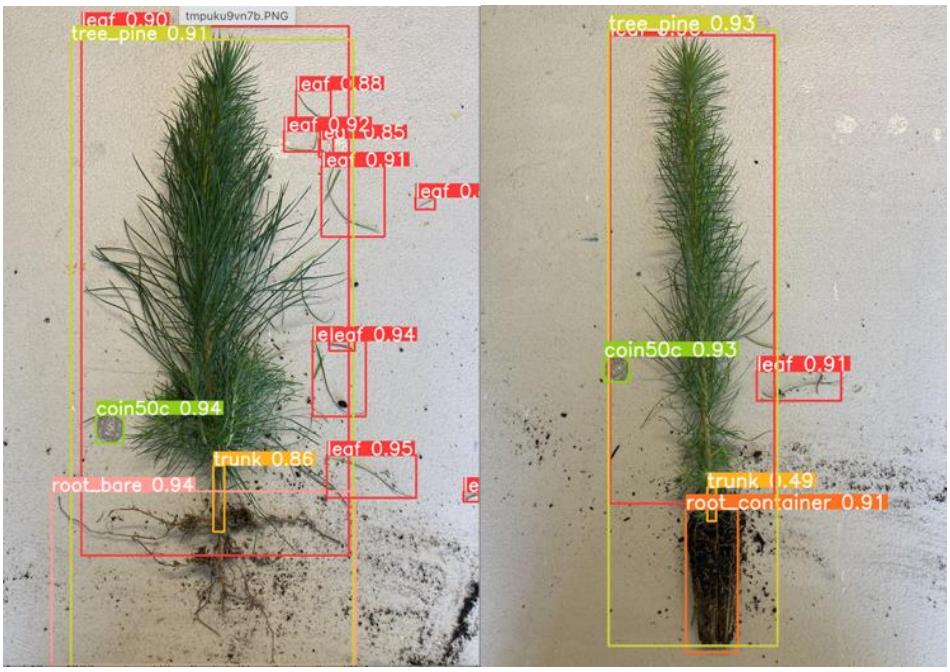
Precision Silviculture Partnership

# Seedling Selection using Computer Vision

**Dr Brian Russell**




**Date:** **20 May 2023**

**Report No:** PSP-T009

FOREST GROWERS RESEARCH

# TABLE OF CONTENTS



**Disclaimer**

This report has been prepared by Contempo Holdings Ltd for Forest Growers Research Ltd (FGR) subject to the terms and conditions of a research fund agreement dated 1 April 2014.

The opinions and information provided in this report have been provided in good faith and on the basis that every endeavour has been made to be accurate and not misleading and to exercise reasonable care, skill and judgement in providing such opinions and information.

Under the terms of the Services Agreement, Contempo Holdings Ltd liability to FGR in relation to the services provided to produce this report is limited to the value of those services. Neither Scion nor any of its employees, contractors, agents or other persons acting on its behalf or under its control accept any responsibility to any person or organisation in respect of any information or opinion provided in this report in excess of that amount.

supported by forestgrowers commodity levy

# EXECUTIVE SUMMARY

In productive forestry, seedlings, such as radiata pine, are grown for scaled forestry planting in nurseries using two planting methods: containerised and bare root. These seedlings are generally lifted and selected manually based on standard measurements. . Machine based harvesting and selection is in development in an effort to scale beyond available labour and improve data collection for research outcomes. The lifting ‘machinery’ can be viewed as a combination of mechanical and computational systems. The aim of the project is to investigate the feasibility of automating seedling selection using camera based computational approaches to accurately select seedlings in representative field environments such as variation in light, plant occlusion, background objects, and orientation. The work shows that a hybrid approach using artificial intelligence, AI, and heuristics could detect different tree species, across orientation, backgrounds, and lighting. The AI model detected individual trees in an image, which included loose dirt and leaves, identifying shoots, trunk, and root. Subcategories included tree species (pine and redwood) and root type (bare and container). Measurements included number of shoots, height of tree, brown leaves, root collar diameter, root metric, tree angle, pixel size. Heuristics used rules to make a pass or fail decision taking into account species and root type. The system output included a report for data collection and explanation of each decision. Total analysis time was 21 ms giving a tree detection rate of 47 per second exceeding the requirement of 10 trees per second. The results show it is possible to accurately detect, measure and select tree seedlings at a speed practical for field use. The feasibility prototype was developed based on an in-shed conveyor belt environment. However, as mechanization develops it is possible that camera based automated seedling detection would be embedded into a mechanized system. This would allow a pathway for total in-field decisioning, which is the preference of many nursery operators today.

# INTRODUCTION

## Background

Seedling trees are grown in nurseries either as containerised or bare root stock. Mechanisation of seedling lifting is under development to enable scale beyond available labour and also enhance longitudinal data analysis for crop optimisation through data collection. Manual lift requires a person to pick the seedlings into bundles then box them ready for short term storage prior to planting in forested areas. The person must physically pull the plant and make an assessment of whether to keep the seedling based on grading rules [1] that can vary between nurseries, geographic location, season and market requirements.

**Computer vision, CV** tasks manage images, perform analysis and communicate or report the output. One library commonly used for CV on plants is called PlantCV [2] shown in Figure 1. Example applications include phenotyping and watching plants grow [3]. The software used for this project was Python with libraries such as OpenCV supporting image manipulation, filtering and deep learning model support such as the open neural network exchange, ONNX [4]. Models written natively in Python machine learning libraries such as PyTorch. CV has been used for pine tree analysis in laboratory settings as outlined in **McGuinness' paper** [5], including work flow shown in. This paper used two cameras in a mechanical frame to calculate height, root collar diameter, RCD, and root analysis. CV approaches typically reduce the computational complexity by making the target in the image reproducibly unambiguous by using a clean white background and white light. When lighting is variable filtering is performed using HSV (hue, saturation, value) rather than RGB (red, green, blue). The CV vision approach can result in unreliable results when lighting or target colour changes as the range of HSV values need to be updated. Furthermore, occlusion of branches or presence of objects in the background can invalidate selection assumptions leading to errors. This makes this approach, at least as the standalone method, unsuitable for our requirements which include conveyor belts with dirt presence, occlusion from overlapping seedling and ultimately. In an 'in-field' embedded machine context with less control of environment.

