale, (**b**) paksoi, (**c**) salad rocket and (**d**) thyme.

An experiment layer is reserved to conduct photoruns of gutters. Photoruns consist of five images (4608x2592 pixels) per gutter for every day the cultivars remain in the growth layer. Images are taken over a timeframe of ten days to examine stressed cultivars for a

prolonged period. Photoruns are executed at night using LEDs for consistent lighting. The gutters are divided into a test group and a control group. For each cultivar, one batch, which is equal to six gutters, is manually seeded per group. The control group receives the standard nutrient solution mix of Growy throughout the lifecycle. Instead of the nutrient solution mix, the test group receives pH-balanced water in the growth phase, lowering the nutrient uptake of cultivars. The EC of pH-balanced water fluctuates between 0.5 and 0.7 dS/m and the EC of Growy's nutrient solution mix fluctuates between 1.8 and 2.0 dS/m. Optimal EC levels are crucial for efficient nutrient uptake which means giving cultivars pH-balanced water causes nutrient deficiency stress. Appendix A showcases test group images of cultivars visually changing over time.

Biologists at Growy manually analysed test group images to indicate the day of stress initiation in cultivars. Salad rocket and dwarf blue kale are visibly stressed from day 5 although the difference between groups is minimal. Paksoi is not visibly stressed after 10 days of suboptimal nutrient uptake. Unfortunately, a robot error caused multiple photoruns of thyme to fail and therefore the decision has been made to exclude thyme from further experiments.

IV-B Data Cleaning

After visual inspection, unwanted cultivar images are removed to improve data quality. Images are not usable for pixel-level feature extraction or model training for one of four reasons:

- Movement blur,
- Loss of camera focus,
- Improper lighting,
- Empty gutter image due to imbalanced distribution of seeds across gutter.

In Table I, the number of total images before and after data cleaning and the number of cleaned images as a percentage of the total number of images are displayed. For pixel-level feature extraction, the duplicate and incomplete photoruns are also excluded as these photoruns skew the mean feature values across gutters. However, these photoruns are useful training data for model training.

TABLE I: Overview of image counts before and after data cleaning for each cultivar. The table includes the total number of images, the number remaining after data cleaning and the proportion of cleaned images relative to the total dataset.

	Dwarf Blue Kale	Paksoi	Salad Rocket
Total images	774	810	797
Images after cleaning	725	774	751
Cleaning as % of total	6.33%	4.44%	5.77%

IV-C Image Preprocessing

Depth estimation, thresholding, and model training are combined to segment the background of cultivar images. In many cases, the original images contain green algae growing on the growfoam. Green algae could interfere with the accurate detection and classification of cultivars. To address this, Depth Anything v2 [40] is used to estimate the relative depth of objects in the images. This model combines DINOv2 [41] as the encoder for feature extraction and DPT [42] as the decoder for depth regression. The pre-trained model from Hugging Face is used to generate depth maps from the original RGB images. To separate the algae from the background, Otsu's method [43] is applied to the depth maps. This method automatically determines the optimal global threshold by minimizing the weighted intra-class variance between foreground and background regions. Thresholding on the optimal global threshold extracts the foreground from the image. Combining relative depth estimation with Otsu's method produces promising results in background segmentation. However, some algae and metal edges still appear in the segmentation masks. To remove the artifacts, a U-Net [44] is trained using a small manually annotated dataset. A total of 18 ground truth binary masks are created by cleaning the segmentation masks from Depth Anything V2 and Otsu's method. The algae and metal are removed from the binary masks using the GIMP image editor [45] to then serve as training data. This allows the model to learn how to filter out unwanted artifacts and improve the overall segmentation quality.

A U-Net is an u-shaped architecture which employs a fully convolutional network for image segmentation. The network consists of an encoder and decoder which contain contracting and expanding paths of convolutional layers. The encoder captures the context using downsampling and the decoder upsamples the features back to the original size. In Figure 6, the U-Net architecture is displayed. The U-Net was trained on resized images of 896x512 pixels and their respective binary masks. The input images are rectangular to maintain the aspect ratio of the original images. The model ran for 100 epochs with a batch size of 6 as a result of the small annotated dataset. The U-Net contains a sigmoid activation and binary crossentropy is used as loss for pixel-level accuracy.

Data augmentation is utilised to increase the size of the dataset available for model training. The implemented data augmentation methods include vertical flip, horizontal flip, translation, rotation, crop and resize, and elastic deformation. For elastic deformation, alpha is 34 and sigma is 4. The values are chosen based on the original U-Net implementation. Every image is turned into six images by data augmentation consisting of the original image and five augmented images. The augmentations all have a 50% chance of occurring per image, allowing for multiple augmentations per image.

IV-D Feature Extraction

A grey level co-occurence matrix (GLCM) [9] is a method which extracts textural features from images. The matrix is defined as the distribution of co-occuring greyscale pixel values

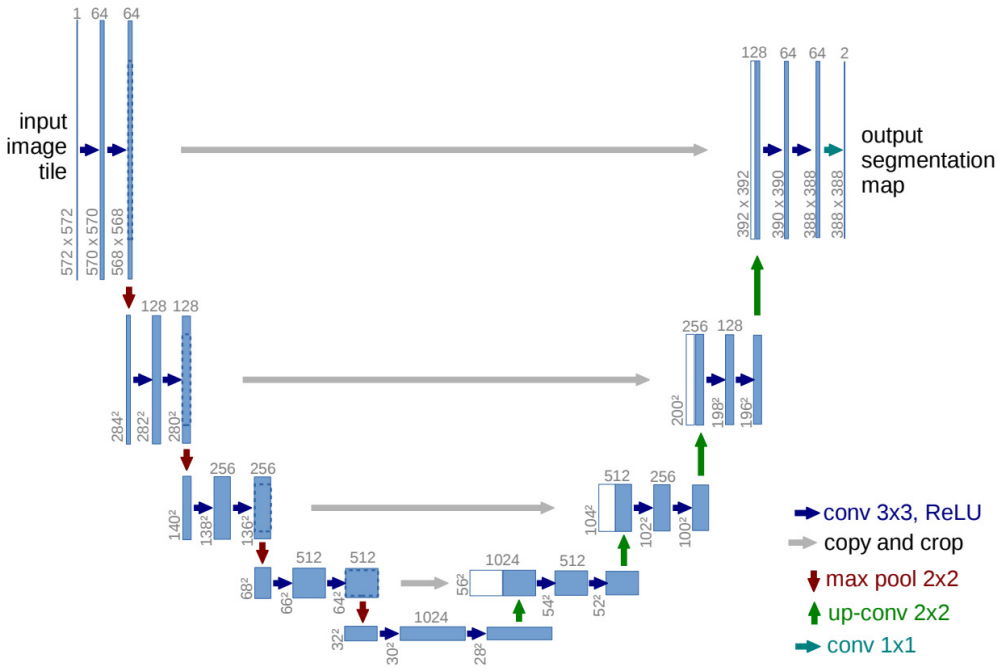


Fig. 6: U-Net architecture. Each blue box represents a multi-channel feature map, with the number of channels indicated above the box. The spatial dimensions are shown at the bottom left corner. White boxes indicate feature maps that have been copied. Arrows illustrate the different operations. Image taken from Ronneberger et al. [44]

at a given offset and angle. The given offset used is 1 pixel, because spatial relationships between directly neighbouring pixels are most important. Because our textural analysis does not care about rotational variance, the angles consist of regular angles i.e. 0, 45, 90, and 135 degrees. GLCM only has to calculate a matrix for angles 0, 45 90 and 135 because the matrices of directly opposite angles can be obtained by transposing the matrices of angles 0, 45, 90 and 135. Various textural characteristics can be calculated from the GLCM matrix. GLCM calculates textural characteristics per patch in an image. To determine a fitting patch size, three patch sizes are compared in their results. The compared patch sizes are 32x32, 64x64 and 128x128 pixels. A small patch size detects small-scale texture details and a large patch size detects coarse textures in images. Additionally, the top gutter in each image was removed by cropping 600 pixels from the top edge of the background-removed images.

The matrix is normalized to get the probability of finding neighboring pixel values instead of the number of instances. Equation 1 calculates the normalized matrix by dividing each value V by the sum of the values in the matrix.

$$P(i,j) = \frac{V(i,j)}{\sum_{i,j=0}^{N-1} V(i,j)} \tag{1}$$

where

- i is the row number for the reference pixel value and j is the column number for the neighbour pixel value,
- $V(i,j)$ is the number of co-occurrences for pixel values (i,j),
- $P(i,j)$ is the probability of co-occurrences for pixel values (i,j),
- N is the specified number of pixel values.

