Computer Vision for Yarding Operations

Evaluating Single-Stage Instance Segmentation for Grapple Carriage Yarding

Frazer French | ENFO410 | 30/10/2022

ABSTRACT

This report investigates the application of recent machine learning techniques for computer vision in the forestry industry, specifically grapple yarding. Increased computing power of embedded systems, coupled with improvements in open-source software have resulted in these techniques being applied in a wide range of sectors for real-time computer vision. The research aims to provide guidance for manufacturers considering automation of the grapple carriage, as to whether current open-source software is fast enough and accurate enough for a real-time application. The software being tested is a single-stage instance segmentation algorithm known as YOLACT (You Only Look At CoefficienTs) and is being assessed for its performance in detecting logs in the footage generated by the grapple camera.

A review of the current applications of computer vision applications in forestry is presented, and a review of the mechanisms of the algorithm being tested for application. The YOLACT algorithm is an open-source machine learning model that can be trained to recognise objects within an image or video. Other applications of YOLACT, as well as the original study, have achieved accuracy in detecting objects >95% and at speeds above 30FPS (Frames per Second), which is sufficient for a real-time vision application. YOLACT is an instance segmentation algorithm, which produces masks to identify the outline of the object being detected.

Two datasets were produced for this study. The first was generated from footage from a grapple yarding operation in Nelson, NZ, from a Falcon Claw grapple camera. The image quality was 720x408p, with approximately 8 hours of video collected. The second was generated from a smaller segment of footage from a grapple yarding operation in Canterbury, NZ, using a DJI drone recording at 1920x1080p. Stills were generated from the footage, and the logs in the images were masked with the Labelstudio tool. The first dataset contained 233 instances of logs to be grappled, and 213 in the second.

The YOLACT model was trained on each dataset, for 2000 iterations. with each dataset producing a trained model for detecting logs. Initial weights from RESNET-50 were used to reduce training time and decrease the effects of using a small dataset. The computer hardware used for evaluation was a GeForce RTX 2070 8GB RAM GPU, which has comparable computing power similar to currently available hardware for embedded machine learning architecture. After training was completed, the trained model was evaluated on still images excluded from the original datasets, to generate insight on the model's ability to predict.

Overall predictive accuracy for the model was poor, achieving a mask AP (0.5) for the grapple camera footage of 12.8%, with the performance dropping significantly to 1.1% at AP (0.9). The higher quality footage obtained by drone achieved a mask AP of 21.6%, although performance reduced to 0% at AP (0.9). Both trained models achieved high speeds for inference during evaluation, with the low-quality footage averaging 50.2FPS processing speed, and the higher quality footage achieving 31.5FPS.

Inferences made by the trained model show difficulty identifying entire logs, and frequently partitions a single log as two or more separate instances. Other issues include misidentification of the claw as a log due to some components having similar shape and colour, and missing logs that should otherwise be identified. Both models had similar issues, although the general accuracy of the higher quality data was superior, with a small reduction in the speed performance. For a robust commercial application, a larger dataset of high-quality images would be required.

1 TABLE OF CONTENTS

2 INTRODUCTION

Extraction of felled logs from the cutover is a critical part of the harvesting process and forestry business model. Logs that have been felled in the forest block need to be returned to a landing or processing location, where they can be cut into logs and transported out of the forest. There are several methods for this process, depending on the terrain being logged, the availability of capital and operation size. For harvest areas with steep slopes (>40%), cable yarding is generally the preferred system for extraction. This system utilises a choker, a claw or a grapple device to hold the logs, with a carriage or cable system used to pull the logs back to the landing [1].

Cable yarding was introduced into New Zealand in the 1950s and has since been the preferred method for extraction where slope steepness prevents ground-based harvesting methods [2]. Cable yarding has other benefits over ground-based harvesting systems, such as reduced soil disturbance, reduced need for earthworks and roading, and reduced machinery travel across steep slopes. [3,4]. Historically, logs were attached to the cable system by means of a choker, with a person on the ground required to manually complete this task, as seen in Figure 1.

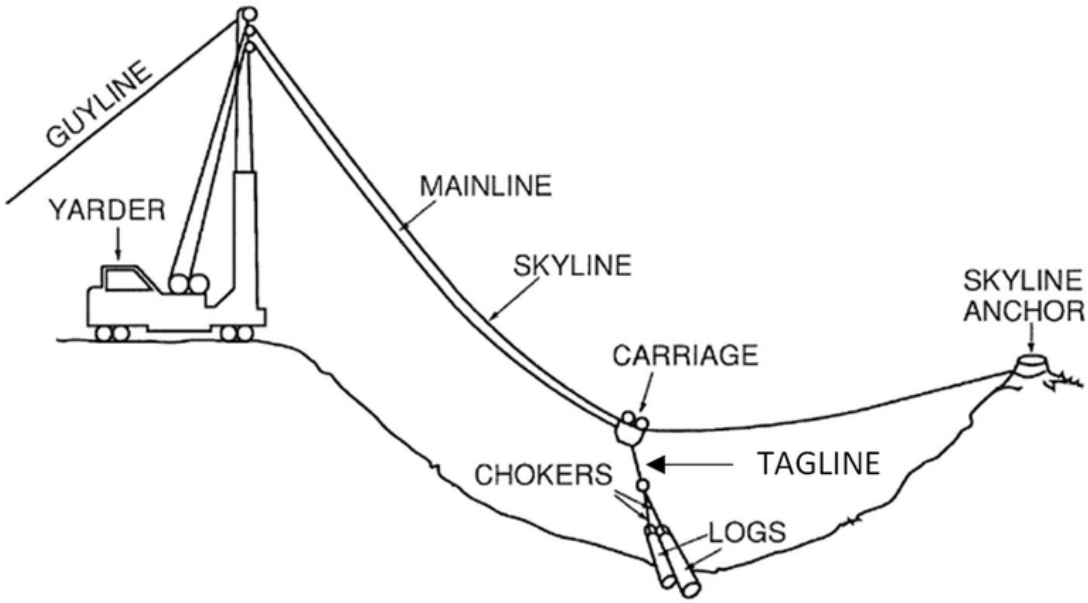

Figure 1 - Basic cable yarding setup. Source: [5]

In recent years, operations have become increasingly mechanized, driven by the need for improved health and safety outcomes for the workforce performing this task, and to improve productivity and value recovery. Grapple carriages avoid the need for workers to be on the ground and reducing this component of the workforce was a key outcome of the NZ steep slope harvesting initiative that ended in 2017 [6]. Surveys of preferred rigging configurations for cable harvesting operations in New Zealand show a trend towards using mechanised and motorised grapple systems to extract timber, with mechanical grapple being the preferred rigging configuration for 38% of contractors in 2018, with a shotgun motorised carriage second most popular at 18%. Nearly 50% of crews in New Zealand are utilising some form of grapple yarding as of 2018, which has sharply increased from 4% in 2011. [7]

Grapple yarding has four stages to the workflow. It begins with the outhaul step, where the carriage leaves the yard and travels out along the cable to the next log to be grappled. Next is the grappling

step, where a log is located and grasped in the claws of the grapple. The return step follows, where the grasped log is dragged back to the yard, and the release, where the log is placed in the yard, before returning to grapple the next log [8]. Other machines may also be present in this process, with a shovelling machine assisting with the grapple step by presenting logs to the claw, and a shovelling machine receiving these logs at the yard. These systems commonly have a camera that feeds a birds-eye view of the underside of the grapple to the operator of the yarder, who uses this information to control the grapple and grasp the logs that have been felled. This footage comes in a range of resolutions and views. Some systems integrate lighting into the system to illuminate the scene and make night-time operation possible.

Figure 2 - Cable yarder operator and controls. Source [2]

Steep slopes forestry blocks are a significant proportion of the forest for harvest in New Zealand and will make up an estimated 60% of the annual harvest for New Zealand by 2025 [9]. As such, there will be a continued need for grapple yarding, and the skilled operators required to operate the machinery. Demand for these workers is high, and the forestry industry and the global labour force are facing challenges. A 2021 survey of the NZ forestry labour force highlighted that most firms were experiencing shortages of machine operators or expected to encounter labour shortages in future [10]. The survey also identified that forestry firms expect to continue to have issues filling positions, especially in experienced roles. To attract more people to the workforce, jobs need to be seen as more comfortable, safe and fulfilling. Experienced operators are demonstrably more productive than inexperienced operators, and as such reducing the time taken to train an operator to an experienced level will only improve the productivity of the work [11]. In addition to this, machine operators are typically a bottleneck in forestry operations [12].

Automation, or partial automation, is a potential pathway for improving both employee satisfaction and improving the lifetimes of machines [13]. Partial automation is the automation of some

component or task of a workflow or job description. This can improve the quality of work by reducing the cognitive load on machine operators, by the removal of menial and repetitive tasks, as well as decreasing the time taken to train operators. This may have an associated benefit of making these roles more attractive and therefore encouraging more people to take on these roles. In addition to making jobs more attractive, partial automation may allow for a harvesting system where a single operator manages more than one machine. As an operator may be 30-40 percent of the operating cost of a machine, this may be a large cost saving for forest harvesting operations. [14].