*Figure 1 Example image analysis using computer vision [1]*

[1] https://plantcv.readthedocs.io/en/stable/

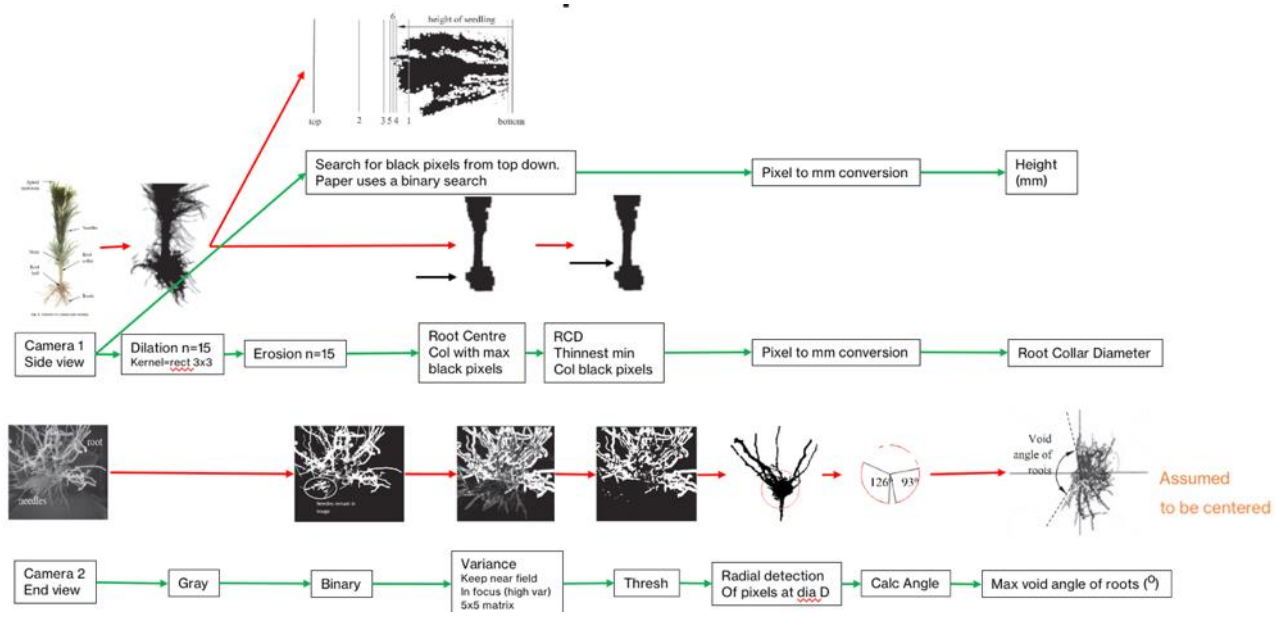



*Figure 2 Workflow for pine tree analysis described by McGuiness.*

**Deep learning** in computer vision applications are more generalisable to different environments and are used in various applications including medical imaging [6], pedestrian detection [7], plant phenotyping [8], plant disease detection [9]. Models have developed rapidly in recent years for both architecture and training methods such as such as faster-CNN (convolution neural network) and YOLOv5[2] (you only look once) [10]. YOLOv5 is a state of the art open source deep learning model designed for realtime object detection [7,9,11] which performs 10x faster than other models with near equivalent accuracy. Used for many realtime applications. An example in figure 3 (pictured right) shows multiple objects detected, the bus is occluded by pedestrians and pixels associated with objects are highlighted using semantic segmentation, where each pixel in the image is associated with a labelled object or the background

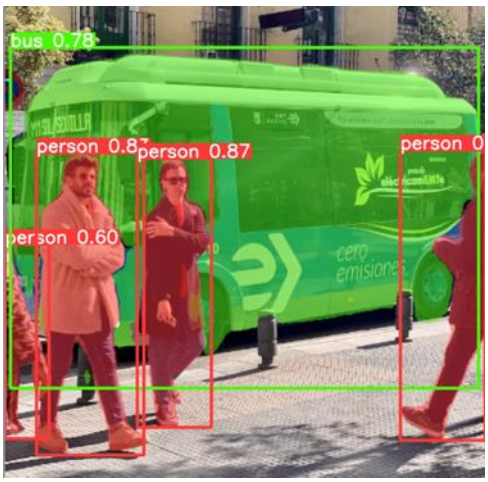



*Figure 3 Object detection with YOLOv5[1]*

**There are trade-offs** between CV and deep neural network, DNN, approaches. CV allows image manipulation, filtering and selection in a reductionist approach that naturally explains the algorithm, does not need large datasets and can be developed quickly and applied reliably once validated with a representative dataset. The disadvantage of CV is that it cannot solve complex problems such as detecting complicated shapes that vary between samples with different lighting conditions. DNN typically requires large datasets, labelling of those images but can perform in changeable conditions and generalise to various unstructured conditions such as changing back grounds. The need for large datasets for DNN can be problematic if data collection is expensive or seasonally limited to certain times in the year. A typically development approach is to start with CV and move to DNN if initial CV results do not meet requirements for deployment.

**This work** researched the development of a software pipeline that could manually label images, train an AI model and perform inference on images followed by a decision model to grade the seedlings and suggest a pass or fail, with an explanation for the decision. The research was staged as shown below.

[2] www.ultralytics.com

- Investigate the best way to classify seedling, choose approach and review (computer platform, decision time, workflow description, use case development, performance specifications)
- Image collection and labelling
- Train computer vision model and test
- Train decision model to make recommendations.
- Demonstrate first prototype.
- Report on performance

## Requirements

There are various requirements for the application deployment based a combination seedling selection rules and in-field practicalities, as shown in Table 1.

*Table 1 Requirements*

| Requirement | Description | Value | Unit |
|---|---|---|---|
| Machinery operations | Should work on different sorting machines | >=2 | |
| Camera amount | Should use the minimum number of cameras, ideally 1. | <=1 | |
| Inference speed | Should process a minimum number of trees per second | 10 | per second |
| Lighting | Should work with different lighting | Sunlight, fluorescent | |
| Dirt and objects | Should work with dirt and random objects in camera view | Conveyor belt, dirt, loose leaves | |
| Obstacles | Should not have to correctly analyse tree if tree is obscured by a certain amount | 5<=% | |
| Calibration | The system should be capable of calibrating the camera to make measurements | | |
| Tree type | Should detect at least one tree type, Pine | pine | |
| Shoot Number | Should be capable of counting number of shoot | >=1 | |
| Shoot Height | Should be capable of measuring shoot height | Shoot height | mm |
| Disease/health | Assess brown needles in top ¼ of tree | Brown/green | % |
| Root type | Should detect two root types, containerised and bare. | Bare, Container | |
| Root Collar Diameter | Should measure RCD diameter | | |
| Root analysis | For containerised assess the percentage of root-with-soil relative to container shape. For bare root asses the quadrants ratio. | | |
| Mycorrhiza | Shall detect and count Mycorrhiza (note this is planned to be implemented at a later stage int eh project) | | |

# METHODS

The methods for this study we've broken into two phases firstly computer vision based on McGuiness' paper we replicated in Python code and assist for accuracy and consistency of detection across the data set of radiata pine and Redwood seedlings. The second phase of the research investigated the use of a deep learning model followed by measurement analysis and finally the heuristics to make the selection rules resulting in a pass fail with explanation of the reasons why a pass or fail occurred.

# PHASE 1. COMPUTER VISION

Ben McGuiness from Waikato University published a paper [5] on seedling sorting. Multiple cameras were used on clean dirt-free seedlings using a seedling handling machine in the laboratory. The approach was elegant for the three measurements produced and the analysis of accuracy is well developed. Grey scale imaging was used which would reduce the possibility of differentiating leaves, trunk or roots and disease detection looking at brown leaves. Dilation operations with the binary black and white image may struggle to determine dirt vs root ball. i.e., this approach requires no dirt on the roots which differs from the example images in the Scion Seedling Selection documents. The approach used computer vision solely and no DL or AI. It was worth considering a deep learning approach in the first stage of this project to understand the tradeoffs between in-field generalizability and accuracy.

Software was developed using the Jupyter Notebook environment to replicate the results in the McGuiness paper and extend the detection and measurement for all the rules needed for pass-fail selection.

Figure 4 shows the HSV spectra for the shoot and root of redwood seedling, which would be used to mathematically subtract from other parts of the seedling or background. When multiple objects are detected in an image then assumptions can be made to further refine detection such is in Figure 5 where the shoot (a) root collar (b) are detected and measured.