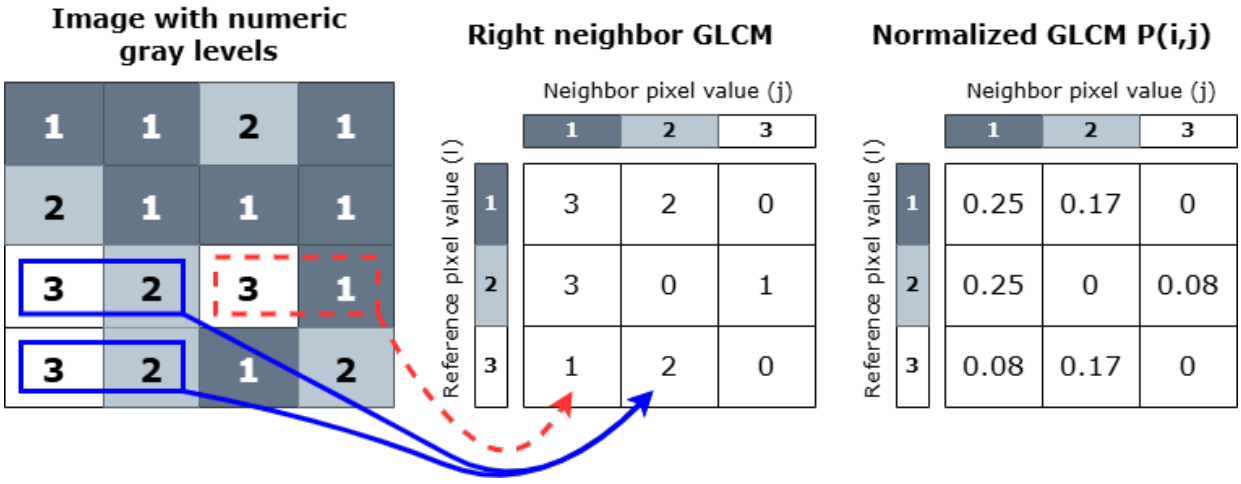


Fig. 7: A right-neighbor GLCM examines the neighbor pixel value j immediately to the right of each reference pixel i and counts how often each pair of gray levels occurs. In this example, (3,1) is counted once and (3,2) is counted twice as the arrows point out. These counts are then normalized to represent the probability $P(i,j)$ of a specific neighboring pixel value occurring next to a given reference pixel.

The characteristics that have been proven to indicate nutrient deficiency symptoms in cucumber seedlings are top projected canopy area (TPCA), entropy, energy, homogeneity and contrast [46], [47]. With the exception of TPCA, the characteristics are related to the contrast and orderliness of the images. The orderliness of an image indicates the presence of structure and harmony or a lack thereof. A detailed explanation of each characteristic is provided below:

1) TPCA

TPCA indicates leaf area. A higher TPCA value indicates a more pronounced growth pattern. In each image mask, white pixels represent 1 and black pixels represent 0. By summing the values of the white pixels across the image, we can determine the number of foreground pixels, which corresponds to the total leaf area. The TPCA feature of an image is defined in Equation 2.

$$\text{TPCA} = \sum_{i,j} I(i,j) \tag{2}$$

2) Entropy

Entropy indicates texture and information complexity. Higher entropy indicates higher texture or information complexity. The Entropy feature of an image is defined in Equation 3.

$$\text{Entropy} = \sum_{i,j=0}^{N-1} -ln(P(i,j)) * P(i,j) \tag{3}$$

3) Energy

Energy (or Angular Second Moment) represents the orderliness or homogeneity of the image. High energy values indicate more uniform texture. The Energy feature of an image is defined in Equation 4.

$$\text{Energy} = \sum_{i,j=0}^{N-1} P(i,j)^2 \tag{4}$$

4) Homogeneity

Homogeneity reflects the closeness of the distribution of elements in the GLCM to the GLCM diagonal. High homogeneity values indicate that elements are concentrated along the diagonal, suggesting a more uniform texture. The Homogeneity feature of an image is defined in Equation 5.

$$\text{Homogeneity} = \sum_{i,j=0}^{N-1} \frac{P(i,j)}{1+(i-j)^2} \tag{5}$$

5) Contrast

Contrast returns a measure of the intensity contrast between a pixel and its neighbor over the whole image. The Contrast feature of an image is defined in Equation 6.

$$\text{Contrast} = \sum_{i,j=0}^{N-1} P(i,j) * (i-j)^2 \tag{6}$$

For each image, feature values are computed by averaging the values across all patches. Then, the mean feature values of each gutter are computed by averaging the values over all gutter images of a singular gutter. Last, the mean feature value of the entire batch is computed by averaging the values over all gutters. Incomplete sets of gutter images due to data cleaning are excluded in the feature extraction results to prevent skewing the mean feature values of singular gutters. Naturally, this creates an imbalance in the data. Using the mean feature values of each batch per day, the progression of feature values in the test and control group can be displayed. Ideally, the feature values of the test and control group diverge due to the onset of nutrient deficiency stress.

IV-E Piecewise-segmented Regression Analysis

Piecewise-segmented regression [48] is used to identify the onset of stress caused by nutrient deficiency affecting cultivar growth. The algorithm iteratively tests every possible combination of two intervals across the data. For each combination of intervals, the sum of squared errors (SSE) is calculated for each of the line segments that are fitted on the data points in the intervals. The meeting point between the two line segments with the lowest combined error constitutes the change point t_c of stress initiation. Before calculating change point t_c, the difference between mean textural features of the test group and control group are calculated. Equation 7 shows how Z is calculated. F_t and F_c are the mean values of the test and control group features at day t_i of the experiment. Equations 8 and 9 represent two line segments which follow the linear formula $y = ax + b$ where α and β are the slope and intercept parameters respectively and t represents independent variable x. These equations show how the change point t_c and predicted dependent variable Z_{break} are calculated.

$$Z(t_i) = F_t(t_i) - F_c(t_i) \tag{7}$$

$$Z_{break} = \alpha_0 t + \beta_0, \text{when } t < t_c \tag{8}$$

$$Z_{break} = \alpha_1 t + \beta_1, \text{when } t > t_c \tag{9}$$

To ensure a fair comparison, data points are plotted based on the exact timestamp each image was captured, rather than by day, since photoruns are occasionally delayed due to robot errors. The dataset is divided into two classes, nutrient-deficient and healthy, using the change point t_c calculated from piecewise-segmented regression. The need for manual labeling is removed and human labeling errors are prevented.

IV-F Image Classification Models

Image classification models are deep convolutional neural networks that are trained on datasets of labeled images. This training process uses backpropagation to adjust the model weights and minimise the loss based on a loss function. Gradients tell the model how to adjust its weights to reduce the error during training. Deep networks may contain gradients that become extremely small or large during training which is called the vanishing gradient problem. This makes it difficult for earlier layers to learn useful feature representations. ResNet50 [49] is a deep residual network with 50 layers that addresses this problem. Skip connections reduce the impact of vanishing gradients and enable the successful training of deeper models for image classification.

DenseNet-121 [50] is a convolutional neural network that uses dense connections between layers to improve feature learning. In DenseNet, each layer receives the feature maps of all preceding layers as additional input. This structure helps preserve information throughout

the network and encourages feature reuse and leads to more efficient learning. Consequently, DenseNet creates detailed representations that are useful for identifying specific patterns in images. This makes the model especially effective at capturing small details in image classification tasks.

EfficientNet-B0 [51] is a family of models designed to balance accuracy and efficiency. It was developed using neural architecture search, an automated method that explores different network designs to find the best one for a given task. EfficientNet balances depth, width and resolution of the network to create a strong model with fewer parameters and lower computational cost. Compared to many other image classification models, EfficientNet is relatively small while still maintaining high accuracy.

The models are used as feature extractors and trained from scratch. The network is implemented without the original classification layers. Then, a global average pooling layer is applied to the output of the models to receive a 1-dimensional feature vector. Next, a fully connected Dense layer with ReLU activation is added to learn high level features. A Dropout layer is added to reduce overfitting by resetting 50% of the neurons during training. The final Dense layer uses a softmax activation to output a probability distribution over the two classes, nutrient-deficient and healthy. The input size of all three models is 224x224 pixels. The images are cropped using a sliding window. The sliding window moves across the image and calculates the number of green pixels per window. The window with the largest number of green pixels is cropped, resulting in images containing many leaves.