Automation of machinery for forestry has made some progress, with machines already having some autonomous capabilities. In the domain of yarding, automation or partial automation is currently operational for two phases of the cycle, the outward-haul and in-haul phases of the extraction cycle. [15]. In addition, some tower yarders can be remotely controlled, such as the Konrad KMS 12U and the Valentini V1500. Studies into the automation of yarders have shown improved productivity in harvest locations where the yarding distance is below 200m, even where no crew had been removed due to the automation. [16]. Other applications of computer vision for specific forestry applications can be found in the literature review of this report.

The next step towards automated yarding using a grapple is to automate the detection and grappling of logs. Automation of yarding has been identified in [17] as the most achievable automation step for forestry in the near term. This study intends to test currently available machine vision algorithms. This will determine whether it is possible to integrate an automated vision system in a grapple yarding system, and whether this can be implemented at a speed adequate for assisting or automating part of the yarding task in a harvesting operation.

3 LITERATURE REVIEW

3.1 COMPUTER VISION OVERVIEW

Computer vision is a broad discipline that uses algorithms to interpret data from images and video to make interpretations and provide information on the scene. In a machinery context, this is to provide information for the machine controls and guide mechanical outputs.

Computer vision is typically one of four tasks. The simplest is image classification, in which an image is labelled with a type or category. Object detection is more complex, where an object in the image is both classified and located by a box. Semantic segmentation is a process in which each pixel in an image is given a label, to accurately position the target in the image. The most complex of these operations, instance segmentation, segments instances of the same type of object in the same image, and labels them as individual objects. Instance segmentation is the task being attempted in this method, as autonomous operation will require exact positioning data to provide guidance to the grapple carriage, and the ability to discern between different logs in that image [18].

These tasks can be approached with two methods of implementation. Traditional computer vision uses explicit mathematical algorithms in sequence to pick features from an image, such as edges, colors or brightness, and determine their location and classify them accordingly. This method is well suited to tasks with adequate illumination, consistent lighting and clearly defined objects [19].

In comparison, deep learning, a subset of artificial intelligence, can be used to achieve the same task. A deep learning model extracts features and context from an image by training it on a large dataset of images where the target object is labelled. Through a complex process of feature extraction, where the algorithm learns through an optimisation process the features common to the target object, it can determine where the target object is with a high level of accuracy. Given its increased complexity, this method is more computationally expensive than traditional methods [20], but is better able to understand context, varied lighting, and variable orientations than traditional approaches. It is for this reason this study investigates machine learning methods, as the cutover is a complex environment with a range of lighting conditions.

As determining the location and position of the log is important, instance segmentation will be the task attempted in this research. Instance segmentation can be performed through one of three methods; detection based, pixel based, or a single-stage methods. Single-stage methods have the greatest computational efficiency and can achieve real-time inference results with current processing power. For this reason, they will be evaluated for this method as real-time inference is a critical component of a vision system for the grapple camera.

3.2 COMPUTER VISION IN FORESTRY

While forestry has used computer vision for years in the sawmill sector, recent advances have capitalised on the improvements in machine learning and increase in computing power of embedded systems and have attempted or investigated detection in the forest environment.

Fortin et al. [18] investigates multi-stage instance segmentation of logs for forwarders and grasping applications on landings. The results did not provide the level of accuracy required for automation (AP 57.53%), but the results indicated that with a larger dataset and more processing power, there is potential for assistance to operators or full automation in the future. The study differs to the proposed method as it did not test single-stage segmentation techniques and thus did not achieve a real- time frame rate, with the fastest method tested achieving 12.3 frames per second. The pose of the logs was also significantly different, as well as the aspect ratios of the logs in the dataset. The dataset comprised of 220 images, containing 2500 log instances.

Instance segmentation of fallen timber has also been achieved by Polewski et al. [22] with a high level of accuracy. The run-time for this algorithm however is in the range of seconds-minutes, which is not suitable for a real time application. The precision achieved in this application was 82%-91%, with a recall of 70%-79%. This method incorporated imagery from the near infra-red channel, which improves accuracy significantly. However, the improvement accuracy comes with a corresponding drop in processing speed.

Real-time object detection has been achieved in the forest environment, with a YOLO object detection framework being implemented to detect obstacles for a machine navigating the forest [23]. This operation had realistic lighting settings, but the objects were simple, such as rocks or stumps, and were well defined in the setting. This method used a relatively small dataset of 200 training and 75 validation images, however, only achieved a frame rate of 7.4FPS. Felled logs pose a more difficult machine vision challenge, as they may not fall fully in the extent of the camera view, and thus distinguishing its edges is not straightforward.

P. Han [24] investigated the use of object detection in the forest environment, to identify logs for measuring sweep. The study claimed a detection success of over 90%, although this was not strictly

quantified in the results section. The paper utilised Mask- RCNN for the training, but the study gave no guidance as to the real-time capability of the method. The method did show the value of using high quality images, as it achieved quality performance from a small number of training samples, and the video quality that the data was derived from was high.

Detection of standing trees was achieved in [25] by da Silva et al., evaluating the accuracy of several machine learning algorithms. Whilst this study investigated object detection, as standing trees have a known pose that allows the co-ordinates of the tree to be understood more easily than a felled log, high accuracy was achieved (90%) at an inference speed of 12.5FPS. This method had a very large (2895 images), high quality dataset that also used image augmentation to improve the accuracy of the model. Image augmentation alters the images slightly to provide the algorithm more training data, with images that are still relevant to the end use, either by rotation, zoom, flipping or blurring.

These methods all have achieved computer vision in the forest environment, although real-time instance segmentation, especially in the context of the grapple camera, is yet to be explored. Of the open-source real-time instance segmentation frameworks available, YOLACT is the most implemented of these [26]. For this reason, alongside the significant success of YOLO networks in the reviewed research, it will be explored as a potential method for achieving real-time instance segmentation with this task.

3.3 YOLACT ALGORITHM

3.3.1 YOLACT (You Only Look At CoefficienTs)

YOLACT [27] is a real time instance segmentation algorithm. It was developed as the first single-stage instance segmentation algorithm, designed to be faster than existing instance segmentation algorithms, such as Mask-RCNN. The paper achieved 29.8 mAP on the benchmark COCO (Common Object in Context) dataset, at a frame rate of 33.5FPS. For an explanation of AP (average precision), this has been outlined in section 5.4 of this report.. YOLACT has been successfully implemented in multiple research papers since its development, with results exceeding 95% mAP [28] with a high-quality dataset for traffic signs, and mAP of 32.44% in an application of crack identification in concrete [29]. The YOLACT algorithm is a single stage instance segmentation model, which takes an object detection framework and then adds a mask branch to the process to perform instance segmentation, as can be seen in figure 3. YOLACT's feature backbone utilises the RESNET-101 architecture and adds a feature pyramid network. The YOLACT architecture is a parallel computation of two separate tasks, to produce segmentation maps. The first task is an object detection head that detects objects in the image and predicts mask coefficients for each anchor point derived from the feature pyramid. The second task in the segmentation branch is generating a dictionary of prototype masks over the entire image via a fully convolutional network. The outputs of these two objects are combined to generate masks for the instances in the image. By using convolutional layers in task 1 to produce spatially coherent masks and fully connected layers in task 2, which produce semantic vectors, the overall instance segmentation can be completed efficiently to give real-time results.

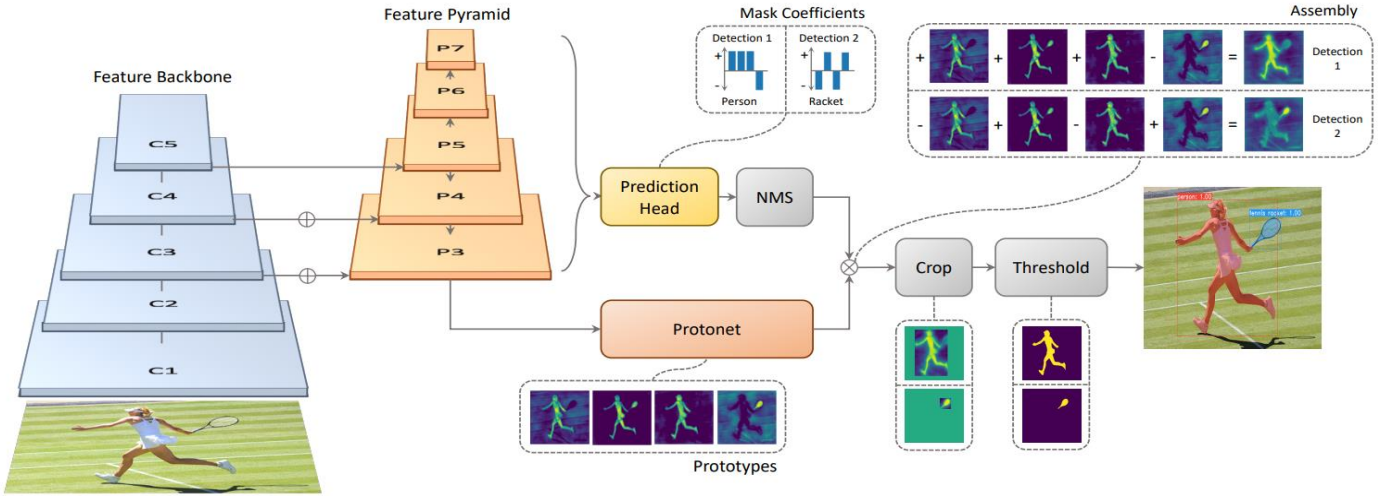

Figure 3 - YOLACT architecture, Source: [27]