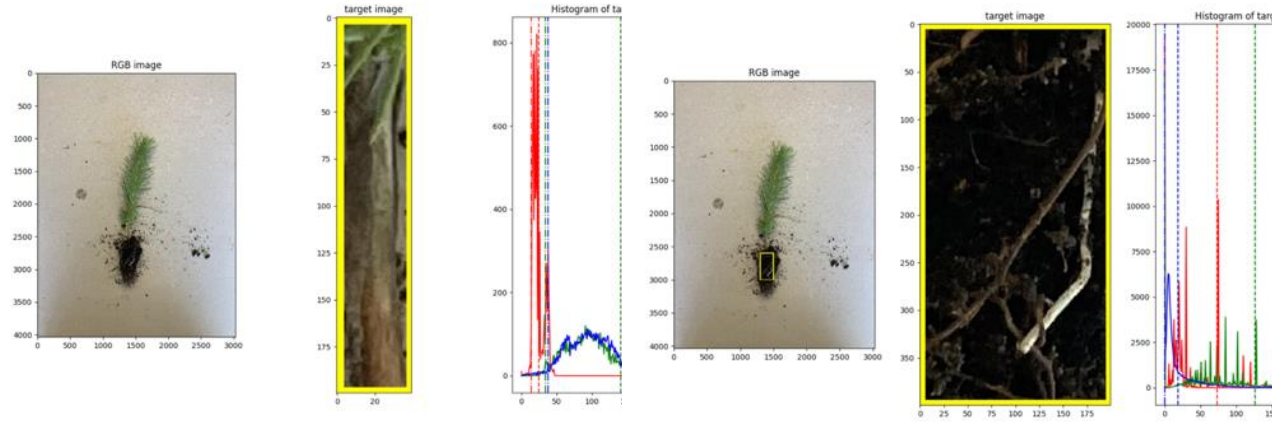
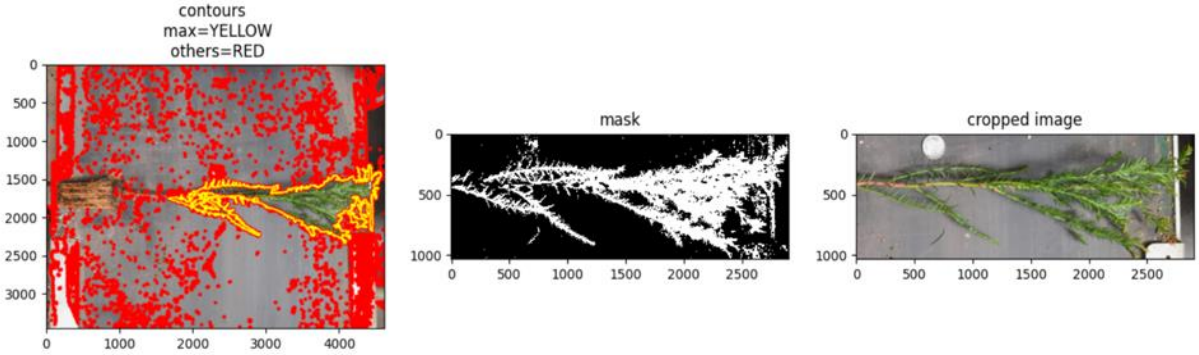



*Figure 4 Hue, saturation and value spectra for (a) root collar and (b) root ball of a pinas radiata seedling*

(a)



(b)

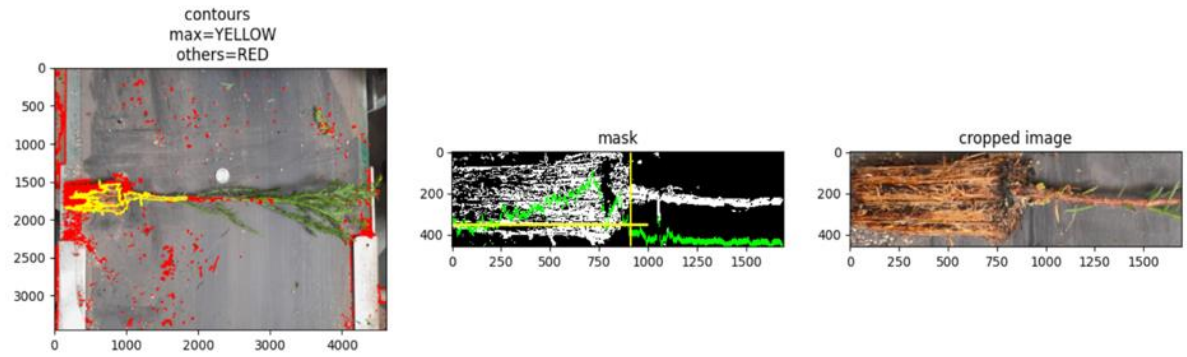



*Figure 5 Background object rejection using contours for (a) redwood shoot and (b) redwood containerised root with number of detected pixels per column (green) and detection threshold for root-to-trunk (yellow)*

# PHASE 2. DEEP LEARNING

The software development for a deep learning system is called a pipeline. This pipeline starts with a data set information and the images are labelled and the AI model is trained with the output being the model architecture and the weights associated with the training. The production system loads the iron models weights and architecture and uses this for detection within each image. Following object detection measurements were taken and passed to the rules engine to determine passed or fail. The final step is to save the measurements pass fail criteria and performance such as frame rate for further analysis if needed.

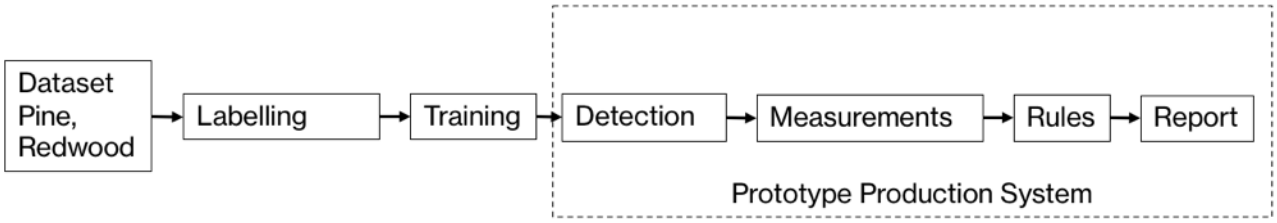



*Figure 6 Software pipeline for AI training and deployment*

## Labelling

The image labelling was performed with a software application, RectLabel[3], initially with object detection boxes and when this did not reliably show all object detection across the data set delay billing method moved to semantic segmentation of objects, as shown in Figure 7. Images where selected to have differences in orientation (horizontal, vertical), tree type (pine, redwood), background (white, black, obstacles, dirt, loose leaves, partial trees, reflections). Classes of objects to inform measurements needed for selection were 'leaf', 'root_bare', 'root_container', 'trunk', 'tree_pine', 'tree_redwood', 'coin50c'. Figure 7 shows example labelling of a pine tree would be a rootball applying pine tree with containerised roots and see I Redwood tree with containerised roots and two trunks. A 50-cent coin was included in each image as this enables diameter to be measured horizontally regardless of angle and this measurement was used to calibrate pixel size for further measurements for the results and expressed in mm. It is not expected that the daily production system will need this image-by-image calibration as the camera will have a fixed distance from the convey belt.