IV-G Training Configuration

The models are implemented and executed on a ml.g4dn.xlarge instance in a AWS Jupyter Lab notebook. The specifications of the instance include:

- Compute: 4 virtual CPUs,
- Memory: 16 GiB of RAM,
- GPU: Nvidia T4 GPU,
- Processor: Intel Xeon Family,
- Clock Speed: 2.5 GHz.

The pre-defined models are loaded from the Keras Applications library inside the TensorFlow pip package [52]. The datasets of all cultivars are divided into training, validation and test sets. For each cultivar, the training set is 72.25% of the dataset, the validation set is 12.75% and the test set is 15%. The models run for 50 epochs with a batch size of 16. The models uses the Adam optimizer with a learning rate of 0.0001. The loss function is categorical crossentropy, because the two target labels are not binary but instead categorical classes.

IV-H Evaluation Metrics

To evaluate the image segmentation and classification models, several evaluation metrics are introduced. True and false positives and negatives are used to calculate precision, recall, F1-score and accuracy. Precision is the proportion of all positive classifications that were actually positive and is defined in Equation 10. Recall, also called true positive rate (TPR), is the proportion of all true positives that were correctly classified as positive and is defined in Equation 11. F1-score is the harmonic mean of precision and recall and is defined in Equation 12. Accuracy is the proportion of classifications that were correctly classified as either positive or negative and is defined in Equation 13. In addition to these metrics, intersection over union (IoU) is calculated to further quantify the image segmentation techniques and is defined in Equation 14. A represents the ground truth binary mask and B represents the predicted segmentation mask.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{10}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{11}$$

$$\text{F1-score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{12}$$

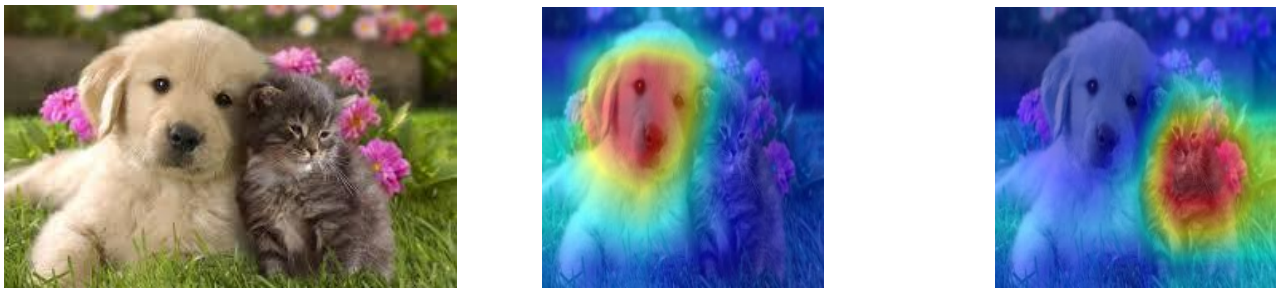
$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{13}$$

$$\text{IoU} = \frac{\text{A} \cap \text{B}}{\text{A} \cup \text{B}} \tag{14}$$

Two visualisation methods to better understand model performance are the AUC-ROC curve and the confusion matrix. The ROC curve plots the true positive rate against the false positive rate at different classification thresholds. The AUC value is the area under the ROC-curve which indicates overall performance of the classifier and its degree of separability between the classes. The confusion matrix is a table which summarizes the performance of a classification model by showing the true positive, true negative, false positive and false negative predictions.

GRAD-CAM [53] is a gradient-based localization method to visualise class activations in image classification models. To better understand the differences between classifying original images and background-removed images, GRAD-CAM generates a heatmap to highlight influential regions of the image for the class prediction of a model. In Figure

8, GRAD-CAM is utilised to visualise the class activations of a trained ResNet50 model classifying the original image in Figure 8a as the class 'dog' in 8b and as the class 'cat' in Figure 8c.



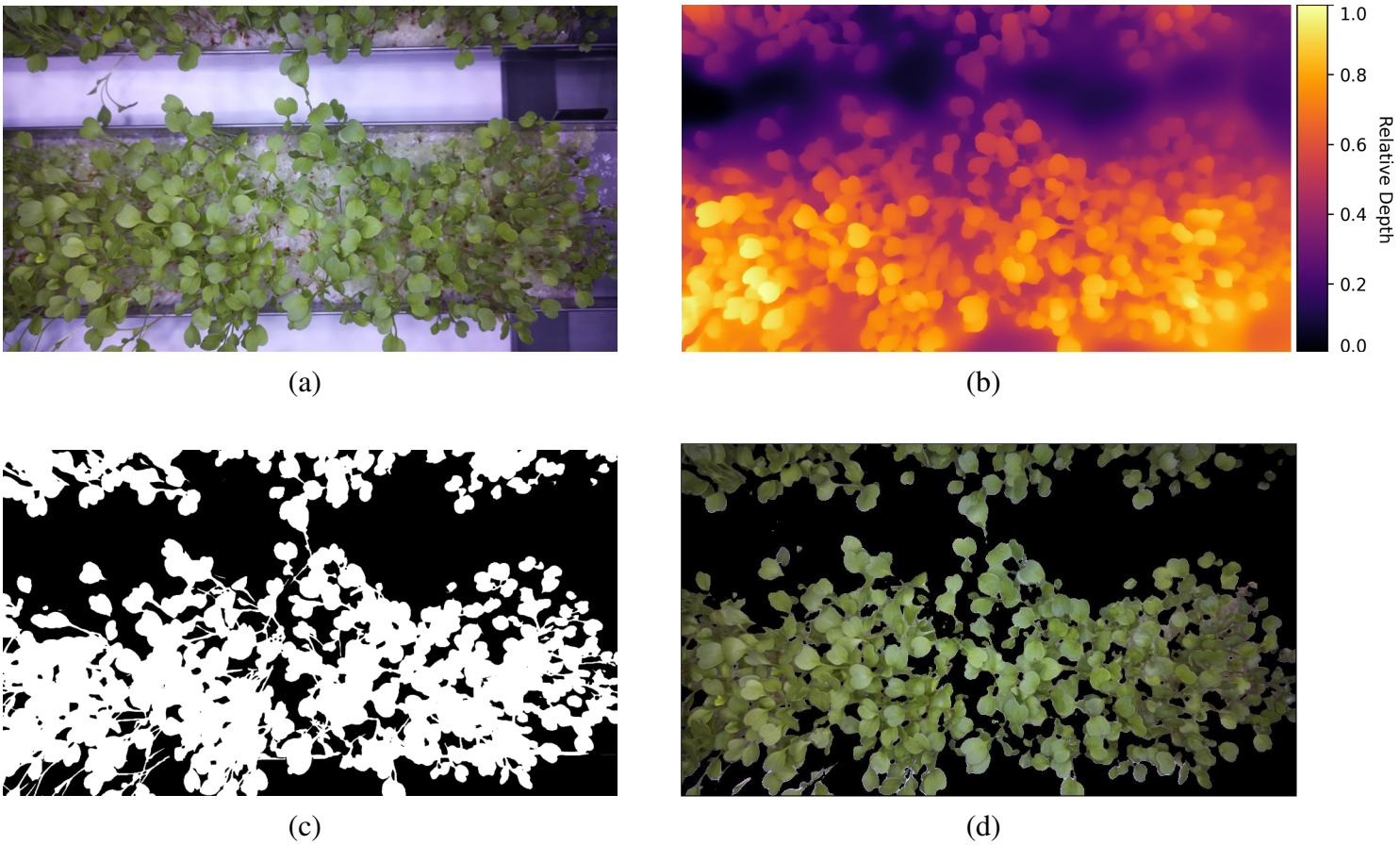
(a) Original image of dog and cat (b) Class 'dog' (c) Class 'cat'

Fig. 8: GRAD-CAM heatmap visualisations of a trained ResNet50 model classifying the (**a**) original image as (**b**) the class 'dog' and (**c**) the class 'cat'. Images taken from pytorch GRAD-CAM implementation [54].

V RESULTS

V-A Background Segmentation

The background segmentation method is illustrated in Figure 9 containing all four steps in the process. Figure 9a contains the original image. In Figure 9b, the relative depth map of the original image is displayed using Depth Anything V2. Next, Figure 9c shows the binary segmentation mask after thresholding the relative depth map using Otsu's method. Only the foreground including metal edges remains with the horizontal metal edges located at the bottom of the image. Finally, Figure 9d depicts the segmentation output obtained by training the U-Net model on the training images and their respective binary segmentation masks and predicting on previously unseen images.