3.3.2 Object Detection

Object detection occurs in the feature backbone and feature pyramid of the YOLACT architecture, and one of the YOLACT branches. The feature backbone is RESNET-101, a convolutional neural network (CNN) with 101 layers. There are three types of layers in a CNN, convolutions, pooling and fully connected or non-linear layers. The overall network takes an image, extracts features and attempts to infer the class of the image, or objects within the image [30]. This process is iterative, as the network compares its estimations with training data and adjusts its weights according to the labelled data and repeats the process. Once trained, the network can be used to inference new images from outside the training set for detection applications.

3.3.3 Convolutions

Convolutional layers are layers within the CNN for extracting features from an image. Convolution layers pass a kernel over the image, with a given set of weights, to produce a new image that emphasises certain features, such as edges or corners. These layers are stacked in series to create a network, which allows for the model to make inferences about complex features such as texture and relationships between features. Figure 4 shows a basic convolution process using a kernel.

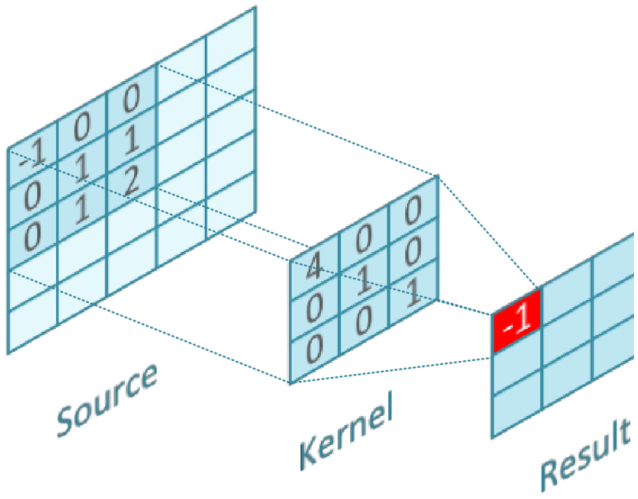

Figure 4 - Convolution layer, Source: [30]

3.3.4 Pooling

Pooling layers take an image, or matrix of values, and reduces the size of that matrix, taking the maximum, minimum, average or any number of complex functions of the pooled area. The pooling step reduces the importance of location of features within the image, by creating down-sampled feature maps [31]. These feature maps can give a summary of the presence of features in a map and reduce the image size, improving computational efficiency. Two different possible pooling functions, maximum pooling and average pooling, are shown in figure 5, demonstrating the downsizing of the input to a pooled output.

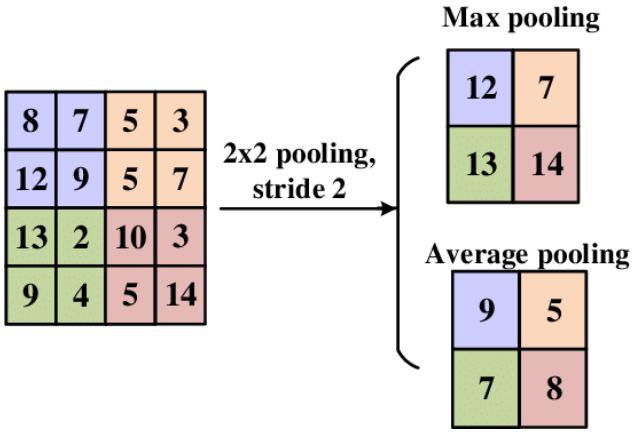

Figure 5 - Pooling layer, Source: [31]

3.3.5 Fully Connected Layers

Fully connected layers are typically the final layer in a CNN. Fully connected layers connect the outputs of previous layers and passes them through a weight matrix, which produces an output vector [32]. The output vector can be used to compare against the training data set, and the weights used to compute the vector updated based on the result. Figure 6 shows a simple diagram illustrating an FCN, where each of the final neurons is connected to all the neurons before it.

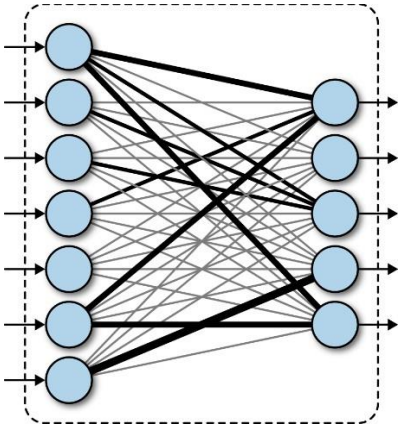

Figure 6 - Fully Connected Layer, Source: [32]

Object detection is utilised again in the YOLACT method in the mask coefficient generation step. The YOLACT object detection here has 3 components, a predictor for each class of object c, 4 coefficients for a bounding box and k coefficients for the mask. The architecture for this step is shown in figure 7.

The additional mask coefficient provides the YOLACT model a coefficient for each mask prototype produced, to compare with training data to improve segmentation.

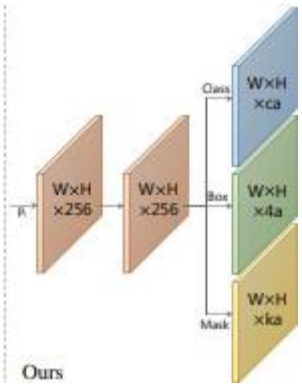

Figure 7 - YOLACT object detection head, Source: [27]

3.3.6 Fully Convolutional Networks (FCN)

Fully convolutional networks are a class of neural network that can be used for segmentation tasks [33]. They are used in second branch of the YOLACT algorithm, to generate prototype masks. An input, typically an image, is passed into the network in the form of an RGB image of size n*n*3. The FCN is comprised of layers, performing pooling, activation, or convolution functions, of which none are fully connected. These layers reduce an image in an encoding process, in which each pixel is labelled based on the context of the image, and then resize the resulting feature into an image of the same size of the input [34]. This resized image contains a mask of objects in the image that have been classified according to the model training. The YOLACT model creates prototype masks via this process to output to the final classification step.

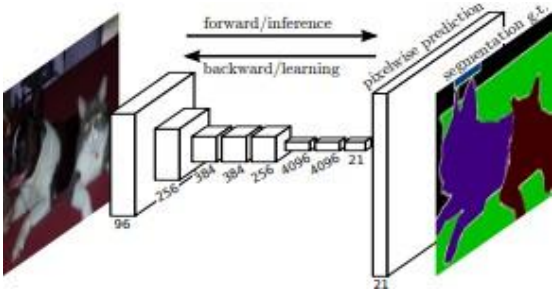

Figure 8 - Fully convolutional network and masked output, Source: [33]

3.3.7 Non-Maximum Suppression

Non maximum suppression is a part of the YOLACT framework, which selects the most appropriate mask for the object detected. After the mask proposals are generated by the FCN, and objects detected within them, the proposals are eliminated by evaluating the intersection over union for each proposed bounding box [35]. Starting with the highest confidence mask, intersection over union for each mask is evaluated, removing lower confidence prototypes and leaving only the instances with the highest confidences. The algorithm will produce one mask for each object detected, with the most confident box being selected, which thresholding and cropping is applied to, generating the output of the model.