[3] https://rectlabel.com

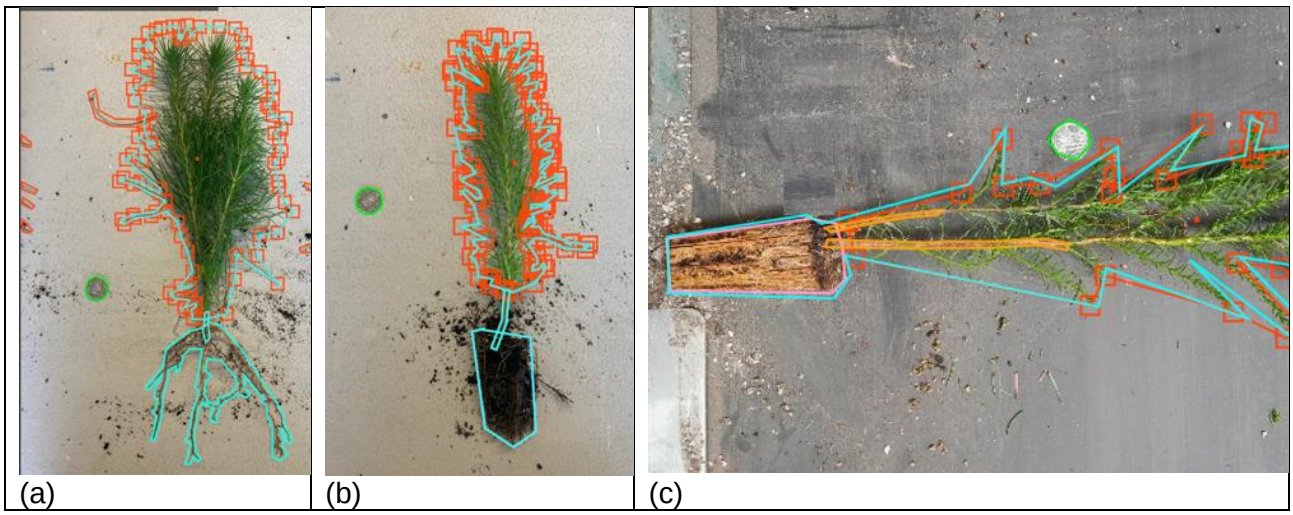



*Figure 7 Labelling of pine and redwood (a) pine with bare root, (b) pine with containerised roots, (c) redwood with two trunks, horizontal and black background.*

## Training

A deep learning model called YOLOv5 was selected based on reported results in real time object detection and accuracy, different model sizes are available trading off processing time and accuracy. Training was performed with computer using Ubuntu operating system with Intel i9 CPU 3.6 GHz and NVIDIA GEOFORCE RTX 2070 Super 8GB GPU. Software was Python 3.10.6 and PyTorch 1.13 with model YOLOv5 v7.0.

The model was trained with a custom dataset described above with 52 images of redwood trees with containerised roots on a black conveyor belt and 25 images of pine trees on a white table with a combination of containerised and bare roots.

Hyper parameters included: Epoch 2048, Batch size 16, learning rate 0.01, with early stopping if performance plateaued.

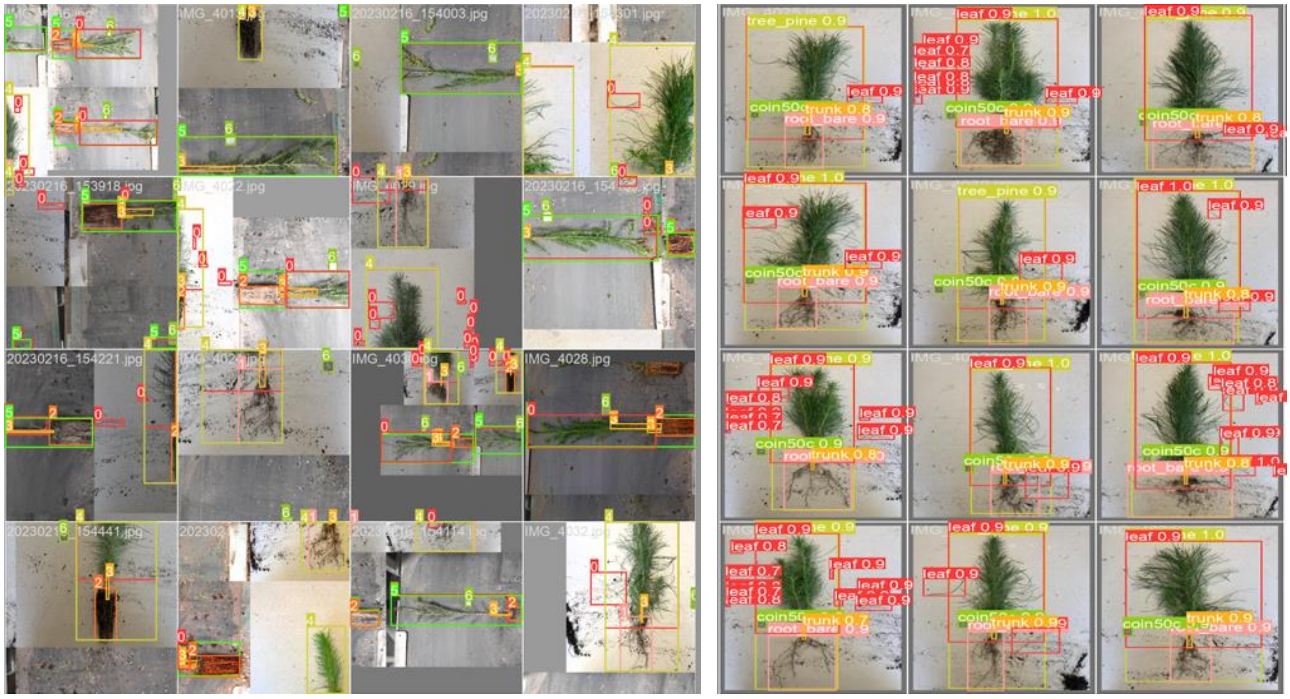



*Figure 8 (a) Training showing YOLOv5 image slicing approach and (b) validation results.*

Training was stopped when accuracy reached a plateau, the reported epochsFigure 10 shows the learning of each size model and how many a pox were required to attain the same level of accuracy. The conclusion was that the nano model is of equal accuracy if trained longer so was chosen for further analysis, however some images did not have some objects detected with the nano model. Due to optical detection challenges with the new model the small model was selected for further development.

| Performance by Model Size | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Model | Images | Instances | Batch | Epochs | Prediction Time (ms Apple Mac) | Precision | Recall | mAP50 | mAP5095 |
| Large | 52 | 230 | 4 | 532 | 1480 | 0.982 | 0.976 | 0.988 | 0.824 |
| Medium | 52 | 55 | 4 | 692 | 760 | 0.986 | 0.974 | 0.988 | 0.847 |
| Small | 52 | 53 | 4 | 1024 | 330 | 0.984 | 0.970 | 0.986 | 0.874 |
| Nano | 52 | 52 | 4 | 2048 | 132 | 0.976 | 0.977 | 0.987 | 0.878 |

*Figure 9 Training performance by model size for Redwood trees*

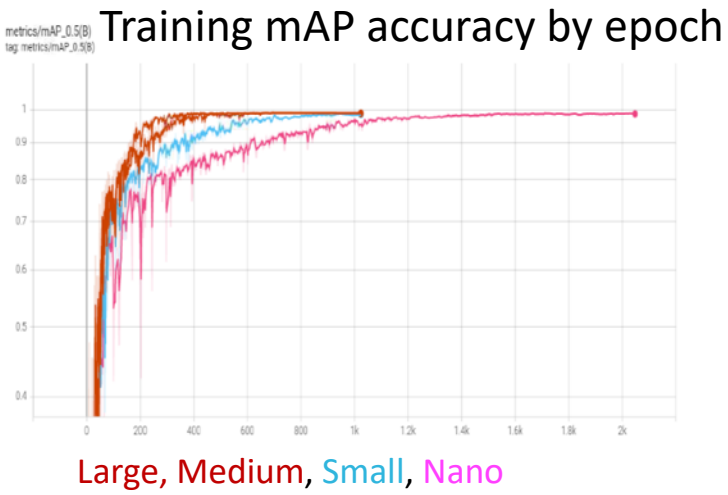



*Figure 10 Training results for model sizes*

Training and validation images are shown in Figure 8, showing how YOLOv5 has a robust image slicing approach to training to improve training rates with less images.