(a) (b)

(c) (d)

Fig. 9: Background segmentation example: (**a**) original input image, (**b**) relative depth map generated with Depth Anything V2, (**c**) foreground extraction using Otsu's thresholding, and (**d**) segmentation output from trained U-Net model.

Using relative depth estimation and Otsu's thresholding already returns promising results. However, green algae and metal edges remain in a substantial number of images and are incorrectly classified as foreground. This problem occurs because the metal edges are upright and measured at the same depth as the cultivars growing in the gutter. Additionally, green

algae are sometimes incorrectly included, because Otsu's method struggles to differentiate between the green hues of algae and cultivars. After visual inspection, the segmentation results indicate that the U-Net model removes metal edges and green algae more effectively and creates more accurate segmentation masks overall. In Table II, the trained U-Net model is evaluated using performance metrics. However, the performance metrics do not capture the full picture.

TABLE II: Performance metrics of background segmentation using trained U-Net model. The performance metrics are computed by comparing segmentation predictions to ground truth binary segmentation masks.

	Accuracy	Precision	Recall	F1-score	IoU
Trained U-Net model	0.856	0.783	0.992	0.875	0.778

To further investigate the performance of the trained U-Net model, the validation accuracy and loss of the trained U-Net model are visualised in Figure 10. The model has high training accuracy but poor validation loss suggesting a lack of generalisability. This may be the result of segmenting three different cultivars using a single U-Net model. Training just one model was deemed necessary, because manually cleaning binary masks using an image editor to obtain training data was time-consuming. Consequently, there was not enough training data to train three separate models. Table III contains the validation accuracy and loss across five folds of k-fold cross-validation of the U-Net model. The mean and standard deviation of the validation loss further suggest that the model is unstable and this was not an isolated run.

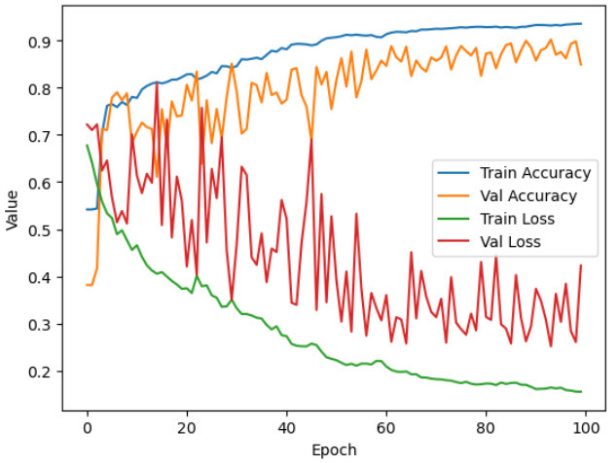


Fig. 10: Train accuracy, train loss, validation accuracy and validation loss of trained U-Net model.

TABLE III: K-fold cross-validation of trained U-Net model. Validation accuracy and loss are calculated for k=5 folds including the mean and standard deviation across all folds.

	Validation Accuracy	Validation Loss
Fold 1	0.8946	0.2857
Fold 2	0.9463	0.1422
Fold 3	0.9148	0.2206
Fold 4	0.8637	0.3701
Fold 5	0.8890	0.2508
Mean $\pm$ std	0.9017 $\pm$ 0.0276	0.2539 $\pm$ 0.0750

V-B Detection of Stress Initiation

The GLCM produced the most accurate results using a patch size of 32×32 pixels. Completely black patches were excluded to avoid skewing the results with informationless data. Larger patch sizes, such as 64×64 and 128×128 pixels, were less likely to consist entirely of black pixels. Since the method relies on calculating the mean across the entire patch, these larger patch sizes diluted the information. Therefore, a 32×32 pixel patch size was selected as the most effective.

In Figure 11, the control and test group mean feature values are plotted for dwarf blue kale, paksoi and salad rocket across ten days. The results show that the control and test groups are not distinguishable based on mean feature values. Control and test group values are initially expected to have similar values and diverge at the point when stress is introduced. The TPCA, entropy and homogeneity feature values exhibit no significant divergence, but instead are nearly identical across all cultivars. Although the energy feature values show some divergence in salad rocket and the contrast feature values show some divergence in paksoi, the other features do not indicate that stress was introduced.

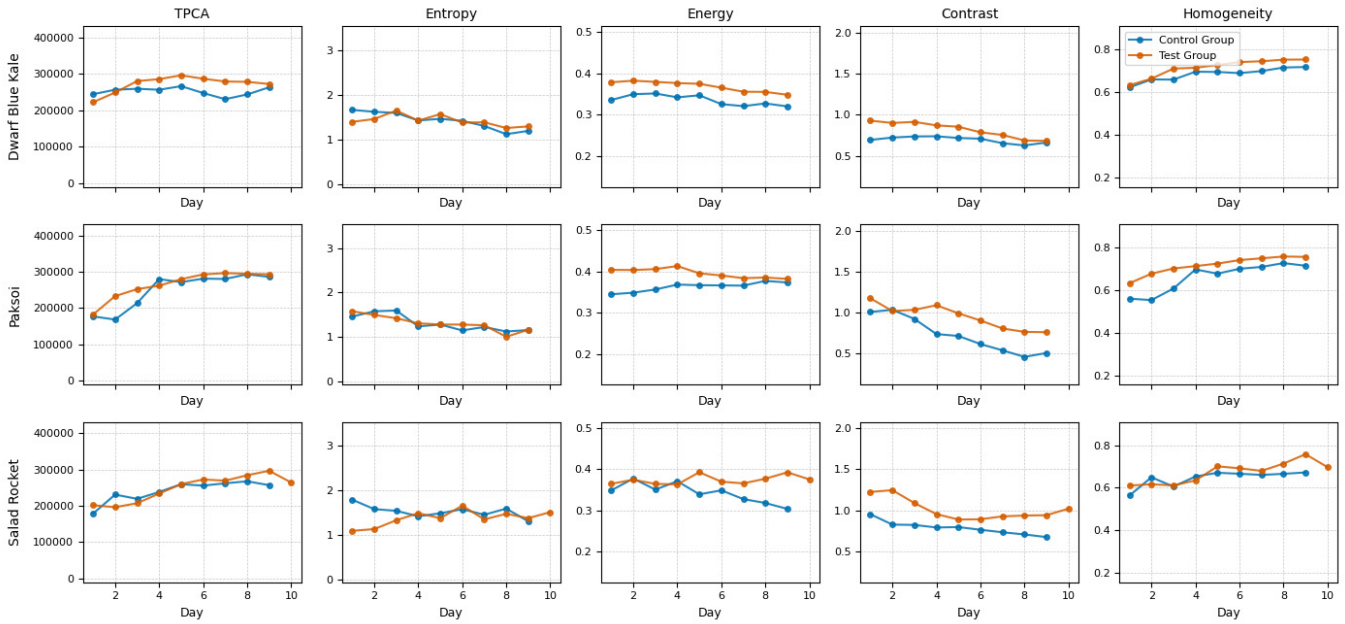


Fig. 11: Test group and control group mean feature values are plotted over a period of ten days. Mean values of TPCA, entropy, energy, contrast and homogeneity using GLCM patch size of 32x32 pixels.

Since no clear overall divergence in any of the features can be observed across cultivars, it is impossible to determine a change point t_c using piecewise-segmented regression. As a result, the dataset is not being split based on an observed point of stress initiation. Instead, the assessment conducted by biologists at Growy may serve as an indication for visible stress. Therefore, the test group is divided into days 1–5 and days 6–10 which represent

the healthy and nutrient-deficient classes respectively.

V-C Image Classification Models

Tables IV and V present the performance of ResNet50, EfficientNet-B0, and DenseNet-121 in binary image classification of cultivar images between healthy cultivars in days 1–5 and visibly stressed cultivars in days 6–10 across dwarf blue kale, paksoi and salad rocket. DenseNet-121 consistently achieved the highest F1-scores, with background-removed images having the best results overall. Interestingly, DenseNet-121 had the highest F1-score of 0.895 for salad rocket using original images. Also interesting to note that DenseNet-121 had the lowest F1-score of 0.632 for paksoi using original images. Dwarf blue kale was classified most accurately across all models and paksoi was the most challenging, especially for DenseNet-121. The standard deviations remain consistent within a margin between original and background-removed images. DenseNet-121 has the largest standard deviations while EfficientNet-B0 has the smallest standard deviations. The difference between the standard deviations of DenseNet-121 with dwarf blue kale and paksoi and DenseNet-121 with salad rocket is considerable.

