PO Box 1127
Rotorua 3040
Ph: + 64 7 921 1883
Fax: + 64 7 921 1020
Email: forestgrowersresearch@fgr.nz
Web: www.fgr.nz

Programme: Harvesting & Logistics

Task No: 3.2 **Report No. H064**

Development of a Prototype Bark Detection System for a Fixed Debarking Facility

Authors:
Paul Milliken, Jaco Fourie, Daniel Lamborn, Jeffrey Hsiao, Dean Williamson, and Ian Brown

Research Provider:
Applied Teleoperation Limited
and
Lincoln Agritech Limited

Date: 31 December 2023

TABLE OF CONTENTS

Disclaimer

This report has been prepared by Applied Teleoperation Limited (ATL) and Lincoln Agritech Limited (LAL) for Forest Growers Research Limited (FGR) subject to the terms and conditions of a Services Agreement dated 20 July 2021.

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, ATL's and LAL's liability to FGR in relation to the services provided to produce this report is limited to the value of those services. Neither ATL, LAL 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

Introduction

Recent changes to the rules around the use of methyl bromide for fumigation of logs for export has led to increasing use of alternative phytosanitary treatments for logs. One of the alternatives is debarking. The quality of debarking is currently assessed visually when logs are in a stack and most of the surface area of the logs is not visible. This project aims to provide a more thorough and objective measure of the quality of debarking.

This document describes the development of a prototype system for real-time bark detection and quantification of the amount of bark remaining. The system uses cameras to identify bark remaining on debarked logs as they travel on a conveyor in a fixed facility, the Kaingaroa Processing Plant (KPP).

Methodology Overview

Hardware Development

A prototype bark-detection system was developed and tested on a conveyor carrying debarked logs at KPP. Hardware included three IP cameras, a central computer, and a touchscreen with a user interface for the operator. Bright halogen lighting was used to illuminate the logs so that images could be captured in high resolution without motion blur. Sample images were collected and provided to Lincoln Agritech Limited so that they could develop and test software.

Software Development

Software development involved writing code for detection of the log ends, log boundary detection, bark detection and real-time image stitching. Bark detection was achieved using a convolutional neural network (CNN) that was trained with the sample images. Other parts of the software were developed using conventional techniques.

Testing and Refinement

A cycle of testing and development was used to guide improvements in software and hardware to the point where the system was able to achieve real-time processing with reasonable accuracy for the majority of logs.

Overview of Results

System hardware

After development of the hardware, the system was set up at KPP to capture video of debarked logs as they travelled on a conveyor at up to 12kph. Three cameras were sufficient to see most of the surface of the logs, however, the bottom surface of the logs was not visible due to the conveyor.

The speed of the conveyor and proximity of the side cameras meant that a very fast shutter speed was required to eliminate motion blur. This, in turn, necessitated bright halogen lighting of 2,000 Watts power. This amount of halogen lighting generates a lot of heat so flicker-free LED lighting is recommended for commercial installations.

Software

The software to process the images consisted of the following four parts:

1. Detection of log start/end frames

2. Detection of areas that are part of a log (excluding pixels that form part of the background)
3. Detection of areas of bark
4. Longitudinal stitching

The resulting metric was an estimate of the proportion of the area of the surface of each log covered by bark.

User interface

A user interface on a touchscreen was developed. Live video from three cameras is shown in three inset views at the top right of the display. Pictures of the sides of the current log are shown in the lower half of the screen. The sides are composed of stitched frames from the video streams. An overlay showing the bark area and the log boundary can be toggled via a button on the touchscreen. The percentage of the area covered by bark is displayed, as well as an average for the last one hundred logs. The operator can view a history of logs and their summary statistics. The results (including the stitched images) usually appear on the screen within 5 seconds of detection of the end of a log.

Performance

The tool worked well for most logs. The report includes images showing the three stitched sides of an example log, together with an image of the same log with bark highlighted in green and the log area with a red border.

Detection of the log ends was not always accurate. Sometimes, the end of the log was falsely detected causing a log image to be cut short. There were also examples when the background was falsely classified as part of a log. Detection of the log was accurate most of the time, although incorrect classifications of log as background and background as part of a log were observed.