## Inference Accuracy of Tree Objects

Object detection was performed using PyTorch Hub in Python, after initially using OpenCV's DNN library which did not perform well in the image data set. Investigation found the OpenCV DNN library simplifies the YOLOv5 model architecture which loses sensitivity for some challenging objects.

Dimensional accuracy is determined by the pixel size as this is calibrated against the known diameter of a New Zealand 50 cent coin. The average pixel size was 0.1454 mm per pixel or 7 pixels per mm.

## Analysis of Seedlings Post Inference i.e. Measurements

The definition of measurements is shown in Figure 11. Further analysis was performed with computer. Figure 12 shows how the root object detection is used to select a sub image and filter pixels as “background” vs “root” which includes dirt and branch material. The root ball ratio is number of pixels in the object rectangle. Figure 13 shows a pine with bare root, where the root ratio uses the position of the trunk to determine the ratio of left to right roots relative to the root object. In both cases disease is detected as brown leaves / green leaves in the top quarter of the tree, taking into account the orientation of the tree.

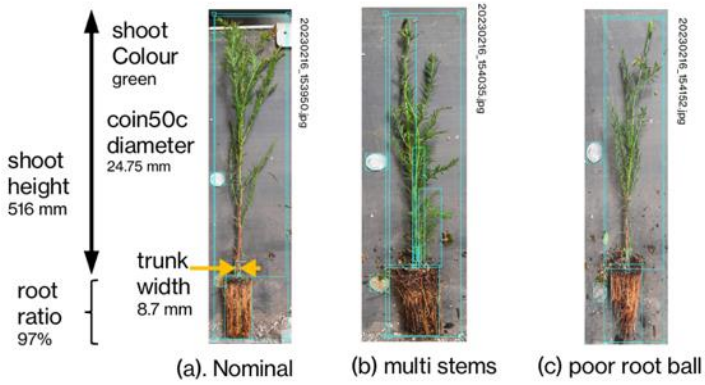



*Figure 11 Definition off seedling measurements*

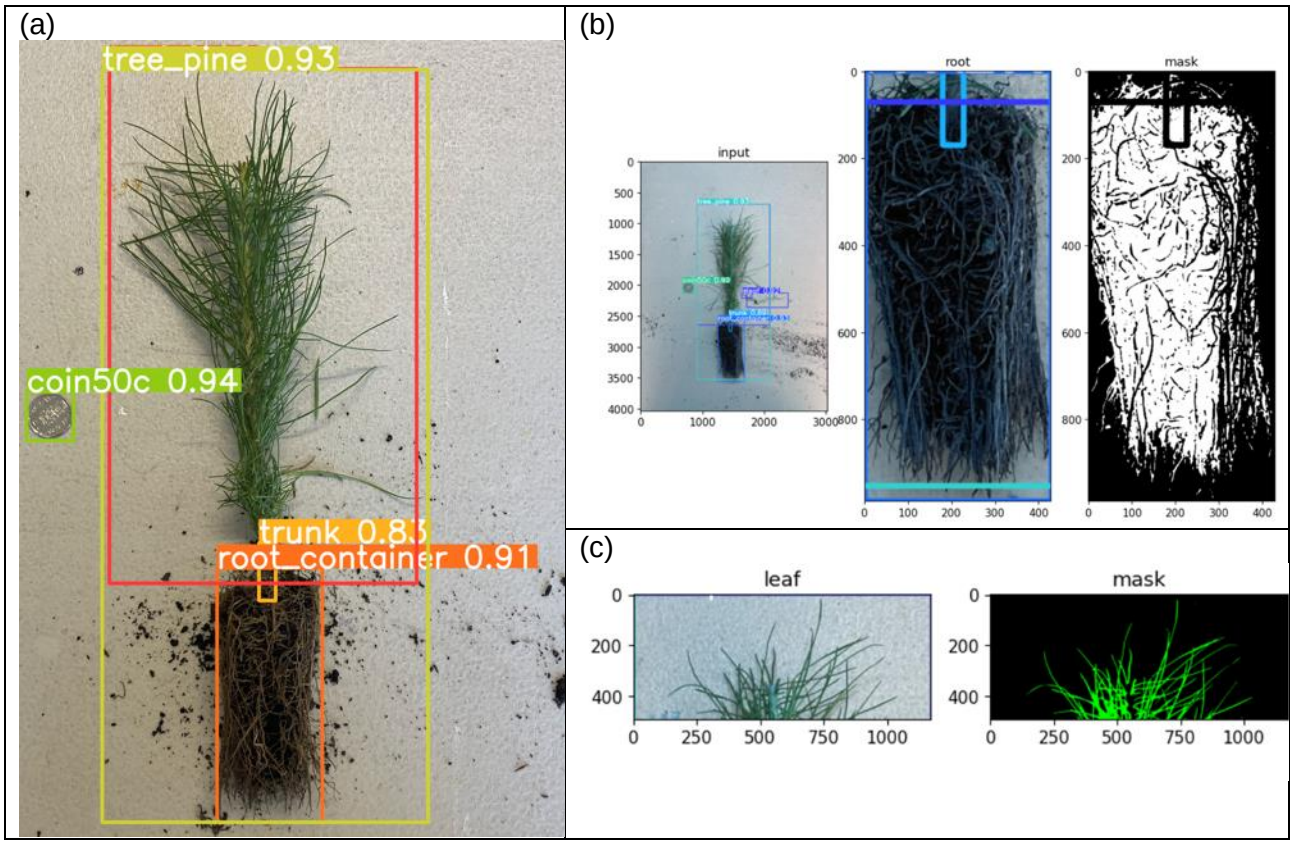



*Figure 12 Pine containerised root (a) Inference results for objects (b) root pixel counting (c) stem colour comparison for top of plant.*