Segmentation clearly improved model performance, especially for ResNet50. F1-scores increased by over 12% in the case of paksoi. EfficientNet-B0 and DenseNet-121 were less affected by background segmentation, but still slightly benefited from segmentation. This suggests that background information in the original images may have introduced noise that disproportionately affected the least complex model which is ResNet50. The performance difference between cultivars remained consistent across models. Dwarf blue kale was generally the easiest to classify, followed by salad rocket and paksoi. Appendices B and C contain additional ROC-AUC curves and confusion matrices for each cultivar and model combination trained on the background-removed images.

TABLE IV: Image classification of original images using ResNet50, EfficientNet-B0 and DenseNet-121 models for two classes: healthy cultivars between day 1–5 and visibly stressed cultivars between day 6–10. The mean and standard deviation of accuracy, precision, recall and F1-score are averaged across five runs.

Model	Cultivar	Accuracy	Precision	Recall	F1-score
ResNet50	Salad Rocket	0.796 ± 0.075	0.814 ± 0.068	0.796 ± 0.075	0.791 ± 0.080
	Dwarf Blue Kale	0.868 ± 0.048	0.867 ± 0.052	0.868 ± 0.048	0.866 ± 0.051
	Paksoi	0.744 ± 0.107	0.748 ± 0.109	0.744 ± 0.107	0.742 ± 0.107
EfficientNetB0	Salad Rocket	0.807 ± 0.017	0.813 ± 0.015	0.807 ± 0.017	0.805 ± 0.018
	Dwarf Blue Kale	0.875 ± 0.017	0.879 ± 0.017	0.875 ± 0.017	0.875 ± 0.017
	Paksoi	0.782 ± 0.042	0.784 ± 0.043	0.782 ± 0.042	0.782 ± 0.041
DenseNet121	Salad Rocket	**0.895 ± 0.015**	**0.901 ± 0.015**	**0.895 ± 0.015**	**0.895 ± 0.016**
	Dwarf Blue Kale	0.822 ± 0.150	0.824 ± 0.152	0.822 ± 0.150	0.796 ± 0.201
	Paksoi	0.700 ± 0.177	0.727 ± 0.217	0.700 ± 0.177	0.632 ± 0.235

TABLE V: Image classification of background-removed images using ResNet50, EfficientNet-B0 and DenseNet-121 models for two classes: healthy cultivars between day 1–5 and visibly stressed cultivars between day 6–10. The mean and standard deviation of accuracy, precision, recall and F1-score are averaged across five runs.

Model	Cultivar	Accuracy	Precision	Recall	F1-score
ResNet50	Salad Rocket	0.841 ± 0.053	0.810 ± 0.077	0.801 ± 0.090	0.788 ± 0.076
	Dwarf Blue Kale	0.900 ± 0.049	**0.890 ± 0.059**	**0.888 ± 0.062**	0.890 ± 0.044
	Paksoi	0.838 ± 0.050	0.795 ± 0.127	0.764 ± 0.142	0.778 ± 0.128
EfficientNetB0	Salad Rocket	0.825 ± 0.039	0.811 ± 0.041	0.810 ± 0.047	0.810 ± 0.041
	Dwarf Blue Kale	0.888 ± 0.012	0.876 ± 0.014	0.873 ± 0.018	0.876 ± 0.014
	Paksoi	0.817 ± 0.035	0.786 ± 0.041	0.769 ± 0.053	0.788 ± 0.040
DenseNet121	Salad Rocket	**0.907 ± 0.014**	0.887 ± 0.021	0.884 ± 0.025	**0.892 ± 0.018**
	Dwarf Blue Kale	0.855 ± 0.155	0.824 ± 0.178	0.780 ± 0.229	0.824 ± 0.178
	Paksoi	0.769 ± 0.228	0.760 ± 0.173	0.722 ± 0.225	0.758 ± 0.173

V-D GRAD-CAM Analysis

To understand how the best performing model makes decisions, GRAD-CAM visualises the class activations of DenseNet-121. For each cultivar, two images were chosen to represent their original and background-removed versions. Images were selected based on the composition of leaves and background, and all images were taken from days 5-6 to create ambiguity. Figure 12 presents Grad-CAM heatmap visualisations to assess DenseNet-121 classification decisions for original and background-removed images of cultivars.

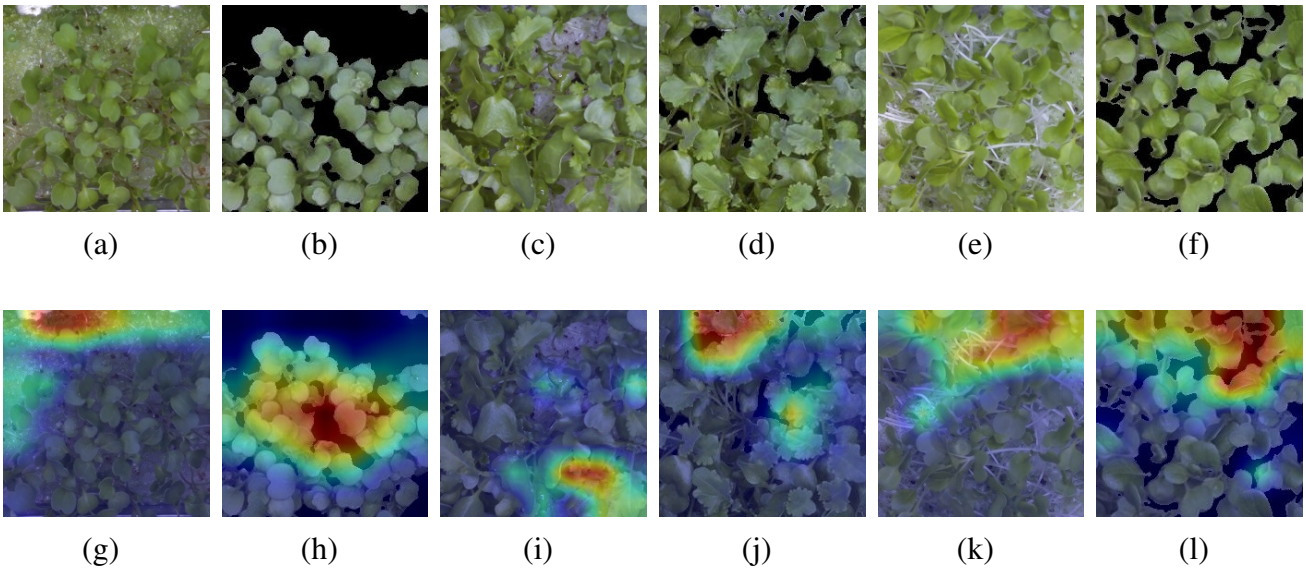


Fig. 12: GRAD-CAM heatmap visualisations of DenseNet-121 classifying both an original image and a background-removed image of salad rocket (**a-b**), dwarf blue kale (**c-d**) and paksoi (**e-f**) as days 6-10. The heatmaps in (**g-l**) visualise the class activation regions of both DenseNet-121 models (trained on original images and background-removed images respectively).

From the left to the right, the images contain salad rocket, dwarf blue kale and paksoi with two images per cultivar. The GRAD-CAM heatmap visualisations in Figures 12g, 12i and 12k indicate that the model trained on original images partially relies on available background information for the classification of the images. For instance, in Figure 12g the model's classification decision is primarily influenced by green algae in the background unlike in Figure 12h. The heatmaps of the background-removed images indicate that the DenseNet-121 model trained on background-removed images focuses mainly on leaf area and outer edges of leaves when no background information is available.

VI Discussion

The U-Net model achieved a F1-score of 0.875 in background segmentation. However, the validation loss is unstable suggesting a lack of generalisability. Training a model to segment three cultivars turned out to be suboptimal. However, given the limited time and the time-consuming process of manually cleaning segmentation masks to remove metal edges and artifacts, it was not feasible to train separate models for each cultivar. In the end, a single U-Net model was trained across all cultivars and outperformed the simpler method of using relative depth estimation and Otsu's thresholding.

Feature extraction using the GLCM failed to distinguish the control group from the test group. No clear feature divergence was observed between the two groups. No features were identified as predictors for nutrient deficiency symptoms in dwarf blue kale, paksoi or salad rocket. The feature extraction results indicate that the test group cultivars were not sufficiently stressed to produce observable symptoms. One possible explanation is that the pH-balanced water still had an electrical conductivity (EC) level too high to sufficiently stress cultivars. Another possibility is that the U-Net model poorly segmented images, due to an unstable validation loss, and consequently distorted the GLCM features. No clear change point t_c could be calculated due to these limitations.