Detection of bark however was usually satisfactory. When there was insufficient lighting, false detection of bark occurred more frequently. Misclassification was most problematic when areas of the background were also misclassified as bark, resulting in over-estimation of bark area.

The performance of the stitching algorithm was acceptable. Measurement or estimation of speed and distance from the camera to the surface of the log would be one possible way to make the stitching more robust. Certain situations would cause the software to slow down or crash. For example, if a log stopped in front of the camera, this caused the central processing computer to freeze.

Feedback from demonstrations

Two demonstrations were performed at KPP. Feedback from industry was that there is interest in the system and in the possibility of interfacing it with control systems for the KPP plant. The possibility of modifying the system to add a new metric was discussed. The new metric that was proposed was the largest contiguous area of bark on each log.

Economic analysis

An economic analysis was undertaken and has been presented in an earlier report (Milliken *et al.*, 2023). The economic analysis considered a possible investment in a bark-detection system for a hypothetical fixed debarker. The benefits were in terms of a reduction in the risk that debarking would be removed as a phytosanitary treatment option resulting in increased costs for fumigation with full recapture. The Net Present Value was positive for the example that was considered.

Conclusions

- A prototype bark-detection system was developed, tested, and demonstrated at the Kaingaroa Processing Plant.

- The system measured and displayed estimates of percentage of bark remaining for each log.

- Software consisted of an algorithm to detect log ends, an artificial intelligence model that was trained for bark detection, an algorithm to detect log area and software for stitching the images together to recreate the surfaces of the logs.

- The software worked adequately most of the time. However, certain situations provoked problems with robustness and accuracy. The system is not yet considered mature enough for a commercial installation.

- The addition of another metric, the largest contiguous bark area, has been recommended by industry.

- An economic analysis has been presented in a separate report (Milliken *et al*., 2023)

INTRODUCTION

Background

Recent changes to the rules around the use of methyl bromide for fumigation of logs for export has led to the use of alternative phytosanitary treatments for logs. One of the alternatives is debarking. The quality of debarking is currently assessed visually when logs are in a stack and most of the surface area of the logs is not visible. This project aims to provide a more accurate and objective measure of the quality of debarking.

This document describes the development of a prototype system for real-time bark detection and quantification of the proportion of bark remaining. The system uses cameras to capture images of the surfaces of the logs. Software is then used to process the images to identify bark remaining as logs travel on a conveyor. The system was tested on a fixed facility (KPP).

Objective

To detect bark remaining on debarked logs so that debarking meets both the domestic wood processing debarking standard (less than 1% bark remaining) and export phytosanitary debarking standard (less than 5% bark remaining per log and less than 2% per batch). The tool is to be developed for, and tested on, debarked logs in a fixed debarking facility.

METHOD

The method consisted of four milestones. Milestones 1, 2 and 3 are described below and Milestone 4 is an economic analysis that is presented in a separate report (Milliken *et al*., 2023).

Milestone 1 – develop hardware and collect sample images.

KPP was chosen as a site for development of the prototype system. Representatives of KPP, FGR and ATL met at KPP to select a suitable site for the installation. KPP modified the sides of the conveyor to allow a camera to be located on each side. An archway was fabricated to support a camera situated above the conveyor. Ethernet cables connected the cameras to a POE network switch in a control room beside the conveyor. A laptop was used for the first rounds of data collection.

Milestone 2 – Develop bark detection software.

The algorithms for the detection and localisation of bark were developed by Lincoln Agritech Limited (LAL), a technology partner in the project. Bark detection fitted into the process as follows:

1. Video frames that belonged to a single log were fed into the bark detection algorithm.
2. Detection of bark on individual log frames was carried out.
3. Stitching of log frames into a single log image was done and bark area was highlighted with an overlay.

The input to the algorithm was a continuous stream of video frames from three cameras, one pointing at the top of the log, one camera pointing at the left side of the log and one to view the right side. In the first step, the algorithm finds the start and end of the log by detecting the log edges and determining whether it is a log start or end by analysing the light intensity gradient across the centre of the image frame. All the frames captured between the start and end of the log were consolidated into batches for efficient processing by the convolutional neural network (CNN) that was trained to find bark patches.