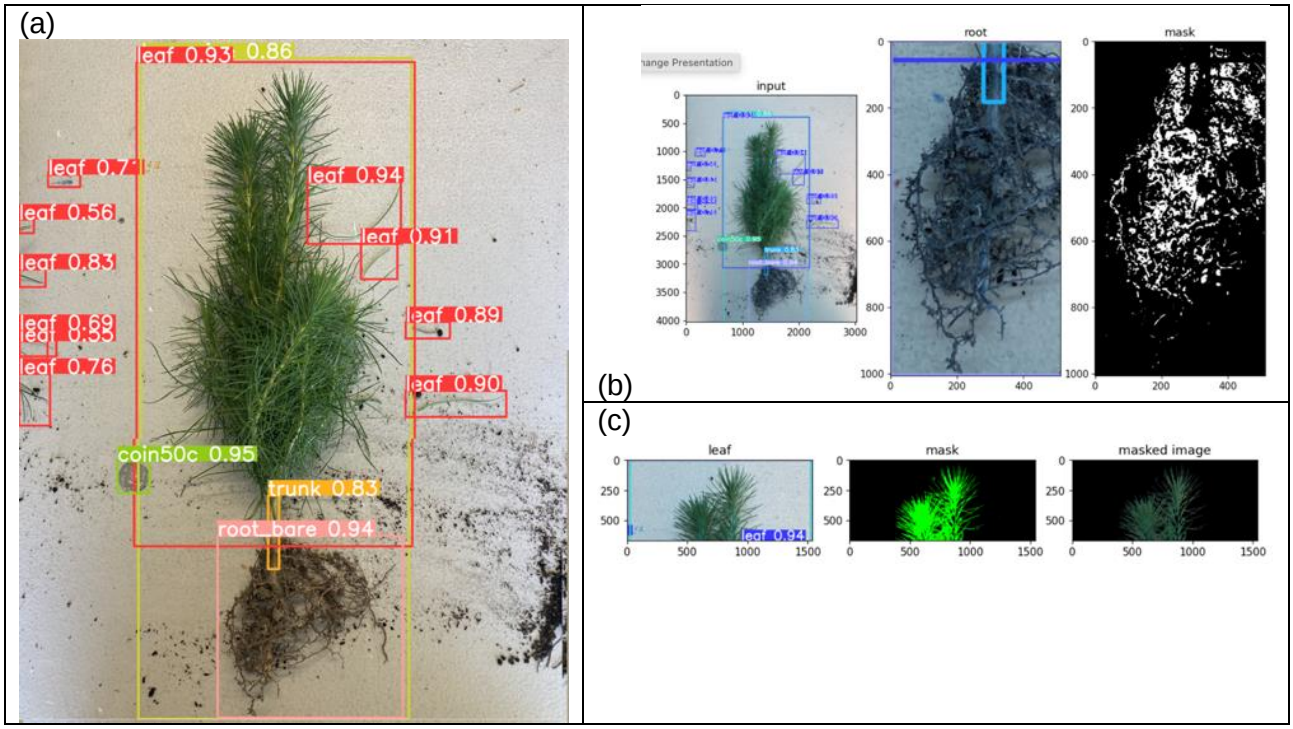



*Figure 13 Pine with bare root (a) Inference results for objects (b) root pixel counting showing left half of root ball (c) stem colour comparison for top of plant.*

## Rules and Selection of Pass or Fail

These measurements are compared against the allowable ranges for each rule as shown in Figure 14. The final result for all images is output to a table, as shown in Figure 17, showing the image label, processing times, objects detected, measurements and pass fail for each rule.

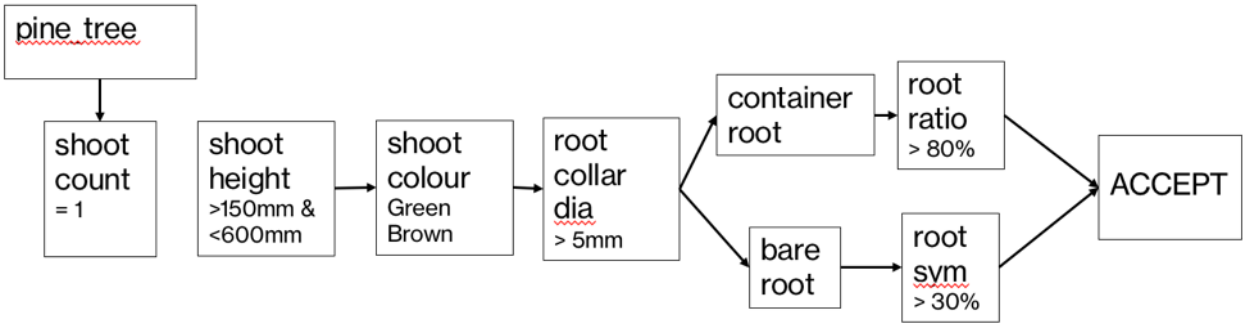



*Figure 14 Rules engine for decision making process for seedling pass/fail.*

# RESULTS

## Computer vision approach

The results of the CV experiment showed this approach was unreliable for seedlings with colour and shape variation, background objects and background colour. Tree detection and root detection was achieved. However, RCD measurement was developed unsuccessfully as the approach did not perform reliably as trunk occlusion meant this method was not reliable for the data images in the dataset. The unreliable nature of detection resulted in no further development. If DL was

needed for reliable detection of some parts of the seedling, then its expected DL is needed for all parts eventually.

## Deep Learning

YOLOv5 Object detection labelling resulted in unreliably detection of trunks. The second approach of YOLOv5 semantic segmentation labelling was successful. It was found that processing time and accuracy where key metrics for the deep learning and just as important was the detection of all objects for all images, such as numbers of trunks when occlusion occurred.

Trunks and bare roots where difficult to accurately identify, resulting in not using the OpenCV DNN library with ONNX models and instead using PyTorch allowing YOLOv5 native model and weights. Furthermore, the nano version YOLOv5 did not perform or some images and the YOLOv5 small model was used.

## Accuracy

The largest measurement errors exist due to object detection ambiguity for the definition of object boundaries, e.g., where does a branch start and finish when there are loose leaves, or where does the root-ball edge occur when there is loose first under roots.

## Processing time

Table 2 shows processing times for three compute capabilities and 2 image resolutions. It can be seen that 640x480 is a faster for both inference and post processing. The small model was used here as the nano model missed a ‘trunk’ on one image. At present the post processing time is dominant so reducing inference times by going to a smaller model would not increase frame rate significantly. Image loading for large images is significant, however this would not be applicable for the end use of a live camera feed.

*Table 2 Image processing times for 24 images on local hard drive, YOLOv5 model = small*

| YOLOv5 Model Size | Computer | Image Size | Image Load (ms) | Inference Time (ms) | Analysis Time (ms) | Save Results To Hard Drive | Total Process Time (ms) | Frame Rate (4) (fps) |
|---|---|---|---|---|---|---|---|---|
| small | Intel I9 GPU(1) | 3024×4032 | 132 | 9 | 22 | 4 | 167 | 28.5 |
| small | Intel I9 GPU(1) | 480 x 640 | 3 | 7 | 12 | 2 | 26 | 47.6 |
| small | Intel I9(2) | 3024×4032 | 130 | 38 | 22 | 4 | 194 | 15.6 |
| small | Intel I9(2) | 480 x 640 | 4 | 31 | 13 | 1 | 49 | 22.2 |
| small | MacBook Pro(3) | 3024 x 4032 | 192 | 334 | 52 | 7 | 585 | 2.5 |
| small | MacBook Pro (3) | 480 x 640 | 5.5 | 244 | 19 | 1.5 | 270 | 3.7 |
| nano | Intel I9 GPU(1) | 3024×4032 | 135 | 8.3 | 27 | 5 | 190 | 24.8 |
| nano | Intel I9 GPU(1) | 480 x 640 | 2.9 | 7 | 13 | 4 | 27 | 41.6 |
| nano | Intel I9(2) | 3024×4032 | 136 | 19 | 24 | 9 | 183 | 19.2 |
| nano | Intel I9(2) | 480 x 640 | 3 | 16 | 12 | 4 | 35 | 31.2 |
| nano | MacBook Pro(3) | 3024 x 4032 | 173 | 114 | 51 | 10 | 348 | 5.7 |
| nano | MacBook Pro (3) | 480 x 640 | 6 | 142 | 30 | 9 | 187 | 5.5 |

Notes:
(1) CPU = Intel i9 3.6GHz, GPU = NVIDIA GEOFORCE RTX 2070 Super 8GB
(2) CPU = Intel i9 3.6GHz
(3) MacBook Pro, Intel i5 3.1 GHz, 500Gb Flash Drive
(4) Frame rate does not include hard drive load time as the system will be using live video, not static images from the hard drive.