The image classification results show that healthy and nutrient-deficient cultivars can be distinguished with high accuracy. DenseNet-121 consistently outperformed both ResNet50 and EfficientNet-B0 across cultivars. This suggests that deeper architectures with more interconnected layers are better suited for this task. Interestingly, DenseNet-121 achieved both the highest F1-score for salad rocket and the lowest F1-score for paksoi. These major differences underline the importance of training cultivar-specific models.

The GRAD-CAM analysis suggests that background segmentation increases the performance of image classification models. The qualitative comparison between original and background-removed images suggests that the model trained on background-removed images relies on available background information for classification. This may confuse models resulting in a lower accuracy and F1-score. However, the model trained on background-removed images cannot be influenced by background information and therefore focuses the model's attention to leaf areas for classification.

To conclude, this thesis presents an approach for segmenting cultivar images from complex backgrounds. Background segmentation helps to build more accurate and interpretable image classification models. DenseNet-121 was the best model for distinguishing between healthy and visibly stressed cultivars. Growy can use this knowledge to build cultivar-specific models for stress detection. This ultimately helps identifying stressful farm condi-

tions which leads to a more efficient vertical farming system.

VI-A Limitations

Various limitations of the vertical farming system may have influenced the experiments. Data collection at Growy was challenging having only one available experiment layer and limited time to conduct experiments. Robot errors resulted in unusable images which may have affected mean feature values. The farm faces unstable climate conditions and conducting control group and test group experiments simultaneously on the same layer was not feasible. The water system can either supply a nutrient solution or pH-balanced water to a layer without being able to differentiate between gutters or batches. Consequently, data collection was time-consuming, so there was no opportunity to repeat experiments and complete another full growth cycle for the cultivars.

Despite using data augmentation, the labeled dataset used to train the U-Net model was relatively small and therefore lacked accuracy and generalisability. Piecewise-segmented regression could not determine the change point t_c due to minimal differences in mean feature values between the test group and control group. The image classification models were trained on data from days 1-5 and days 6-10 which makes it unclear whether the models are learning to differentiate between the ages of the cultivars or the visual symptoms of stress. Further research is needed to separate these factors and better understand what the models are actually detecting. There is also the possibility that stress might arise at different change points across cultivars.

VI-B Future Work

In future research, separate U-Net models should be trained for each cultivar using larger datasets. These models will likely have a more stable validation loss and higher accuracy in segmentation tasks. The data quality may improve if the control group and test group captured images simultaneously in the same environment. The pH-balanced water may not have produced enough stress, so lower EC levels should be used to widen the gap between the test group and control group EC levels. This will hopefully lead to more diverging measurements of feature values between the test group and control group.

Additional cultivars could be investigated to get more insight into which features predict stress. Additionally, other methods such as hyperspectral or thermal imaging may be able to identify early stress symptoms not visible in RGB images. Furthermore, cultivar-specific segmentation and classification models for each cultivar may result in more reliable stress detection, image segmentation and image classification. Lastly, a region-based CNN could be trained to gain more information about the location of stress in the images.

VII Conclusion

This thesis demonstrates the potential of deep learning techniques, such as U-Net for image segmentation and DenseNet-121 for image classification, in the vertical farming industry. Growy aims to automate stress detection to reduce the time spent manually searching the farm for stressed cultivars. This gives biologists at Growy the time to focus their efforts on finding the causes of stress rather than locating stressed cultivars.

The instability of the U-Net validation loss and the feature extraction limitations highlight some challenges in the approach presented in this thesis. The feature extraction results suggest that the cultivars in the test group may not have experienced sufficient nutrient stress. This could potentially be caused by insufficiently low EC levels in pH-balanced water. Another reason could be that the U-Net model does not segment images sufficiently skewing the GLCM feature values.

Even though the feature extraction results are underwhelming, the binary image classification models are able to accurately distinguish between healthy and visibly stressed cultivars. GRAD-CAM heatmap visualisations indicate that DenseNet-121 trained on background-removed images focuses more on relevant leaf area while DenseNet-121 trained on original images relies more on available background information.

The identification of stress symptoms and specific features that indicate and predict nutrient deficiency was limited. However, being able to distinguish healthy from visibly stressed cultivars using image classification models indicates it is possible to apply cultivar-specific stress detection models. Future work could focus on larger datasets, cultivar-specific segmentation models and non-RGB methods to predict stress in cultivars. Future work could ultimately lead to building robust models capable of automating stress detection in vertical farming systems.

REFERENCES

[1] M. A. Altieri and C. I. Nicholls, "The adaptation and mitigation potential of traditional agriculture in a changing climate," *Climatic change*, vol. 140, no. 1, pp. 33–45, 2017.

[2] M. S. Mir, N. B. Naikoo, R. H. Kanth, F. Bahar, M. A. Bhat, A. Nazir, S. S. Mahdi, Z. Amin, L. Singh, W. Raja *et al.*, "Vertical farming: The future of agriculture: A review," *The Pharma Innovation Journal*, vol. 11, no. 2, pp. 1175–1195, 2022.

[3] S. Van Delden, M. SharathKumar, M. Butturini, L. Graamans, E. Heuvelink, M. Kacira, E. Kaiser, R. Klamer, L. Klerkx, G. Kootstra *et al.*, "Current status and future challenges in implementing and upscaling vertical farming systems," *Nature Food*, vol. 2, no. 12, pp. 944–956, 2021.

[4] R. Uchida, "Essential nutrients for plant growth: nutrient functions and deficiency symptoms," *Plant nutrient management in Hawaii's soils*, vol. 4, pp. 31–55, 2000.

[5] J. Behmann, J. Steinrücken, and L. Plümer, "Detection of early plant stress responses in hyperspectral images," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 93, pp. 98–111, 2014.

[6] D. Neina, "The role of soil ph in plant nutrition and soil remediation," *Applied and environmental soil science*, vol. 2019, no. 1, p. 5794869, 2019.

[7] A. V. Barker and D. J. Pilbeam, *Handbook of plant nutrition*. CRC press, 2015.

[8] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 24, no. 7, pp. 971–987, 2002.

[9] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural features for image classification," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-3, no. 6, pp. 610–621, 1973.

[10] H. Waghmare, R. Kokare, and Y. Dandawate, "Detection and classification of diseases of grape plant using opposite colour local binary pattern feature and machine learning for automated decision support system," in *2016 3rd international conference on signal processing and integrated networks (SPIN)*. IEEE, 2016, pp. 513–518.

[11] N. Sabri, N. S. Kassim, S. Ibrahim, R. Roslan, N. N. A. Mangshor, and Z. Ibrahim, "Nutrient deficiency detection in maize (zea mays l.) leaves using image processing," *IAES International Journal of Artificial Intelligence*, vol. 9, no. 2, p. 304, 2020.

[12] M. J. Sulastri, D. R. Sulistyaningrum, and H. Nurhadi, "Detection of nutrient deficiency in rice plants based on leaf image," in *2021 International Conference on Advanced Mechatronics, Intelligent Manufacture and Industrial Automation (ICAMIMIA)*, 2021, pp. 143–148.

[13] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun, "Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer," 2020. [Online]. Available: https://arxiv.org/abs/1907.01341