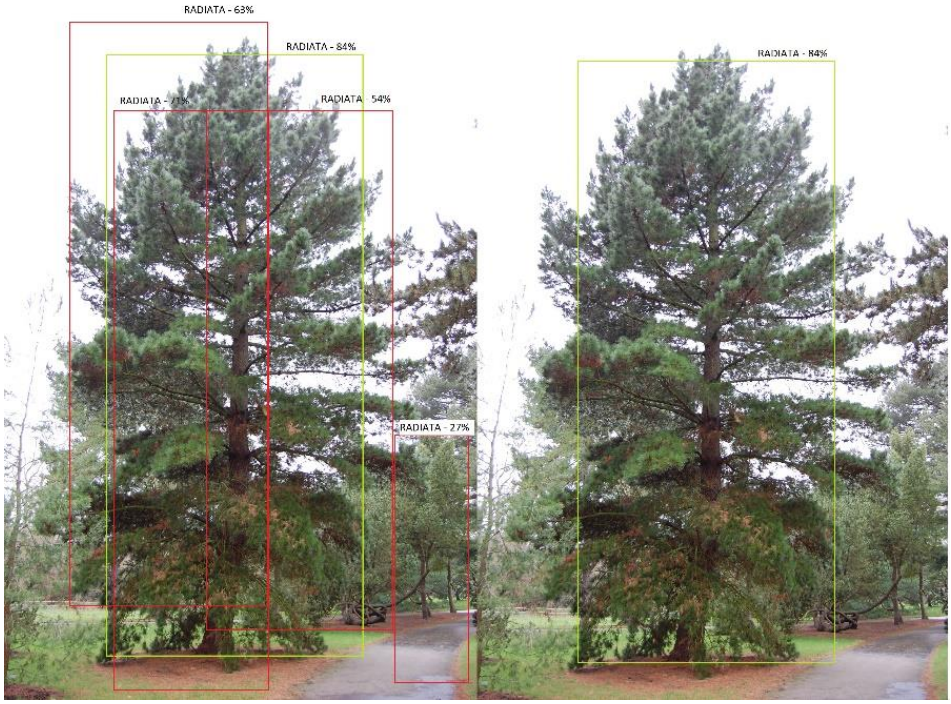

Figure 9 - Non-maximum suppression of pine trees, Source: [36]

3.3.8 Transfer Learning

Transfer learning is a technique used in machine learning applications to use weights from other datasets to assist with inference on new objects, which assists the training with detecting features common to many objects [37]. This reduces training time for new objects and has been implemented in this method. The proposed method uses the RESNET-50 YOLACT weights generated from the COCO (Common Object in COntext) dataset. RESNET-50 weights gave the best results for mAP in the original YOLACT implementation [27], and have been utilised in this method for processing the images from the harvest site.

4 OBJECTIVES

The primary objective of this study is to evaluate a currently available instance segmentation algorithm for use in grapple yarding operations. Evaluating this software will provide guidance as to whether an autonomous yarding system could be built implementing this technology. Instance segmentation has been chosen as the proposed method, as an autonomous yarding system would require the position and orientation of the log being detected. Instance segmentation can provide this information to a system via the mask output, whereas a bounding box in an object detection algorithm cannot. Due to the number of potential logs in each frame, each instance needs to be identified, therefore semantic segmentation will not be adequate for the task. The YOLACT algorithm has been selected for its speed performance in other applications, which makes it the most likely method to achieve real-time detection for this task with available hardware.

The criteria for this assessment will be average precision (AP), a metric used in evaluating the accuracy of predictions of machine vision systems. This measure is defined in the method section of this report. The task of the algorithm is to identify and mask the outline of a "target log." A target log for this study has been defined as:

-A log that is at least 50% visible along its length, within the frame of the image

-Not already held in the grapple claw, where a grapple claw is present

-Is in the cutover, as opposed to on the landing

A target log may be partially obscured by other logs, or by foliage, as this is realistic to expect in a cutover. As only logs that are lying in the cutover need to be detected for grappling, there is no need for the algorithm to detect logs already grappled/when the grapple is closed, or logs on the landing. However, it is likely that the algorithm would still be capable of detecting these.

In addition to the accuracy of the model, the speed will be assessed. Frame rate required for real-time operation in machines is typically in the range of 15-30FPS [38], and an AP typically above 90% depending on the application and level of automation being designed for. This FPS will be assessed on a hardware system comparable to currently available hardware for embedded systems.

In addition, the two performance variables will be assessed on footage of different video quality. This will provide information on what level of data quality will be necessary to build a functional automated yarding system, and how the video resolution influences the speed of inference.

A secondary objective of this study is to build several high-quality, annotated datasets for log identification. These may be used by researchers to supplement machine learning training in future research, or to progress the development of a commercial log detection software. These datasets will be made available on GitHub as a repository for future projects to download and access.

5 METHODS

5.1 DATASET CONSTRUCTION

5.1.1 Dataset A – D.C. Equipment footage

The dataset was constructed from four segments of footage from a grapple yarding operation in the Nelson region of New Zealand, all approximately two hours long, and one shorter, half-hour segment. The footage shows the view from the grapple camera, which is a birds-eye view of felled logs on the cutover. The footage collected is from the New Zealand summer period, where outdoor lighting is strong and consistent, on days where the weather was fine and sunny. The 5 separate pieces of footage are from the same forest block. Due to the duration of the footage, lighting conditions do change, with shadows lengthening and shortening due to the angle of the sun. However, as the footage is taken primarily from midday, the appearance of the logs is relatively consistent, especially in colour. This can vary considerably in sunset and sunrise conditions. The video footage is from D.C. equipment, with the footage being captured by a DMAC-LCP Ag-Cam installed on a Falcon Claw grapple. The footage is recorded at 1280x720p, however the capture card used to record the footage has a maximum output of 704x480p, hence this footage is down sampled to 704p in this study.

Figure 10 – Sample images from D.C. dataset (pre-annotation)

To create the sample used for training the selected model, the video footage was separated into still images using VLC media player. This method conserves the aspect ratio and quality of the images provided in the video footage; hence these will be generated at 704x480p. Stills were generated at 2 second intervals from the footage.

As a significant portion of the footage does not contain targets to be annotated for network training, only a selection of the stills will be used for the model training. A selection of 100 images have been chosen for the annotation process based on the following criteria.

-Image is clear/not blurry

-Contains a clear view of a target log/s

-Logs are well lit by the sun

-Grapple is at the cutover, as opposed to the yard

Although the same log may be present in multiple images included in the training dataset, all images are unique. The same log, but in a different part of the frame or in a different context, for example having a nearby log removed, may appear multiple times in the training dataset. It still has use in training the model, as it gives the model understanding of the different positions and contexts the same log may be found in. Further simplification of the model has been achieved by only including scenes where the grapple is travelling over or lowering to a target log. The dataset therefore excludes images of logs at the landing, as well as images where the grapple is in its return cycle. This is appropriate, as an automated yarding system would only require log detection on the outgoing part of the yarding cycle, as once it has grappled a log, it needs only to return to the landing. A further 12 images were selected from the stills to be used in model inference, that were not annotated.

5.1.2 Dataset B – Rayonier Drone footage

The second dataset was generated from footage taken from a DJI done in a forest in North Canterbury. The total period of footage used to generate stills is approximately 5 minutes of flying footage, but is feature-rich, and therefore contains enough logs to generate a dataset for training. The stills are generated at 2 second intervals, with the footage at 1920x1080p. Conditions in the forest were overcast, and taken at midday in September, early spring. Despite the weather, the logs in the images are well lit and easily distinguishable from the surroundings. Two segments of footage were from the live harvesting operation, and two from a selection of piled logs just aside from the harvesting operation. As this footage is from a drone, the height above the cutover varies due to manual operation, and in general is higher than would be expected from a grapple camera. As such, there are typically more logs in each image. The freshly harvested trees have a large amount of foliage included, with the older stems being more visible along the trunk, as can be seen in figure 11. The images were subjected to the same criteria as the D.C. imagery. 5 additional images were selected from the dataset to be used for model inference.

Figure 11 – Sample images from D.C. dataset (pre-annotation)

5.2 Annotation

After creating the dataset of images, the images were annotated using the Labelstudio tool. This open-source tool allows a user to define the edges of log instances, to mask a log in the cutover. In images where multiple logs are present, they are each annotated as a single instance, as opposed to labelling them as groups or piles of logs. Labelstudio annotates the COCO format as a .JSON file, which can be interpreted by the algorithm for training.

An additional 12 images from the DC dataset, as well as 5 from the Rayonier dataset, were selected for use in qualitative inference in the results, to assess how the model views the data from a human

perspective and to measure the speed. The speed is measured by providing the trained model with a video to inference on and averaging the speed of inference over the process in a Python script in Jupyter Notebook. The code for this step can be viewed in the GitHub repository.

5.3 Network Training

The YOLACT algorithm is provided with a set of base weights to initialise the model. These allow the algorithm to take the learning from other object detection tasks before fine tuning to the specific task of identifying logs. The RESNET-50 weights have been utilised in this study, as these performed the best in the initial YOLACT paper [27]. RESNET-50 weights are from the RESNET architecture trained on the ImageNet database, a collection of nearly 15 million annotated images. These weights are not forestry-specific, but provide the YOLACT algorithm with some understanding of what general physical objects look like and their properties. This is known as transfer learning and is an industry standard practice for deep learning in computer vision applications. It has the benefit of reducing the size of the dataset required to train the model [39].

As the model takes images of any size, no additional pre-scaling or data adjustment was required for the dataset before passing to the model. Batch size used for training was 8, with the model trained for 2000 iterations. The learning rate initialises at $1x10^{-3}$ and decreases linearly by a factor of 10 during training at steps 1200, 1600 and 1800. The YOLACT algorithm has built in testing to prevent it from over-training or generalising to the target images, with the iterations specified will be sufficient to fully train the model.