CNNs are a very common and well-established approach to using artificial intelligence (AI) for object detection. A standard model (YOLOv7) was adapted for robust and fast detection by retraining the model on manually labelled examples of bark patches on logs. Retraining of the model was done by using the pre-trained model as a starting point and fine-tuning it to recognise bark patches on the kinds of images that were expected from the cameras mounted on the conveyor belt.

LAL extracted a diverse set of 6,263 images from the test videos captured at KPP and they were labelled by adding polygons around the bark patches. This was a time consuming and repetitive process, but it was critical that labels were consistent so that the model was able to learn which textures to associate with bark and which were to be ignored. This resulted in 9,561 bark labels that were split into a training set and a validation data set.

Around 80% of the labelled images were used for training the model and 20% were kept apart to verify the model’s accuracy on unseen examples. A selection of these labelled examples of bark were verified by experts from FGR to ensure that bark and cambium were distinguished. Figure 1 shows an example of a log image that has bark regions manually annotated for training. In this example two labels have been added to the data set.

Figure 1: Example of a log image that has bark regions manually annotated for training.

The final step was to combine the frames from a single log into a stitched image, add the combined bark overlay and calculate the total amount of bark as a percentage of the total log area.

This step was complicated by the fact that real-time operation was required. The system had to detect log ends, detect bark areas, and construct a stitched image, all before the next frame from the video stream arrived. This meant it was not possible to use the traditional approach to constructing stitched images that involves the extraction of key-points from each frame, the calculation of homographies between key-points, and warping of overlapping frames based on the calculated homography transform.

Instead, it was necessary to develop a faster stitching algorithm that took advantage of the simple motion of the logs as they traversed almost linearly in front of the cameras. Instead of key-point extraction and homography transforms, simple correlation of image patches was used to calculate a constant pixel-shift between subsequent frames. The result is a stitched image that does not have the quality of a traditional key-point based result but can be constructed much faster and is of sufficient quality to provide an approximate reconstruction of the bark on the log.

Milestone 3 - Testing and refinement

The software was developed on the images captured in Milestone 1. The system was tested at KPP. The computational requirements to run the model were greater than expected and even a single video stream was not able to be processed in real-time on the camera as initially planned.

An ATL TC2 Camera was upgraded to include a module with significantly more processing capability. Even with this module there were stability problems, and it was not clear that the camera would keep up with the demands of detection and stitching.

Therefore, a decision was made to use a tower computer with two Nvidia GPUs for image processing. A Nvidia Gforce GTX1650 card was used. This was found to be sufficient for real time operation with a single camera. This was what was used for the demonstration on 10 October 2023.

Optimization was done to get all three streams processing on the same computer in real time. The CPU (as distinct from the GPU) usage was found to be quite high, and much of the code used only a single core (it was not multi-threaded).

Synchronization of the sides of each log was not trivial to solve because a side of a log could be missed if processing took too long. This would occur if logs came in faster than the time required to process them. A time stamp was assigned when the start of each log was detected to facilitate grouping the three sides correctly.

To get the prototype working with all three cameras simultaneously, a new motherboard with a 20 core i7 CPU and Nvidia Gforce RTX3060 was purchased. This was just sufficient to allow bark detection from all three streams to run simultaneously with the current software.

The stitching algorithm was modified to run more quickly, and improvements were made to the log detection algorithm. The background images were critical for log detection, so more refinement was made to the way background images were used.

The system was tested again at KPP and more improvements to the software were made before demonstration of the prototype system on 4 December 2023.

Milestone 4 – Economic analysis

Milestone 4 comprised an economic analysis, which has been documented in a separate report (Milliken *et al.*, 2023).

RESULTS

Prototype System Hardware and configuration system.

Existing Conveyor

Representatives of FGR, ATL and KPP met at KPP to choose a location for the installation. The system was installed on a conveyor to measure logs post debarking, bucking and removal of reject material. The logs travel towards the camera at a speed of up to 12kph (3.33 m/s). Temporary lighting was used for the development of the prototype system. Figure 2 shows the installation of the lighting. Halogen lamps were chosen for testing as they produced a broad natural spectrum of light, and they caused no problems with image flicker or banding.

Figure 2: Installation of the lighting.

Three cameras were installed. Cameras were located on either side of the conveyor and one camera was positioned above the conveyor as shown in Figure 3.

Figure 3: Cameras were located on either side of the conveyor (top right and top left) and one camera was positioned above the conveyor (bottom)

Network architecture.