[14] R. Birkl, D. Wofk, and M. Müller, "Midas v3.1 – a model zoo for robust monocular relative depth estimation," 2023. [Online]. Available: https://arxiv.org/abs/2307.14460
[15] S. F. Bhat, R. Birkl, D. Wofk, P. Wonka, and M. Müller, "Zoedepth: Zero-shot transfer by combining relative and metric depth," 2023. [Online]. Available: https://arxiv.org/abs/2302.12288
[16] B. Ke, A. Obukhov, S. Huang, N. Metzger, R. C. Daudt, and K. Schindler, "Repurposing diffusion-based image generators for monocular depth estimation," 2024. [Online]. Available: https://arxiv.org/abs/2312.02145
[17] A. Bochkovskii, A. Delaunoy, H. Germain, M. Santos, Y. Zhou, S. R. Richter, and V. Koltun, "Depth pro: Sharp monocular metric depth in less than a second," 2024. [Online]. Available: https://arxiv.org/abs/2410.02073
[18] P. K. Sahoo, S. Soltani, and A. K. Wong, "A survey of thresholding techniques," *Computer vision, graphics, and image processing*, vol. 41, no. 2, pp. 233–260, 1988.
[19] P. C. Chen and T. Pavlidis, "Segmentation by texture using a co-occurrence matrix and a split-and-merge algorithm," *Computer graphics and image processing*, vol. 10, no. 2, pp. 172–182, 1979.
[20] F. Cheevasuvit, H. Maitre, and D. Vidal-Madjar, "A robust method for picture segmentation based on a split-and-merge procedure," *Computer vision, graphics, and image processing*, vol. 34, no. 3, pp. 268–281, 1986.
[21] S.-Y. Chen, W.-C. Lin, and C.-T. Chen, "Split-and-merge image segmentation based on localized feature analysis and statistical tests," *CVGIP: Graphical Models and Image Processing*, vol. 53, no. 5, pp. 457–475, 1991.
[22] J. Canny, "A computational approach to edge detection," *IEEE Transactions on pattern analysis and machine intelligence*, no. 6, pp. 679–698, 1986.
[23] I. Sobel, G. Feldman *et al.*, "A 3x3 isotropic gradient operator for image processing," *a talk at the Stanford Artificial Project in*, vol. 1968, pp. 271–272, 1968.
[24] D. Marr and E. Hildreth, "Theory of edge detection," *Proceedings of the Royal Society of London. Series B. Biological Sciences*, vol. 207, no. 1167, pp. 187–217, 1980.
[25] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," 2021. [Online]. Available: https://arxiv.org/abs/2010.11929
[26] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, "Segment anything," 2023. [Online]. Available: https://arxiv.org/abs/2304.02643
[27] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson, E. Mintun, J. Pan, K. V. Alwala, N. Carion, C.-Y. Wu, R. Girshick, P. Dollár, and C. Feichtenhofer, "Sam 2: Segment anything in images and videos," 2024. [Online]. Available: https://arxiv.org/abs/2408.00714
[28] T. Ren, S. Liu, A. Zeng, J. Lin, K. Li, H. Cao, J. Chen, X. Huang, Y. Chen, F. Yan, Z. Zeng, H. Zhang, F. Li, J. Yang, H. Li, Q. Jiang, and L. Zhang, "Grounded SAM:

Assembling open-world models for diverse visual tasks," 2024. [Online]. Available: https://arxiv.org/abs/2401.14159

[29] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, Q. Jiang, C. Li, J. Yang, H. Su, J. Zhu, and L. Zhang, "Grounding DINO: Marrying dino with grounded pre-training for open-set object detection," 2024. [Online]. Available: https://arxiv.org/abs/2303.05499

[30] A. Neubeck and L. Van Gool, "Efficient non-maximum suppression," in *18th international conference on pattern recognition (ICPR'06)*, vol. 3. IEEE, 2006, pp. 850–855.

[31] X. Zhao, W. Ding, Y. An, Y. Du, T. Yu, M. Li, M. Tang, and J. Wang, "Fast segment anything," 2023. [Online]. Available: https://arxiv.org/abs/2306.12156

[32] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural computation*, vol. 1, no. 4, pp. 541–551, 1989.

[33] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," 2015. [Online]. Available: https://arxiv.org/abs/1411.4038

[34] L. Deng, "The mnist database of handwritten digit images for machine learning research [best of the web]," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, 2012.

[35] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.

[36] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[37] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, Eds., vol. 25. Curran Associates, Inc., 2012. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf

[38] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," 2014. [Online]. Available: https://arxiv.org/abs/1409.4842

[39] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2015. [Online]. Available: https://arxiv.org/abs/1409.1556

[40] L. Yang, B. Kang, Z. Huang, Z. Zhao, X. Xu, J. Feng, and H. Zhao, "Depth anything v2," 2024. [Online]. Available: https://arxiv.org/abs/2406.09414

[41] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, M. Assran, N. Ballas, W. Galuba, R. Howes, P.-Y. Huang, S.-W. Li, I. Misra, M. Rabbat, V. Sharma, G. Synnaeve, H. Xu, H. Jegou, J. Mairal, P. Labatut, A. Joulin, and P. Bojanowski, "Dinov2:

Learning robust visual features without supervision," 2024. [Online]. Available: https://arxiv.org/abs/2304.07193

[42] R. Ranftl, A. Bochkovskiy, and V. Koltun, "Vision transformers for dense prediction," 2021. [Online]. Available: https://arxiv.org/abs/2103.13413

[43] A. S. Abutaleb, "Automatic thresholding of gray-level pictures using two-dimensional entropy," *Computer vision, graphics, and image processing*, vol. 47, no. 1, pp. 22–32, 1989.

[44] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," 2015. [Online]. Available: https://arxiv.org/abs/1505.04597

[45] The GIMP Development Team, "Gnu image manipulation program (gimp), version 3.0.4. community, free software (license gplv3)," 2025, version 3.0.4, Free Software. [Online]. Available: https://gimp.org/

[46] S. Islam, M. N. Reza, S. Ahmed, Samsuzzaman, K.-H. Lee, Y. J. Cho, D. H. Noh, and S.-O. Chung, "Nutrient stress symptom detection in cucumber seedlings using segmented regression and a mask region-based convolutional neural network model," *Agriculture*, vol. 14, no. 8, p. 1390, 2024.

[47] M. Kabir, F. Unal, T. C. Akinci, A. A. Martinez-Morales, and S. Ekici, "Revealing glcm metric variations across a plant disease dataset: A comprehensive examination and future prospects for enhanced deep learning applications," *Electronics*, vol. 13, no. 12, p. 2299, 2024.

[48] K.-P. Lu and S.-T. Chang, "An advanced segmentation approach to piecewise regression models," *Mathematics*, vol. 11, no. 24, p. 4959, 2023.

[49] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015. [Online]. Available: https://arxiv.org/abs/1512.03385

[50] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," 2018. [Online]. Available: https://arxiv.org/abs/1608.06993

[51] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," 2020. [Online]. Available: https://arxiv.org/abs/1905.11946

[52] M. Abadi *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous systems," 2015, https://www.tensorflow.org/.

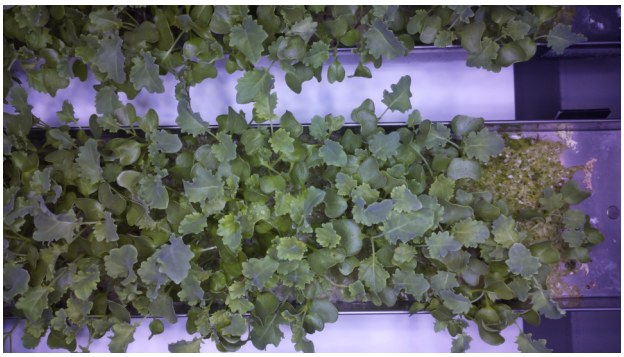
[53] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," *International Journal of Computer Vision*, vol. 128, no. 2, p. 336–359, Oct. 2019. [Online]. Available: http://dx.doi.org/10.1007/s11263-019-01228-7

[54] Jacob Gil, "Introduction: Advanced explainable ai for computer vision — advanced ai explainability with pytorch-gradcam," https://jacobgil.github.io/pytorch-gradcam-book/introduction.html, n.d., accessed: 2025-07-10.

APPENDIX A

NUTRIENT DEFICIENCY IN CULTIVARS OVER TIME

(a) Dwarf Blue Kale – Day 1



(b) Dwarf Blue Kale – Day 10

(c) Paksoi – Day 1

(d) Paksoi – Day 10

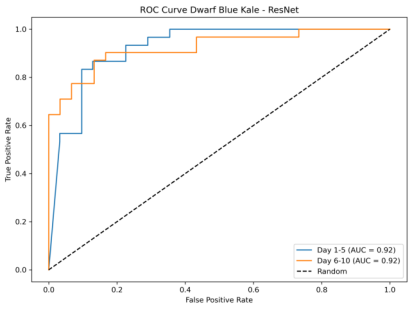
(e) Salad Rocket – Day 1

(f) Salad Rocket – Day 10

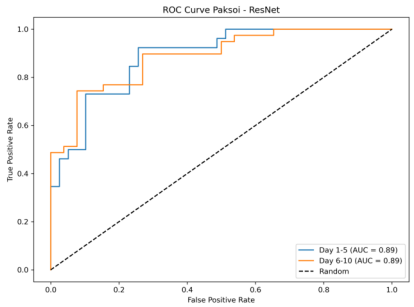
Fig. 13: Cultivar stress in test group over time: day 1 (left) to day 10 (right).