To train the network, the annotated dataset is split into training and test data. The model will “learn” what a log is based on the training data, and then test these assumptions on the test data, which gives it feedback as to whether its guess was correct, allowing it to iterate and improve. Industry standard practice is to provide 80% of the data to training and the remaining 20% to test, which has been employed in this study.

Training and evaluation will be completed on a machine using a GeForce RTX 2070 8GB RAM GPU with CUDA 11.4, with an Intel Core i7-8700 6-thread CPU. Modifications to the code and adaptation for this application in Python was via the Atom IDE on the Linux Mint 20.3 OS. The specifications of this computer system is comparable with currently available rugged embedded computing, such as the NVIDIA AGX Xavier Industrial. Measuring the computing power of AI systems is computed in FLOPS [40], with the GeForce RTX 2070 having 7.5 TFLOPS of power, and the NVIDIA Xavier achieving approximately 11 TFLOPS [41].

5.4 Measurement

The results are interpreted in terms of average precision (AP), which is a common metric for evaluating machine vision applications. Average precision combines several identification metrics into a score that can be used to evaluate the overall performance of the model.

Precision is the primary component of the AP score, and is calculated as follows. In this study, a true positive is the algorithm predicting a log where one exists, and a false positive is a log is predicted that does not exist.

$$Precision = True\ Positives/(True\ Positives + False\ Positives)$$

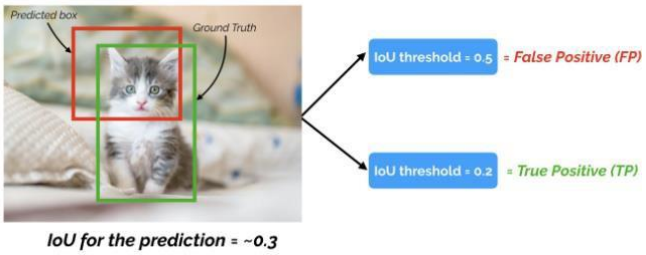

Figure 12 - Calculation of precision, Source: [42]

As there will be some discrepancy between the mask produced by the annotation and the predicted mask of the algorithm, a threshold must be identified. The measure for this threshold is intersection over union, which is calculated as seen in figure 13 below. It quantifies the amount of overlap between two boxes, or masks as are used in this application.

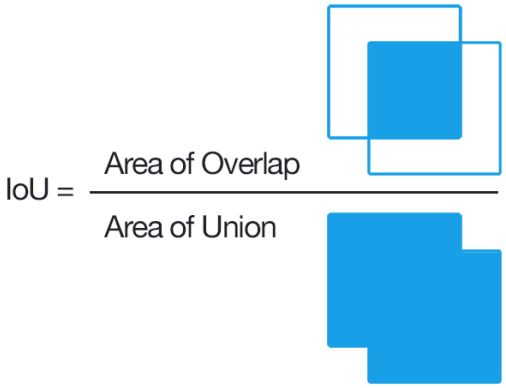

Figure 13 - Calculation of Intersection over Union. Source: [42]

For this study, consistent with [18], an IoU of 0.5 has been set as the threshold for correctly identifying a log. Although in practice this may be less that the accuracy required to pick up a log, as the grapple carriage nears the target log, it will have an increasingly detailed view and therefore be able to refine its recognition of the log.

As precision on its own does not account for incorrect predictions, the recall metric is also calculated. Recall is calculated as

$$Recall = True\ Positives/\ (True\ Positives + False\ Negatives)$$

A false negative is registered when an annotation provided in the training dataset is not identified by the algorithm. This represents a log that has not been identified on the cutover and would pose a problem for a working prototype, as logs would be left on the cutover.

Average precision is then calculated as the area under the precision-recall curve. AP is commonly evaluated at a range of differing IoU thresholds, which will be presented in the results section, with a high value being indicative of a robust and accurate vision application [42]. Mean average precision is used when multiple classes of objects are being detected, however as there is only one class in this study, average precision is used.

6 RESULTS

6.1 RESULTS OF DATASET CONSTRUCTION

6.1.1 D.C. Dataset

The dataset creation yielded a total of 16563 images. The training dataset was constructed from these images, with 112 images selected from the stills. Stills from all five video segments were used to ensure that the samples were representative of slight variations in conditions between the videos, and that the model will be able to inference on logs.

6.1.2 Rayonier Dataset

The dataset creation for the Rayonier data yielded 165 images. 27 of these were selected for annotation, with a further 5 kept for inference.

6.2 RESULTS OF ANNOTATION

100 images were successfully labelled from the stills created in 6.1.1. In the 100 images for the D.C. dataset, 233 log instances were masked. These annotations were split 80:20 into a training and validation set using the open source Cocosplit tool, and used for network training. The 27 images in the Rayonier dataset contained 213 log instances that were annotated. The data is available online in the GitHub repository. Figures 14 and 15 show examples of annotation in the Labelstudio tool.

Figure 14 – D.C. dataset images, before and after annotation

Figure 15 – D.C. dataset images, before and after annotation

6.3 Results of Network Training

6.3.1 D.C. Equipment footage

6.3.1.1 Quantitative Results

The training was able to train the model to achieve an AP (0.5) of 12.76%. As a comparative measure, the algorithm evaluation also computes a value for the accuracy of a bounding box. This was also relatively low, achieving 17.2 %. As can be expected, the model performance decreases significantly as the IoU threshold decreases, illustrating the difficulty in producing masks of logs at higher levels of precision.

IoU Threshold	***0.5***	***0.6***	***0.7***	***0.8***	***0.9***
Box (Average Precision)	17.2	11.1	6.2	2.7	1.1
Mask (Average Precision)	12.8	8.6	4.9	1.5	0.0

Table 1 - Quantitative Model Performance, D.C.

The speed of the network was evaluated by taking the raw data from the model evaluation script with test video footage in Python. The average processing over the evaluation of the video was 50.2 FPS. Raw data is available in the GitHub repository.

6.3.1.2 Qualitative Results

Figure 16 below shows inferences on 12 images by the trained model. These images came from the stills generated for the annotation dataset but were withheld to show how the model performs on data it has not seen previously. Each instance is identified by a box, with the extent of the instance colored where the algorithm believes the target object is. The number in the corner of each bounding box shows the algorithm confidence in its prediction. The threshold for this inference has been set at 0.3 to maximise the information that can be gleaned about the workings of the model.

In several of the inferences, 1,3,5,6,8 and 12, no log is identified by the algorithm. Although the training dataset contains several instances where vertical logs are present, although partially occluded by the edge of the frame and the grapple itself, these logs do not appear to be picked up by the machine. This is of particular concern as this is an orientation that logs are likely to be presented at to the grapple, and as such would not be detected and therefore yarded in an automation application. As the machine learning algorithm is somewhat of a "black box," it is difficult to infer what is causing the issue, but it is likely that the target object beginning or ending outside of the image make it difficult for the algorithm to classify correctly. This is discussed further in the discussion, section 7.

There are also several examples of the algorithm only detecting part of a log, specifically inferences 2, 7 and 10 where only part of a log are identified as an entire instance. This would also pose issues to a potential application of this software, as the position of the log provided to the yarder would be incorrect, resulting in the log being either picked up in the wrong spot, creating difficulties in returning the log to the landing. In addition, inference 10 identifies one log as two separate instances. This would cause similar problems as identified above, with grappling and yarding the log difficult where the machine does not understand where the true centre of mass of the log is.