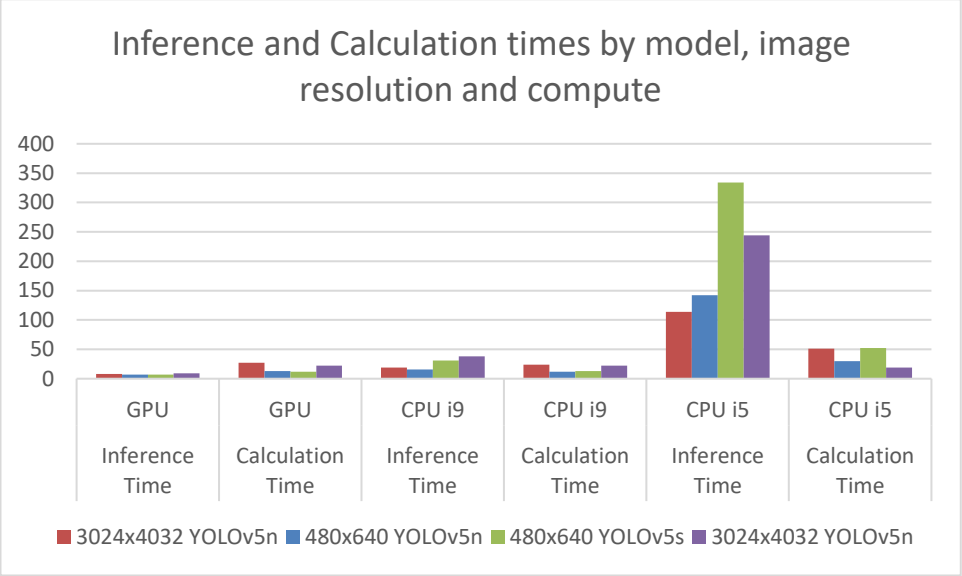



Figure 15 Inference time and Analysis Times by image size, YOLOv5 model size and compute

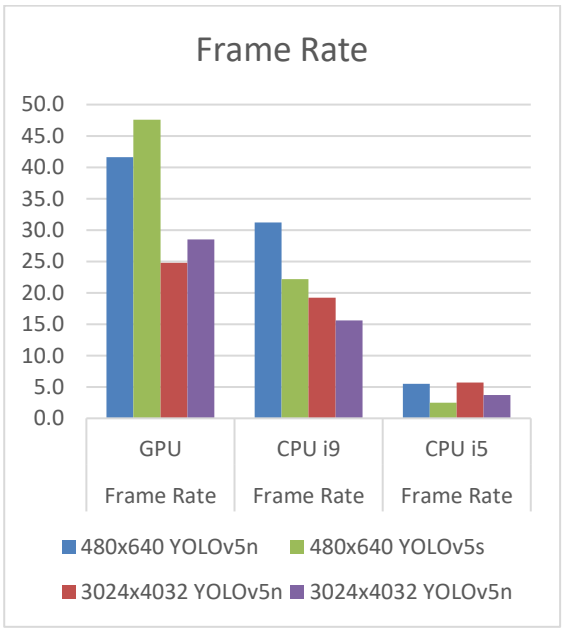



Figure 16 Frame Rate by image size, YOLOv5 model size and compute

| | stem_count | shoot_height | brown_leaf_ratio | root_ball_ratio | RCD | image_name | image_width | image_height | loop_time | tree_type | root_type | pixel_size | orientation | angle | root_pixels | stem_count_min | stem_count_max | shoot_height_min | shoot_height_max | brown_leaf_ratio_mi | brown_leaf_ratio_ma | root_ball_ratio_min | root_ball_ratio_max | RCD_min | RCD_max | num_fails | pass-fail | infer_time | loop time |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 201 | 0 | 17.48148148 | 8.5 | IMG_4032.jpg | 640 | 480 | 0 | pine | root_bare | 0.853448276 | vertical | 357 | 499 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | TRUE | 0.011 | 0.032 |
| 1 | 1 | 192 | 0 | 2.230769231 | 9.3 | IMG_4028.jpg | 640 | 480 | | pine | root_bare | 0.853448276 | vertical | 357 | 42 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | TRUE | 0.007 | 0.025 |
| 2 | 1 | 246 | 0 | 0.015873016 | 12.8 | IMG_4034.jpg | 640 | 480 | | pine | root_bare | 0.916666667 | vertical | 7 | 64 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | TRUE | 0.007 | 0.024 |
| 3 | 1 | 218 | 0 | 0.633254717 | 11.8 | IMG_4014.jpg | 640 | 480 | | pine | root_container | 1.076086957 | vertical | 0 | 4296 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | TRUE | 0.007 | 0.024 |
| 4 | 1 | 228 | 0.001775498 | 0.281456954 | 9.7 | IMG_4037.jpg | 640 | 480 | | pine | root_bare | 0.883928571 | vertical | 3 | 387 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | TRUE | 0.007 | 0.024 |
| 5 | 1 | 166 | 0 | 0.676268412 | 8.6 | IMG_4018.jpg | 640 | 480 | | pine | root_container | 1.076086957 | vertical | 355 | 4132 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | TRUE | 0.007 | 0.024 |
| 6 | 1 | 174 | 0 | 0.494212524 | 7.3 | IMG_4013.jpg | 640 | 480 | | pine | root_container | 0.916666667 | vertical | 2 | 5209 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | TRUE | 0.007 | 0.024 |
| 7 | 1 | 278 | 0 | 0.102564103 | 13.2 | IMG_4025.jpg | 640 | 480 | | pine | root_bare | 0.883928571 | vertical | 2 | 43 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | TRUE | 0.007 | 0.024 |
| 8 | 1 | 194 | 0 | 0.542820803 | 10.1 | IMG_4016.jpg | 640 | 480 | | pine | root_container | 1.125 | vertical | 5 | 3841 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | TRUE | 0.007 | 0.024 |
| 9 | 1 | 306 | 0.000223914 | 3.12 | 14 | IMG_4024.jpg | 640 | 480 | | pine | root_bare | 0.825 | vertical | 2 | 103 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | TRUE | 0.007 | 0.024 |
| 10 | 1 | 198 | 0 | 0.487903226 | 7.6 | IMG_4022.jpg | 640 | 480 | | pine | root_container | 0.951923077 | vertical | 358 | 3872 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | TRUE | 0.007 | 0.023 |
| 11 | 1 | 187 | 0 | 0.529365392 | 7.6 | IMG_4021.jpg | 640 | 480 | | pine | root_container | 0.951923077 | vertical | 3 | 4029 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | TRUE | 0.007 | 0.023 |
| 12 | 1 | 237 | 0 | 0.276553106 | 9.3 | IMG_4029.jpg | 640 | 480 | | pine | root_bare | 0.853448276 | vertical | 9 | 637 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | TRUE | 0.007 | 0.024 |
| 13 | 1 | 267 | 0 | 4.127272727 | 9.1 | IMG_4036.jpg | 640 | 480 | | pine | root_bare | 0.916666667 | vertical | 358 | 564 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | TRUE | 0.007 | 0.024 |
| 14 | 1 | 162 | 0 | 0.538370998 | 9.1 | IMG_4012.jpg | 640 | 480 | | pine | root_container | 0.916666667 | vertical | 359 | 4574 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | TRUE | 0.007 | 0.023 |
| 15 | 1 | 171 | 0 | 0.70032 | 7.5 | IMG_4019.jpg | 640 | 480 | | pine | root_container | 1.076086957 | vertical | 4 | 4377 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | TRUE | 0.007 | 0.023 |
| 16 | 1 | 184 | 0 | 2.666666667 | 8.2 | IMG_4033.jpg | 640 | 480 | | pine | root_bare | 0.825 | vertical | 356 | 264 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | TRUE | 0.007 | 0.023 |
| 17 | 1 | 276 | 0.010775407 | 0 | 11.4 | IMG_4035.jpg | 640 | 480 | | pine | root_bare | 0.951923077 | vertical | 357 | 256 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | TRUE | 0.007 | 0.024 |
| 18 | 1 | 186 | 0 | 0.420437519 | 7.3 | IMG_4023.jpg | 640 | 480 | | pine | root_container | 0.916666667 | vertical | 352 | 4209 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | TRUE | 0.007 | 0.023 |
| 19 | 1 | 214 | 0 | 8.433333333 | 9.7 | IMG_4030.jpg | 640 | 480 | | pine | root_bare | 0.883928571 | vertical | 353 | 849 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | TRUE | 0.007 | 0.024 |
| 20 | 1 | 223 | 0 | 0.612206823 | 10.7 | IMG_4015.jpg | 640 | 480 | | pine | root_container | 1.076086957 | vertical | 4 | 4594 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | TRUE | 0.007 | 0.024 |
| 21 | 1 | 129 | 0 | 0.561440381 | 17.1 | IMG_4011.jpg | 640 | 480 | | pine | root_container | 0.951923077 | vertical | 5 | 4007 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | TRUE | 0.007 | 0.023 |
| 22 | 1 | 118 | 0 | 0.365272282 | 14.8 | IMG_4010.jpg | 640 | 480 | | pine | root_container | 0.99 | vertical | 9 | 3877 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | TRUE | 0.007 | 0.023 |
| 23 | 1 | 183 | 0 | 6 | 8.2 | IMG_4027.jpg | 640 | 480 | | pine | root_bare | 0.825 | vertical | 357 | 21 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | TRUE | 0.007 | 0.023 |
| 24 | 1 | 273 | 0 | 0 | 11.5 | IMG_4026.jpg | 640 | 480 | | pine | root_bare | 0.825 | vertical | 6 | 108 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | TRUE | 0.007 | 0.024 |
| | | | | | | | | | | | | | | | | | | | | | | | | | | min | | 0.007 | 0.023 |
| | | | | | | | | | | | | | | | | | | | | | | | | | | max | | 0.011 | 0.032 |
| | | | | | | | | | | | | | | | | | | | | | | | | | | average | | 0.007 | 0.024 |
| | | | | | | | | | | | | | | | | | | | | | | | | | | frame_rate | | | 41.696 |