The network architecture is shown in the diagram overleaf (Figure 4).

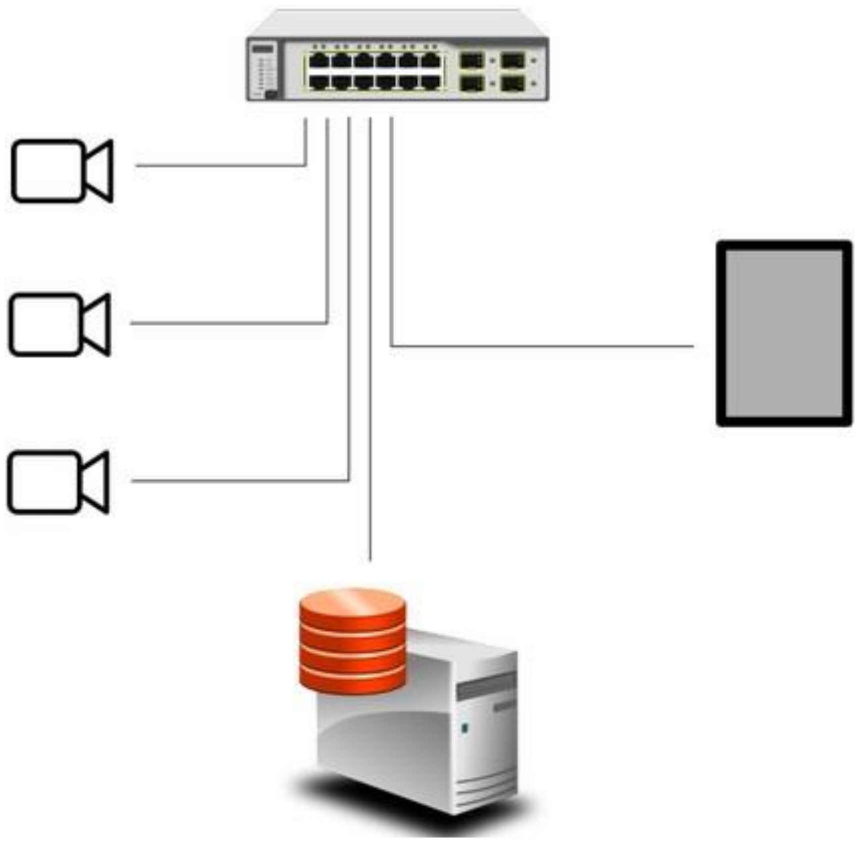


Figure 4: Network architecture diagram

The three IP cameras sent images over the Transmission Control Protocol/Internet Protocol (TCP/IP). The main computing unit received the video streams from the three cameras and performed the analysis. An industrial touch screen displayed video, processed images and summary data to the operator.

Camera Hardware

The cameras that were used were ATL TCamera2 IP cameras. Lenses were selected and configured so that, for a large log, the log would occupy most of the frame. Optically corrected wide-angle lenses were used to remove fish-eye distortion. However, the proximity of the side cameras to the log was not ideal and resulted in a noticeable parallax effect in the frames.

Camera settings

This section of the report contains technical details of how the cameras were configured. Cameras 1, 2 and 3 were assigned IP addresses of 192.168.1.85, 192.168.1.86 and 192.168.1.87, respectively. The cameras were each set up to run a Gstreamer pipeline that sent video over both TCP/IP and UDP/IP. The frame rate was set to 15fps. The UDP stream was lower resolution and was used to display live video from the cameras while the TCP stream was used for analysis.

Video settings were fixed so that the images would be as consistent as possible. The exposure time range parameter in the pipeline was set to 500 microseconds and exposure compensation was -2. The resolution was 1080p and encoding was H265 with a bit rate of 20 Mbps.

Lighting

Bright lights were required because the shutter speed needed to be fast to avoid motion blur. The amount of light required was determined by experimentation. Halogen lights totalling 2,000 Watts were found to provide the correct amount of illumination. Williams (2023) states that halogen lamps produce 25 lumens per Watt. So, this is around 50,000 lumens distributed around the log. The lights were around 1.5 metres from the surface of the log and illuminated a length of around two metres of the log length.