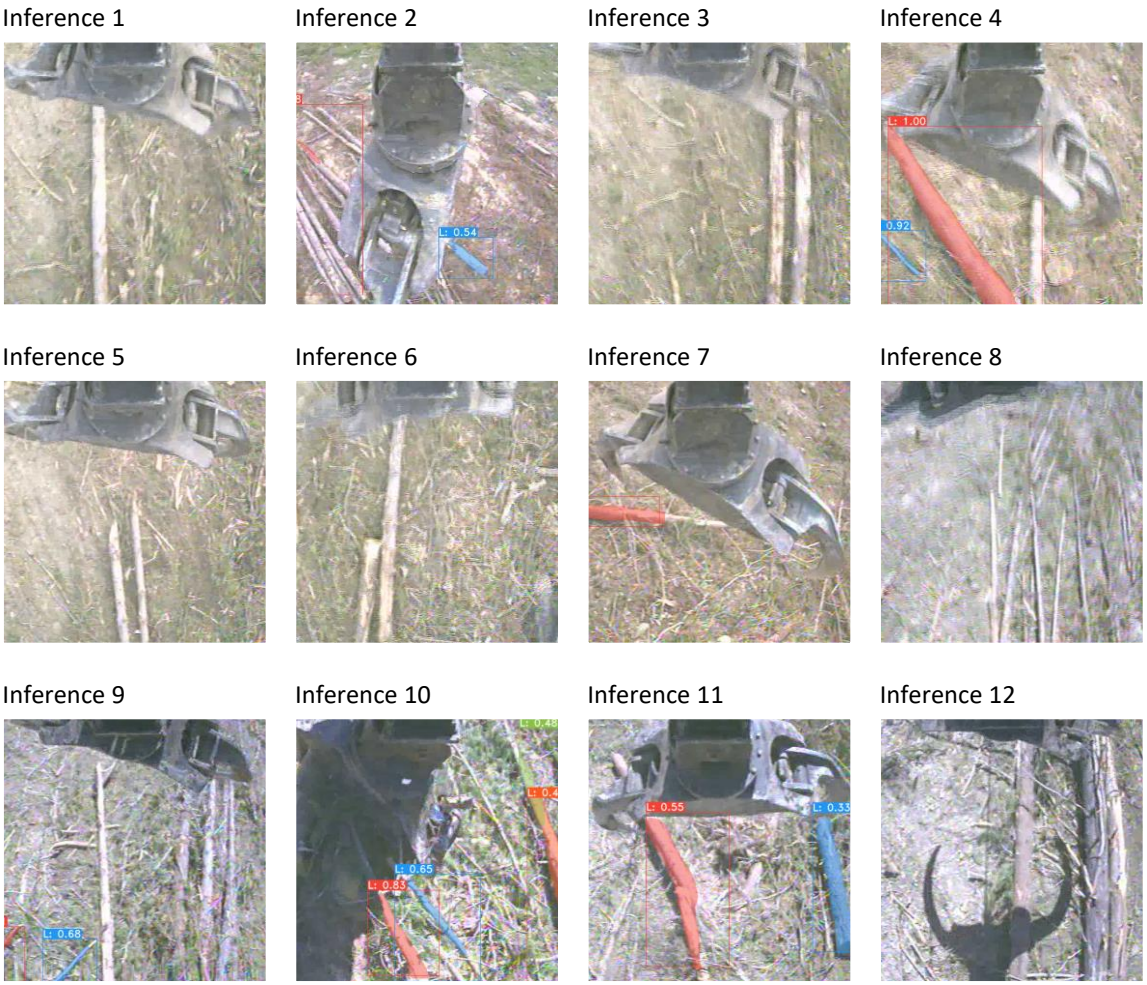

Figure 16 - Inferences on 12 images not included in the D.C. annotated dataset

6.3.2 Rayonier Footage

6.3.2.1 Quantitative Results

IoU Threshold	***0.5***	***0.6***	***0.7***	***0.8***	***0.9***
Box (Average Precision)	27.9	14.1	11.7	0	0
Mask (Average Precision)	21.6	18.0	9.6	2.0	0

Table 2 - Quantitative Model Performance, Rayonier

The quantitative results indicate that the model trained on the Rayonier footage performs significantly better than the DC footage. Mask AP at the 0.5 threshold was 21.6%, and box at 27.9%. Performance drops off significantly as the IoU threshold increases, and interestingly, box precision is lower at 0.8 than mask precision, although the low predictive accuracy of mask (1.98) in this instance shows there is little inference to be made from this observation

Using the evaluation function within the YOLACT using a video, the average FPS of the processing ability was 31.5FPS over the video.

6.3.2.2 Qualitative Results

Figure 17 above shows the trained model's inference on data excluded from the training dataset. The inferences show issues that are similar to the DC dataset. Of note, is that the model misses multiple logs in the cutover, especially in inference 5 and inference 1. Again, several logs are not identified as single instances, rather as multiple smaller logs. Inference 5 also has an instance where two logs have been detected as a single instance. However, inference 2 shows an example of a successful detection of a small log that is part of a mostly obscured stem.

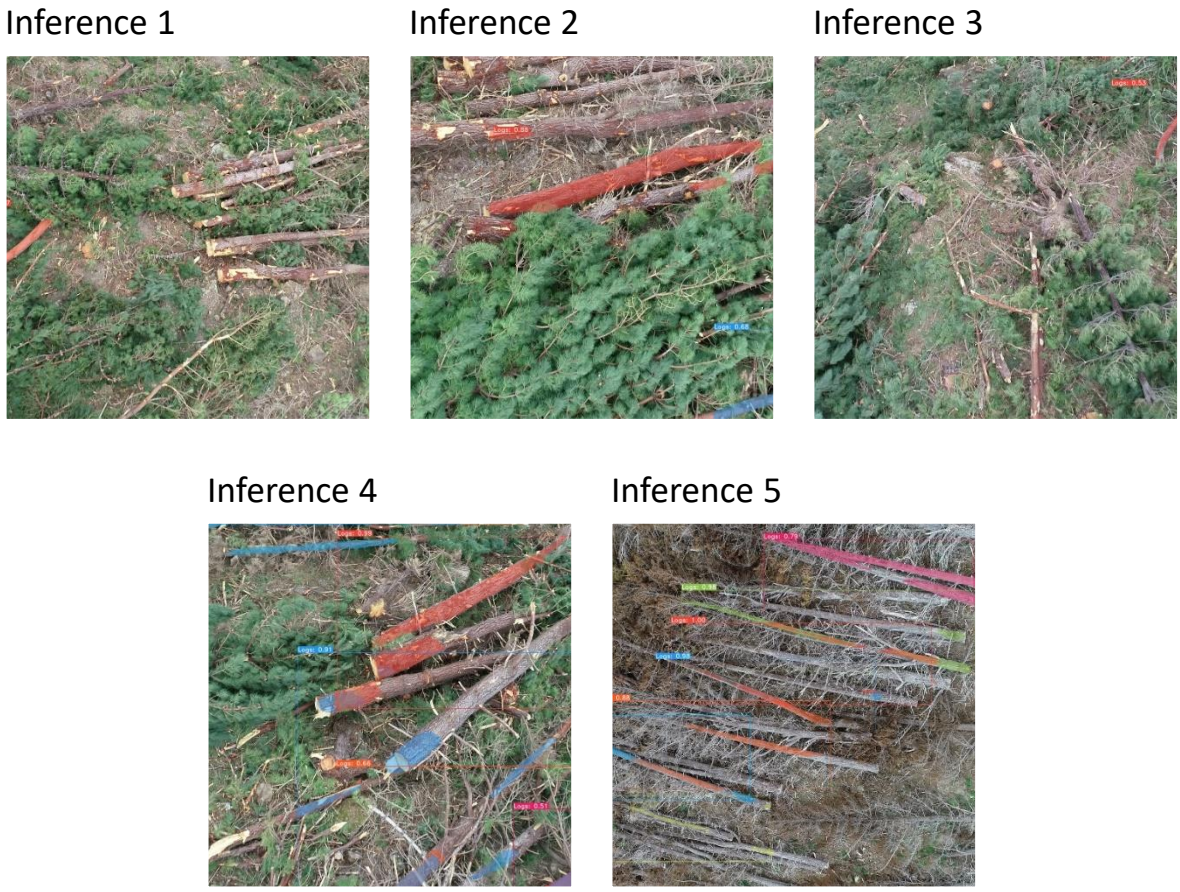

Figure 17 - Inferences on 5 images not included in the annotated Rayonier dataset

7 DISCUSSION

The overall results of the study show that both models, trained on their respective datasets, perform poorly when inferencing new data. Compared with other applications of the YOLACT algorithm [27,28], the model has poor accuracy in detecting and masking a log and is especially prone to missing logs that should be identified by the program.

7.1 DATA QUALITY

All machine learning models rely on the quality of the features that it can identify in the training set. The more consistent, detailed, and distinct from the surrounding environment and objects these features are, the higher the likelihood that the model can distinguish the target objects from its surroundings. There are several issues in these datasets, and the environment that the logs are in, that affect the quality of features within the dataset annotations.

The DC dataset has very low image quality, which may have contributed to the lack of accuracy in the results. Although this allowed it to perform at high speed, it is likely that it reduced the amount of information the algorithm was able to obtain from the images and discern logs from their surroundings. The machine learning algorithm relies on the image providing features of the target object to give the model an understanding of the features that logs contain. As the images are generally low quality, there are fewer discernible features with which to differentiate the logs from its surroundings. The D.C. logs main discernible features are straightness and colour, but these are common to other objects in the image, such as branches and the ground.

The Rayonier footage is of higher quality, with features being more detailed. Small details, such as bark texture, small branches and a greater depth of colour can be identified as compared with the DC footage. This contributes to its greater predictive accuracy compared with the DC model, as features unique to the logs such as butt ends or areas on the log where bark has been stripped, are well defined. In addition to increasing feature density, generating quality masks in the annotation section is easier with higher quality data. This is due to the easier identification of the boundaries of the log, which compounds the effect of high-quality data.

While not explored in this study, adding markers or discerning features to objects for detection is a common practice in machine vision applications, for example QR codes. While this would add complexity to the harvesting process, spray paint or other marking could be applied by the harvesting head at the time of felling, providing more information to the model as to what a log is, as it will be the only object in the frame with the colour of the marking.