Figure 17 Results including measurements, rules and inference times for Mac Book Pro.

# DISCUSSION

Computer vision approaches did not perform well, due to pine needles obscuring other seedlings, branches and also being transparent to the background compared to solid leaves such as a cabbage leaf. Trunks are often obscured by needles and the start of a trunk and start of root ball or leaves is ambiguous even for a human labeller.

DNN-semantic segmentation performed well after relabelling with trunk including some roots in the object definition and training for up to 2000 epochs. Training accuracy was similar for different size models with more epochs required for smaller models to achieve the same accuracy. Note smaller models perform inference faster, but risk missing difficult to detect objects. Nano missed on ‘trunk’ detection at 3024x4032 resolution.  Further software optimisation is possible to improve accuracy, sensitivity and processing time.

Frame rate is dominated by post processing not inference, allowing larger deep learning models to be used for increased accuracy or detection of obscured objects such as trunks behind branches. GPU or Intel i9 processor is sufficient to achieve speed requirements for either 3024x4032 or 640x480 images. Accuracy and sensitivity were not reduced by going to the lower resolution images.

## Future Work

A larger dataset of images in representative situations is required to ensure a validated approach in the future.

Root ratio for bare root could be simplified to left vs right based on position of trunk. Frame rate optimisation between inference and post processing. Machine needed to meet 10 fps is an Intel I9 minimum. Note multi-processing and other code optimisation may assist here with more code development, particularly if each image is saved to hard drive.

The code did not analyse multiple seedlings in the same frame, tracking individual seedlings, communicating with machine to affect the pass/fail mechanical selection mechanism.

# CONCLUSION

It has been shown that seedling identification and grading decisions can be automated using deep learning with variable back grounds. A single camera may be possible depending on bare root grading requirements.

More data is required, and the software proof of concept will need to be made ready for deployment. Speed of operation needs to be confirmed with a camera verses images from the hard drive.

A graphical user interface, GUI, for the operator and data collection to enable future research are recommended next stagers.

The YOLOv5 DNN model performed accurately in different environments and could accurately differentiate tree types between pine and redwood and also root type between bare root and containerised. Camera calibration used a known object (50 cent New Zealand coin) that was circular to avoid camera to object angle differences. Small models that were trained for longer performed at similar accuracy to larger models and had a faster inference speed.

An Intel I9 computer is needed to meet 10 fps minimum seedling per second speed requirement.

# ACKNOWLEDGEMENTS

Thanks to the team that contributed to the definition of requirements and helped make trade-offs to reach an understanding of what mattered. This included Bruce Mennie and Kevin Haine. Also thanks to Kevin for the images of Redwoods and Robert Appleton for assisting with the Pine images.

# REFERENCES

1. Ford, C.M.; SCION *Plant Quality Visual Aids – Bare-root Pinus radiata*; 2020;
2. PlantCV: A Python-based Image Analysis Toolkit for Plant Phenotyping. Available online: https://github.com/danforthcenter/plantcv.
3. Bell, J.; Dee, H.M. Watching plants grow - A position paper on computer vision and Arabidopsis thaliana. *IET Comput. Vis.* **2017**, *11*, 113–121, doi:10.1049/iet-cvi.2016.0127.
4. The open neural network exchange Available online: https://onnx.ai.
5. McGuinness, B.; Duke, M.; Au, C.K.; Lim, S.H. Measuring radiata pine seedling morphological features using a machine vision system. *Comput. Electron. Agric.* **2021**, *189*, doi:10.1016/j.compag.2021.106355.
6. Liu, S.; Wang, Y.; Yang, X.; Lei, B.; Liu, L.; Li, S.X.; Ni, D.; Wang, T. Deep Learning in Medical Ultrasound Analysis: A Review. *Engineering* **2019**, *5*, 261–275, doi:10.1016/j.eng.2018.11.020.
7. Vikram Reddy, E.R.; Thale, S. Pedestrian Detection Using YOLOv5 For Autonomous Driving Applications. *2021 IEEE Transp. Electrif. Conf. ITEC-India 2021* **2021**, doi:10.1109/ITEC-India53713.2021.9932534.
8. Pound, M.P.; Atkinson, J.A.; Townsend, A.J.; Wilson, M.H.; Griffiths, M.; Jackson, A.S.; Bulat, A.; Tzimiropoulos, G.; Wells, D.M.; Murchie, E.H.; et al. Deep machine learning provides state-of-the-art performance in image-based plant phenotyping. *Gigascience* **2017**, *6*, 1–10, doi:10.1093/gigascience/gix083.
9. Chen, Z.; Wu, R.; Lin, Y.; Li, C.; Chen, S.; Yuan, Z.; Chen, S.; Zou, X. Plant Disease Recognition Model Based on Improved YOLOv5. *Agronomy* **2022**, *12*, doi:10.3390/agronomy12020365.
10. Mahendrakar, T.; Ekblad, A.; Fischer, N.; White, R.; Wilde, M.; Kish, B.; Silver, I. Performance Study of YOLOv5 and Faster R-CNN for Autonomous Navigation around Non-Cooperative Targets. *IEEE Aerosp. Conf. Proc.* **2022**, *2022-March*, 1–12, doi:10.1109/AERO53065.2022.9843537.
11. Zhang, J.; Xu, C. Research on The Identification of Benign and Malignant Thyroid Nodule Ultrasound Images Based on Deep Learning Model. In Proceedings of the 2022 IEEE Asia-Pacific Conference on Image Processing, Electronics and Computers, IPEC 2022; 2022; pp. 904–907.