LED lights run cooler than halogen lamps and require less maintenance. However, the LED lights that were tested in the initial stages of development caused banding of the video due to the rapid flicker of the LED lights. Later in the project, further testing was done on a selection of lights, and an LED light that did not cause banding was found. The model was: “Urban Interior Mons.60 LED batten PLU 3451”. The higher power version “Urban Interior Mons.60 batten PLU 3293” (Lightingplus, 2023), shown in Figure 5, is recommended for permanent installations.

Figure 5: “Urban Interior Mons.60 LED batten PLU 3451” high power lighting.

Software implementation of algorithm

Architecture of software

A diagram of the software architecture is shown in Figure 6:

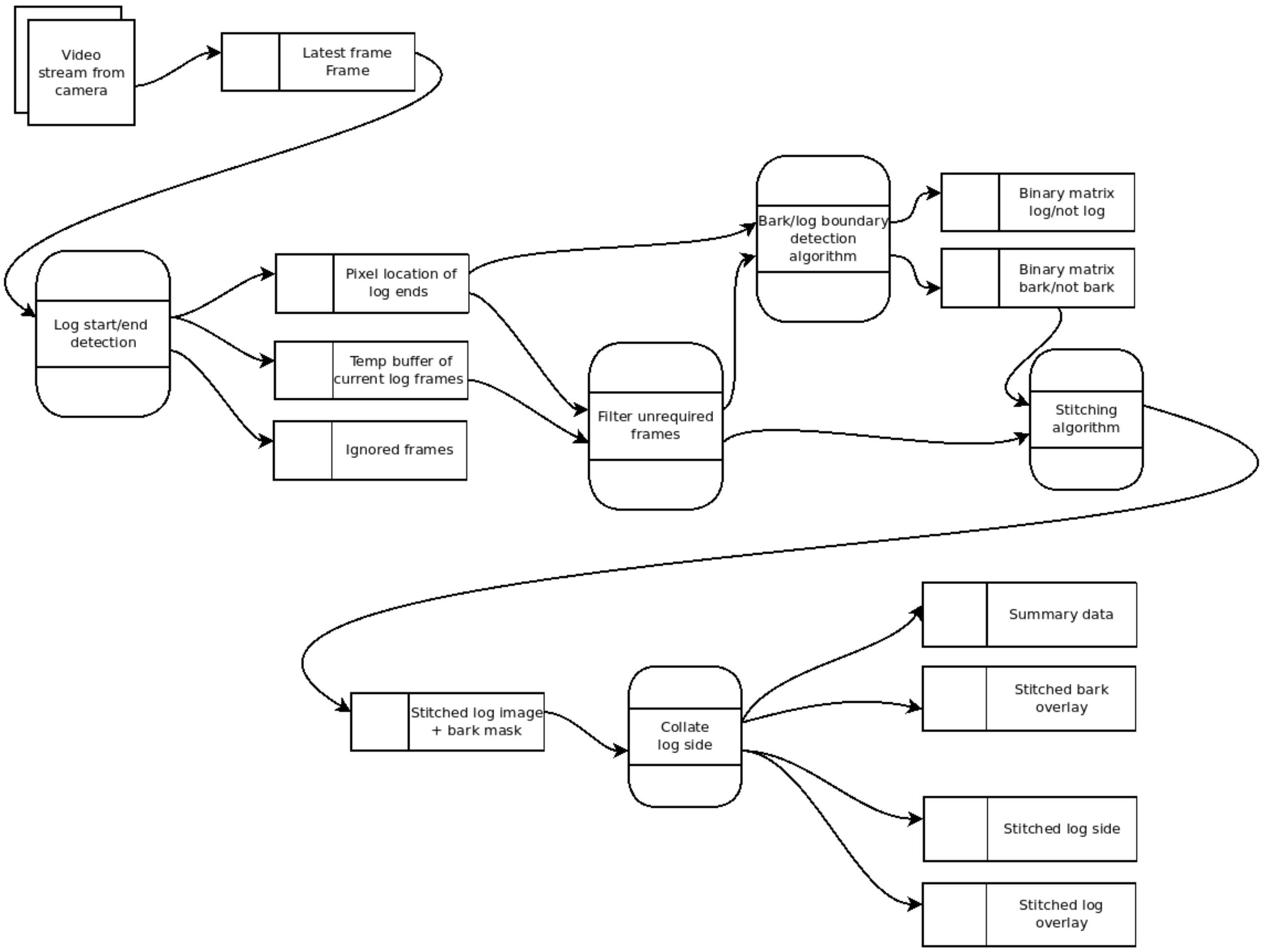


Figure 6: Yourdon diagram of the software architecture

A description of how each element of the software was implemented is described in the following sections of the report.