7.2 INFERENCE ABILITY

Although YOLACT, as with all machine learning models, is a “black box,” meaning that the exact reasons as to why it performs in a certain way are not known, some insights can be delved from the inference images in figures 16 and 17. In general, the model is poor at detecting logs, with it missing several target logs in each frame. Particularly in the D.C. data, figure 16, many of the logs that should be identified as target logs are missed by the algorithm. The algorithm appears to struggle with logs that are not completely within the frame, such as inferences 1,3,4 and 6, although also fails to detect the end of a log in inference 9. Most logs that are vertically oriented are missed, which may point to insufficient examples of logs in this orientation in the training data. Logs at orientations other than vertically aligned to the image are detected. Misidentification of branches is seen in inference 4,9 and

10 of the DC dataset, which demonstrates the algorithm difficulty in discerning a log and a branch, as they share similar features (bark, colour, straightness). In addition, it has issues identifying logs as single entities, and instead separates a single log into multiple instances, seen in instance 10. This would pose issues for a working prototype, as determining a point on the log to grapple it would be difficult with multiple entities being recognised.

In the Rayonier data, many logs are also missed, but several target logs are identified in their entirety, particularly in inferences 2 and 5. Inference 5 shows an example image from a very dense pile of logs, where the algorithm does not correctly identify all logs in the image. Smaller sections of many of the logs are identified, which illustrates the difficulty the algorithm has in identifying the boundary of the log, which may be due to the similarities between the background of the logs, often dirt, and the log itself. Inferences 4 and 5 also contain examples where several different logs are identified as one log, despite there being foliage or ground in between them. This may have arisen from the inclusion of logs in the dataset that were partially obscured, so the algorithm believes that the small sections are part of the same log.

7.3 SPEED

Speed results for the model exceeded expectations, especially for the lower quality DC dataset. Other literature, as well as the YOLACT benchmark study on the COCO dataset, did not reach FPS speeds above 33.5 [18], which shows that if the accuracy of the program could be improved, it is possible that this method could be used for a real-time implementation in a machine. The increased speed relative to the original study is likely due to the slightly higher processing power of the graphics card used for this study and lower image resolution. While this card has more processing power than most current embedded hardware options, some high-end embedded systems do have the processing power available to achieve these results. This hardware continues to become cheaper and more accessible as machine learning applications become more common and would be a realistic inclusion in a grapple system if cost benefits from reducing operator load could be achieved.

7.4 LIGHTING

Lighting may also have contributed to the low general accuracy. Most machine vision applications are in consistent lighting, typically indoor settings, or settings where the machine can illuminate the scene to a consistent brightness. Although most humans can discern that an object in two different lightings is the same object, a computer vision model does not understand this context. While all efforts were made to control for this and use consistently lit logs, the difference in time during the footage will change the intensity of light, shadow angles and colour between training images. Inference 12 in figure 16, from the D.C. dataset contains an example where the shadow from the grapple carriage is cast on the logs to be grappled, and results in the logs not being detected.

For the Rayonier dataset, the lighting was significantly dimmer than the DC, which had some benefit as it removed the presence of shadows. However, the reduced light may have contributed to fewer features being discernible in the annotated images. As the Rayonier data is from a smaller section of video footage, the lighting differences throughout the video are less significant than the lighting changes over the hours of operation for the D.C. grapple footage.

7.5 Dataset Size

Both datasets are smaller than would typically be used for a commercial application. Typically, 1000 images per type of object is used to train a model, and these typically contain clearly defined and relatively uniform objects. Due to the time available for data collection and in particular annotation, the datasets were of a small size. For good general predictive ability, a large dataset with a large range of logs in different contexts improves is required to give consistent predictive accuracy across a range of conditions as would be encountered by a grapple camera. Especially in the case of the Rayonier dataset, the small dataset will make the overall accuracy lower than what could be expected with a large amount of data. The transfer learning implemented in the model reduces the need for large dataset size, but this model could still be improved by increasing the dataset size.

7.6 General Comments

It is critical in interpreting these results that the inference data is contextualised. Because the inference data comes from the same dataset with relatively similar conditions for time of day, log age etc., the predictive accuracy on this data will be greater than inference on conditions from another operation.

Logs on the cutover in general are difficult to discern. The criteria for log annotation included material that obscured the log in question. Because this material is likely to be present throughout the cutover, not only covering the logs, it is more difficult to discern a partially obscured log from its surroundings. Determining the effect of this would require annotating many different versions of the same dataset and comparing between them, with different inclusion criteria for annotations in each version.

In addition, the grapple claw in the D.C. footage poses an interesting problem for identification of logs. As the end goal is to segment the entire log, the claw often obscures the top of the image, where a significant proportion of a target log may lie. In this case, the entirety of the log, which may still be accurately detected, will be represented by only part of the logs mass. An automated grapple system would need to choose an appropriate position on the log, and as this position will need to be a certain distance from the centre of mass for a successful drag, there will be issues determining a point at which to grapple the log.

7.7 Commercialisation Potential

In general, this method is not a suitable candidate for use as a commercial application. Although the speed of the inference meets the likely requirements for such an application, the predictive accuracy is several orders of magnitude lower than what would be required for a viable product. While improvements in the dataset size and quality as discussed above would improve predictive accuracy, it is unlikely that these improvements would be sufficient to reach 90+%.

In addition, this study confined the conditions of yarding to certain lighting and weather. For a commercial product to have value, it would need to be generalisable across a range of conditions. This would require a large expansion of the dataset to include other conditions, so that the model can develop a broad understanding of possible contexts for the logs.

7.8 Further Research

Further research in this domain could improve the understanding and full capability of this technology for use in grapple yarding. Investigating how increasing dataset size improves model performance

would be beneficial to understanding the requirements for a complete dataset. Training the dataset with other instance segmentation algorithms to compare across models would provide a greater understanding of this model's performance and inference speed. In addition, determining whether an instance segmentation model not designed for real-time speed can identify logs at the necessary accuracy would provide a baseline for accuracy performance. Studies investigated in the literature review have determined logs can be identified at 93% precision, 82% recall [21], where speed was not a requirement, but this was achieved using multiple sensors. When performing this additional research, using images of the highest quality available would aid in improving model performance and mask quality.

In addition to improving the quality of data, other sensors could also be coupled with RGB imagery to provide more information on the logs position in the frame. LiDAR, infra-red cameras or radar could all be integrated with this approach to provide more information on the scene. Although this would increase the computational requirements, this may improve the overall accuracy of the model. In addition, experimenting with differing colour bands for the image may also provide additional details for each log that are not evident with the RGB band.

Another potential research avenue would be to constrain the detection problem to a very specific scenario first. If logs could be presented consistently, for example laid out as single logs in a consistent orientation, the performance of the model would be likely to improve. Other constraints could be making the lighting more consistent by using lighting from the grapple camera to provide a consistent lighting intensity. By exploring if these scenarios are feasible first, the dataset could then be extended to other lighting and log orientations in a gradual broadening of the machine capabilities.

8 CONCLUSION

The ability of the algorithm trained on these datasets to detect and mask logs is insufficient for commercial application, reaching only 12.8% AP (0.5) on the D.C. dataset, and 21.6% AP (0.5)on the Rayonier dataset. While this paper did not achieve the necessary performance in detecting logs to apply the technique in a commercial setting, it does identify issues with working with machine vision in the forestry domain. It highlights the need for high-quality images due to the relative similarity of logs and their surrounding environment. High quality imagery provides more features to the algorithm to discern between logs and their surroundings.

While this technique may be inadequate for the task, the issues encountered provide guidance for a scene where logs could be identified. As logs are similar to their surrounding environment, changing either the log or the environment may prove beneficial if it makes logs more distinguishable. Placing a marker or feature on logs, such as a clearly visible paint, may make them identifiable in the cutover. Other options that could be explored would be alternate lighting, such as lighting the scene at night using floodlights, or alternate sensors such as LiDAR or infra-red to provide more information to a detection algorithm, and more consistent features in the imagery.

The speed performance of the models, 50.2FPS on the D.C. data and the 31.5FPS on the Rayonier data, shows that these techniques could be used in a commercial setting. Although the model on its own would not be sufficient, the speed of this technique is such that with improved accuracy, such as possible with additional sensors or constraining or changing the environment, it may form part of a solution. The excess speed performance of this model also indicates that a more computationally expensive method, with higher accuracy, may be used and still perform at an adequate speed.

In summary, other techniques need to be explored beyond direct application of YOLACT for this task. The cutover is a complex environment, and the wide variety of positions and contexts a log may exist in requires that machine learning applications in this domain will require large datasets for adequate performance in a range of harvesting settings.