APPENDIX B

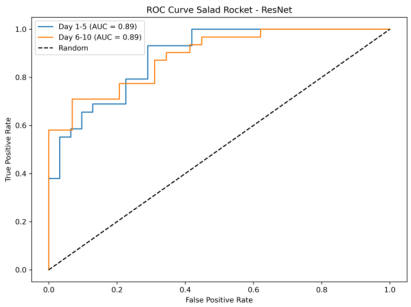
ROC-AUC CURVES



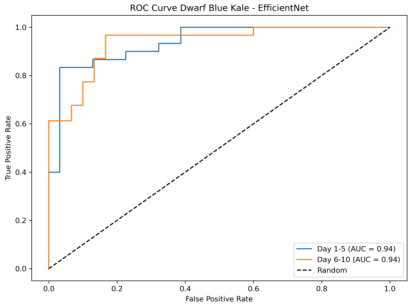
(a) Dwarf Blue Kale (ResNet50)



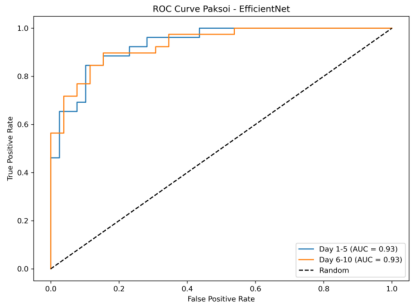
(b) Paksoi (ResNet50)



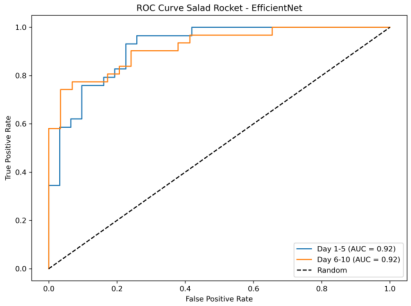
(c) Salad Rocket (ResNet50)



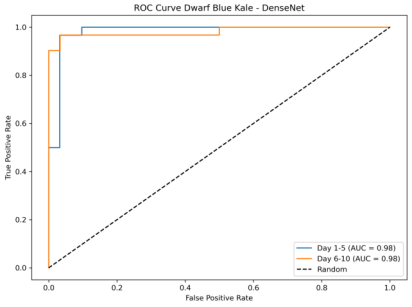
(d) Dwarf Blue Kale (EfficientNet-B0)



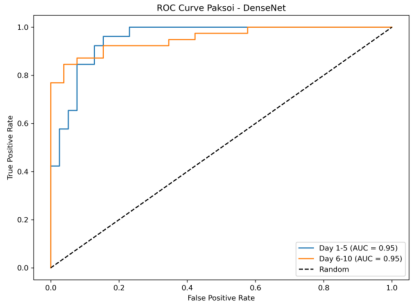
(e) Paksoi (EfficientNet-B0)



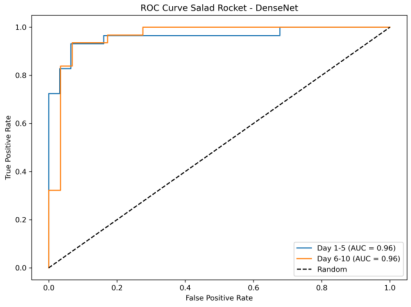
(f) Salad Rocket (EfficientNet-B0)



(g) Dwarf Blue Kale (DenseNet-121)



(h) Paksoi (DenseNet-121)

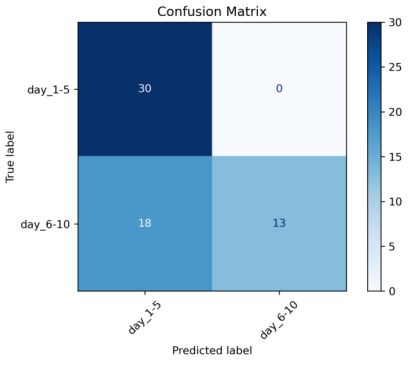


(i) Salad Rocket (DenseNet-121)

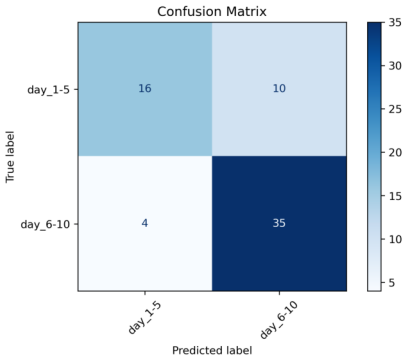
Fig. 14: ROC-AUC curves of cultivars across three models - ResNet50 (top row), EfficientNet-B0 (middle row) and DenseNet-121 (bottom row) - trained on background-removed images.

APPENDIX C

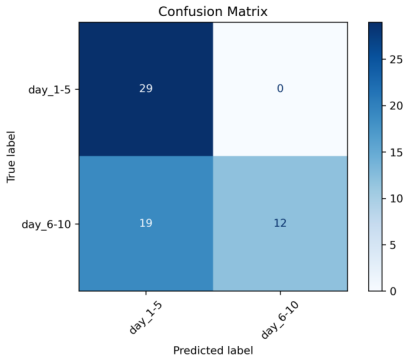
CONFUSION MATRICES



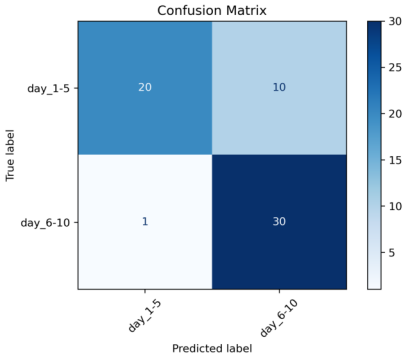
(a) Dwarf Blue Kale (ResNet50)



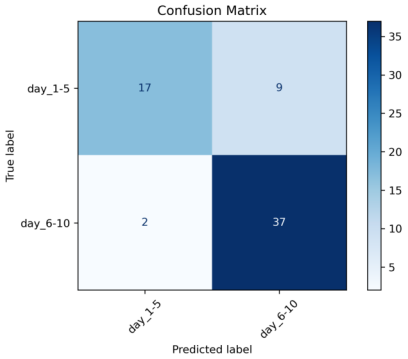
(b) Paksoi (ResNet50)



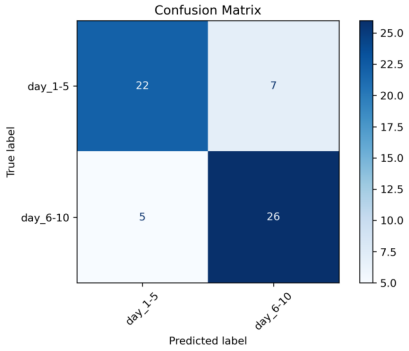
(c) Salad Rocket (ResNet50)



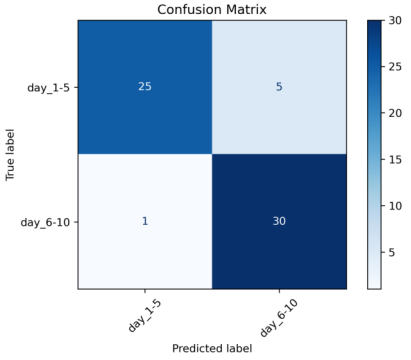
(d) Dwarf Blue Kale (EfficientNet-B0)



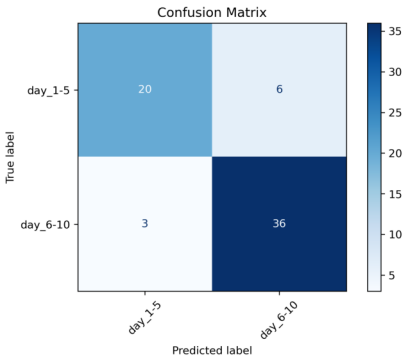
(e) Paksoi (EfficientNet-B0)



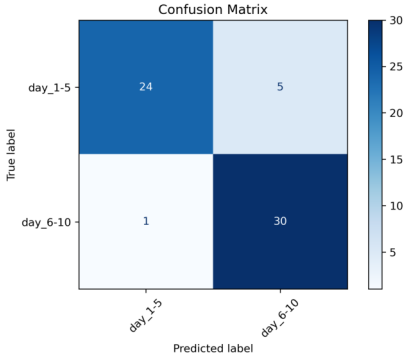
(f) Salad Rocket (EfficientNet-B0)



(g) Dwarf Blue Kale (DenseNet-121)



(h) Paksoi (DenseNet-121)



(i) Salad Rocket (DenseNet-121)

Fig. 15: Confusion matrices of cultivars across three models - ResNet50 (top row), EfficientNet-B0 (middle row) and DenseNet-121 (bottom row) - trained on background-removed images.