Video frames

Video was received by the central computer from three cameras over TCP/IP at a speed of 15 frames per second. The resolution was 1080p and the video was encoded as H265. The gstreamer pipeline, in Python, was:

```
pipeline = Gst.parse_launch(f"tcpclientsrc host={args.input} port={args.port} ! matroskademux "
                             f"! decodebin ! videoconvert ! video/x-raw, format=BGR ! appsink")
```

Detection of log boundaries

Detection of the log ends was done on a per-frame basis by following these steps on each incoming video frame:

1. Extract a central slice 30 pixels wide that goes through the centre of the log and covers the entire frame.
2. Calculate a pixel sum of the slice laterally across the log resulting in a single-pixel line along the centre of the log.
3. Add the appropriate zero-padding and convolve this pixel vector with a five-pixel averaging window to smooth out any noise and sharp textures on the log.
4. Calculate the gradient of the resulting vector to highlight sharp discontinuities in the image brightness that could indicate a log boundary.
5. Extract minimum and maximum gradient peaks and use predetermined thresholds to determine whether a log end has been found.
6. Depending on whether the gradient peak is positive or negative and whether the log moved from the top of the frame or the bottom, classify the pixel location corresponding to the gradient peak as either a log start or log end point.
7. Filter out false detections based on whether a log start or log end is expected (based on previous frames).

Bark detection.

Once the frames that belong to a specific log have been identified by finding the log boundary frames, the individual log frames are pushed through a pre-trained CNN that identifies where bark areas are located. The images must be resized and cropped to a region-of-interest (ROI) before being sent to the CNN. This was done to maximise the speed of operation as performing inference on a CNN of this size is a computationally expensive operation.

The output from the CNN is a collection of pixel masks that can be overlaid on the frames to identify bark. For this real-time application, a trimmed version of the YOLOv7 model was used that had been trained on manually labelled images of logs with examples of bark highlighted.

Stitching

The creation of a larger combined image that shows the entire length of a log instead of individual log frames is known as stitching. Both the original frames as well as the bark masks needed to be stitched so that they lined up exactly.

The stitching was performed by first calculating the amount of overlap between subsequent frames. A simple image correlation approach was used to ensure that stitching could be completed in real time.

First a small 130x260-pixel patch was extracted from the initial frame. This patch was then correlated with each possible location on the following frame and the position of highest correlation was compared with the location where the patch was extracted in the initial frame. The pixel-difference between the initial location and the highest-correlation location indicates the approximate inter-frame overlap. We calculated this for all the frames in the log and then took the median overlap over all the frames as the image shift for the entire stitch. Here it was assumed that the speed of the conveyor belt was approximately constant. This was done because stitching based on individual frame-overlaps added too much noise that could not be filtered out fast enough for real-time operation.

Given the calculated image shift, the stitched image was assembled by applying subsequent images to the stitch with the overlap until all the images had been applied and the end of the log had been reached. The software checked that, when overlaying subsequent images, bark from the previous frame was not hidden by the overlaid frame. We calculated the maximum bark area for each part of the frame that may be replaced by a subsequent frame and determined which frame should be "on top" when the new frame was applied to the stitch.

This process was repeated using the same image shift to construct the bark mask that was overlaid on the stitched log to identify the bark locations.

Summary metrics

The metric for debarking was an estimate of the proportion of the surface area of the log that was covered by bark. Equivalently the bark-on proportion (p_b) is given by:

$p_b = A_b / A_l$, where A_b is the bark area and A_l is the log area (including bark).

At the first demonstration of the prototype system, carried out on 10 October 2023, it was suggested that another metric which is the largest contiguous area of bark could be added. This is discussed in more detail in the Discussion.

User interface, storage of summary data and operator display

A photo of the operator's screen is shown in Figure 7 below.

Figure 7: Screenshot of the operator's screen

The bark overlay can be turned on and off via the touchscreen. Live video streams from the three cameras are shown at the top of the screen.