9 REFERENCES

1. Visser, R., & Harrill, H. (2017). Cable yarding in North America and New Zealand: a review of developments and practices. *Croatian Journal of Forest Engineering: Journal for Theory and Application of Forestry Engineering*, *38*(2), 209-217.
2. Harrill, H., Visser, R., & Raymond, K. (2019). New Zealand Cable Logging 2008–2018: a Period of Change. *Current Forestry Reports*, *5*(3), 114–123. https://doi.org/10.1007/s40725-019-00092-5
3. Brenda. R. Baillie. (2010). *Forest harvest practices in and around sensitive areas – a literature review* (No. 978-0-478-11039–1). Scion.
4. Dave Palmer, Shane Mcmahon, Daniel Fraser, & Rien Visser. (1996). *Skyline Logging to Minimise Impacts on Native Vegetation* (No. 1171–6932). LIRO. https://fgr.nz/documents/skyline-logging-minimise-impacts-native-vegetation/
5. Sessions, J., Lyons, K., & Wimer, J. (2021). Maximizing the Standing Skyline Log Load Using a Variable Length Tagline. *Forests*, *12*(7), 927. https://doi.org/10.3390/f12070927
6. Ministry for Primary Industries & Forest Growers Research. (2018). *Steepland Harvesting Programme - Post-Programme Report 2018*. Forest Growers Research. https://fgr.nz/documents/steepland-harvesting-programme-annual-report-2018/
7. Hunter Harrill, Rien Visser, & University of Canterbury. (2018). *Survey of Yarders and Rigging Configurations: 2018* (No. HTN10-04 2018). Forest Growers Research. https://fgr.nz/documents/survey-of-yarders-and-rigging-configurations-2018/
8. Pierzchała, M., Kvaal, K., Stampfer, K., & Talbot, B. (2017). Automatic recognition of work phases in cable yarding supported by sensor fusion. *International Journal of Forest Engineering*, *29*(1), 12–20. https://doi.org/10.1080/14942119.2017.1373502
9. *Scion - Boosting forest productivity*. (n.d.). Retrieved September 25, 2022, from https://www.scionresearch.com/science/growing-the-value-of-forests/boosting-forest-productivity
10. New Zealand Institute of Economics Research. (2021). *Forestry and wood processing labour force survey*. Ministry for Primary Industries New Zealand. https://www.mpi.govt.nz/dmsdocument/48667-2021-Forestry-and-wood-processing-labour-force-survey
11. Brzózko, J., Zychowicz, W., & Bartosiewicz, M. (2012, October). The Influence of Operator Experience on Productivity of Mechanized Timber Harvesting From Windfall Stands. In *Proceedings of the 45th International Symposium on Forestry Mechanisation:"Forest Engineering: Concern, Knowledge and Accountability in Today's Enviroment", Dubrovnik, Croatia* (pp. 8-12).
12. Thomas Hellstrom, Tomas Nordfjell, Pär Lärkeryd, & Ola Ringdahl. (2008). *Autonomous Forest Machines - Past, Present and Future* (UMINF 08.06). Umea University. http://umu.diva-portal.org/smash/record.jsf?pid=diva2%3A141956&dswid=3632
13. Schunnesson, H., Gustafson, A., & Kumar, U. (2009). Performance of automated LHD machines: A review. In *International Symposium on Mine Planning and Equipment Selection: 16/11/2009-19/11/2009*.
14. Hellström, T., Lärkeryd, P., Nordfjell, T., & Ringdahl, O. (2009). Autonomous Forest Vehicles: Historic, envisioned, and state-of-the-art. *International Journal of Forest Engineering*, *20*(1), 31–38. https://doi.org/10.1080/14942119.2009.10702573

15. Campbell, T. (2016). *Assessment of the Opportunity of Modern Cable Yarders for Application in New Zealand* [MA Thesis]. University of Canterbury.
16. Spinelli, R., Visser, R., Magagnotti, N., Lombardini, C., & Ottaviani-Almo, G. (2020). The effect of partial automation on the productivity and cost of a mobile tower yarder. *Annals of Forest Research*, *63*(2), 3-14.
17. Visser, R., & Obi, O. F. (2020). Automation and Robotics in Forest Harvesting Operations. *Croatian Journal of Forest Engineering*, *42*(1), 13–24. https://doi.org/10.5552/crojfe.2021.739
18. J. Fortin, O. Gamache, V. Grondin, F. Pomerleau, and P. Guigere, "'Instance Segmentation for Autonomous Log Grasping in Forestry Operations.,'" *arXiv preprint* arXiv:2203.01902, Mar. 2022.
19. M. Carpentier, P. Giguere, and J. Gaudreault, "Tree Species Identification from Bark Images Using Convolutional Neural Networks," *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS),* Oct. 2018.
20. S. Li and H. Lideskog, "Implementation of a System for Real-Time Detection and Localization of Terrain Objects on Harvested Forest Land," Forests, vol. 12, no. 9, p. 1142, Sep. 2021.
21. P. Polewski, J. Shelton, W. Yao, and M. Heurich, "Instance segmentation of fallen trees in aerial color infrared imagery using active multi-contour evolution with fully convolutional network-based intensity priors," *ISPRS Journal of Photogrammetry and Remote Sensing, vol. 178*, pp. 297–313, Aug. 2021.
22. S. Li and H. Lideskog, "Implementation of a System for Real-Time Detection and Localization of Terrain Objects on Harvested Forest Land," *Forests, vol. 12, no. 9, p. 1142*, Sep. 2021.
23. Han, P. (2021). *Automating Log Sweep Assessment* [Honours Thesis]. University of Canterbury.
24. D. Q. da Silva, F. N. dos Santos, A. J. Sousa, and V. Filipe, "Visible and Thermal Image-Based Trunk Detection with Deep Learning for Forestry Mobile Robotics," *Journal of Imaging, vol. 7, no. 9, p. 176*, Sep. 2021.
25. D. Tian, Y. Han, B. Wang, T. Guan, H. Gu, and W. Wei, "Review of object instance segmentation based on deep learning," *Journal of Electronic Imaging, vol. 31, no. 04*, Dec. 2021.
26. D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, "YOLACT: Real-time Instance Segmentation," *Proceedings of the IEEE/CVF international conference on computer vision*. 2019.
27. S. S. Heng, A. U. bin Shamsudin, and T. M. M. Said Mohamed, "Road Sign Instance Segmentation By Using YOLACT For Semi- Autonomous Vehicle In Malaysia," *2021 8th International Conference on Computer and Communication Engineering (ICCCE)*.
28. L. Piyathilaka, D. M. G. Preethichandra, U. Izhar, and G. Kahandawa, "Real-Time Concrete Crack Detection and Instance Segmentation using Deep Transfer Learning," *7th International Electronic Conference on Sensors and Applications*.
29. B. Wicht, "Deep Learning feature Extraction for Image Processing." PhD thesis, University of Fribourg, Switzerland, 2018.
30. H. Yingge, I. Ali, and K.-Y. Lee, "Deep Neural Networks on Chip - A Survey," *2020 IEEE International Conference on Big Data and Smart Computing* (BigComp), Feb. 2020.
31. B. Ramsundar and R.B. Zadeh, "Fully Connected Deep Networks" in TensorFlow for deep learning : from linear regression to reinforcement learning. Beijing: O'reilly Media, 2018.
32. E. Shelhamer, J. Long, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," *IEEE transactions on pattern analysis and machine intelligence, vol. 39, no. 4*, pp. 640–651, 2017.

33. R. Draelos, “Segmentation: U-Net, mask R-CNN, and medical applications,”Glass Box, Internet: https://glassboxmedicine.com/2020/01/21/segmentation-u-net-mask-r-cnn-and-medical-applications/,21-Jan-2020 [2-May-2022].
34. K. Sambasivarao, “Non-maximum Suppression (NMS),” Medium, Internet: https://towardsdatascience.com/non-maximum-suppression- nms-93ce178e177c, Oct. 01, 2019 [2-May-2022].
35. Davis, J. (2013, February 20). *Pinus Radiata*. https://davisla.wordpress.com/2013/02/20/plant-of-the-week-pinus-radiata/
36. V. Kůrková, Y. Manolopoulos, B. Hammer, L. Iliadis, and I. Maglogiannis, “A survey on Deep Transfer Learning” in *Artificial Neural Networks and Machine Learning – ICANN 2018 27th International Conference on Artificial Neural Networks*, Rhodes, Greece, October 4-7, 2018, Proceedings, Part III. Cham Springer International Publishing, 2018.
37. Mohan, A., Kaseb, A. S., Gauen, K. W., Lu, Y. H., Reibman, A. R., & Hacker, T. J. (2018, April). Determining the necessary frame rate of video data for object tracking under accuracy constraints. In *2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)* (pp. 368-371). IEEE.
38. Warden, P. (2017, December 14). *How many images do you need to train a neural network?* Pete Warden’s Blog. https://petewarden.com/2017/12/14/how-many-images-do-you-need-to-train-a-neural-network/
39. *Understand measures of supercomputer performance and storage system capacity*. (n.d.). Copyright 2022, the Trustees of Indiana University. Retrieved October 25, 2022, from https://kb.iu.edu/d/apeq
40. *Jetson AGX Xavier*. (2022, January 15). NVIDIA Developer. https://developer.nvidia.com/embedded/jetson-agx-xavier
41. S. Yohanandan, “Map (mean average precision) might confuse you!,” Medium, Internet: https://towardsdatascience.com/map-mean-average- precision-might-confuse-you-5956f1bfa9e2, 09-Jun-2020 [20-May- 2022