In the prototype system, data were stored as files on a file server (NFS). The stitched images and the masks for bark detection and log detection were stored as reduced resolution files. Images for each side were stored separately and the filenames were used as identifiers. The bark percentage that is displayed is the average of the bark percentage for the three sides. A moving average is also displayed (average of last 100 logs).

The data were stored as files in the following pattern:

- The file share/[cam-num]/[epoch]_data.json contains summary data
- [cam-num]/[epoch]_mask_stitch.png
- share/[cam-num]/[epoch]_stitch.jpg

The json file includes log_num, camera_id, epoch, and bark_percentage. The user could scroll back through images and data from previous logs.

Performance

Errors in the estimation of bark proportion

The estimate of bark percentage depends on the estimate of bark area (the numerator) and the estimate of log area (the denominator). Errors could occur due to the following:

- False positives and false negatives for bark area
- False positives and false negatives for log area
- False detection of end of log (this can result in a foreshortened log)

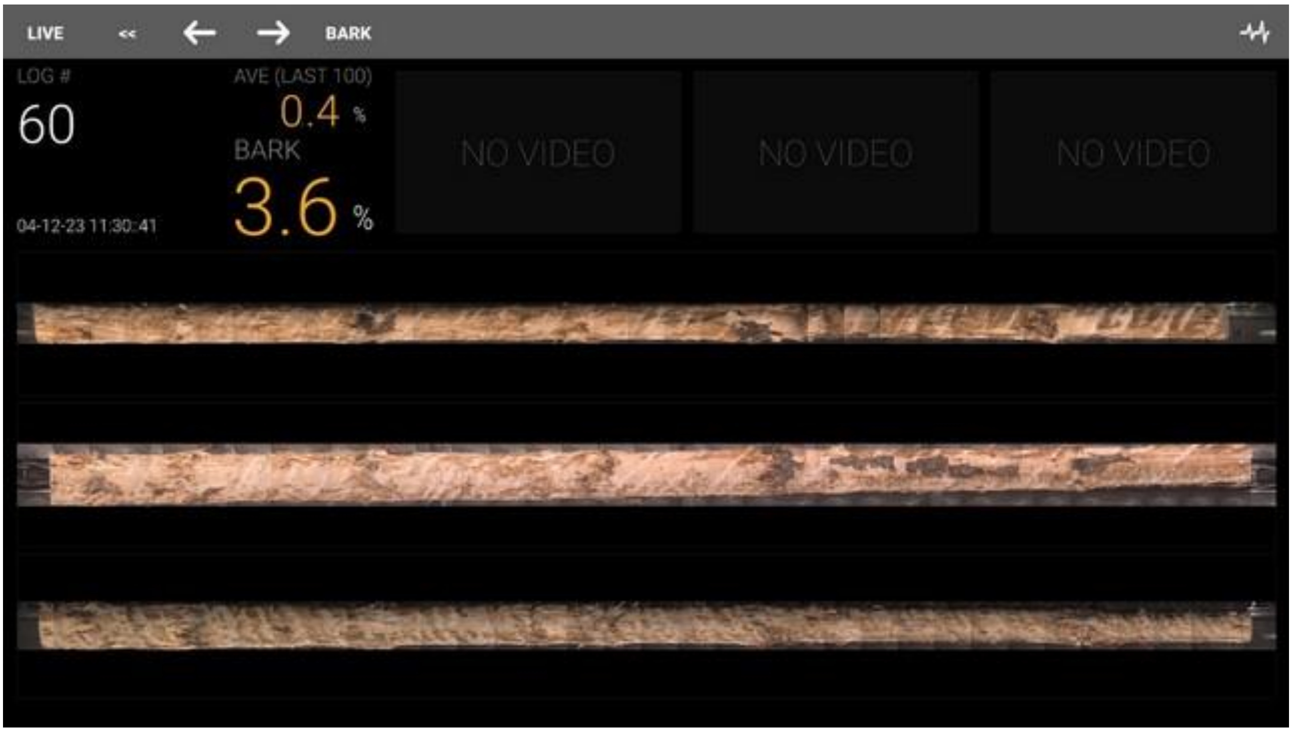
Errors could also occur due to the following effects:

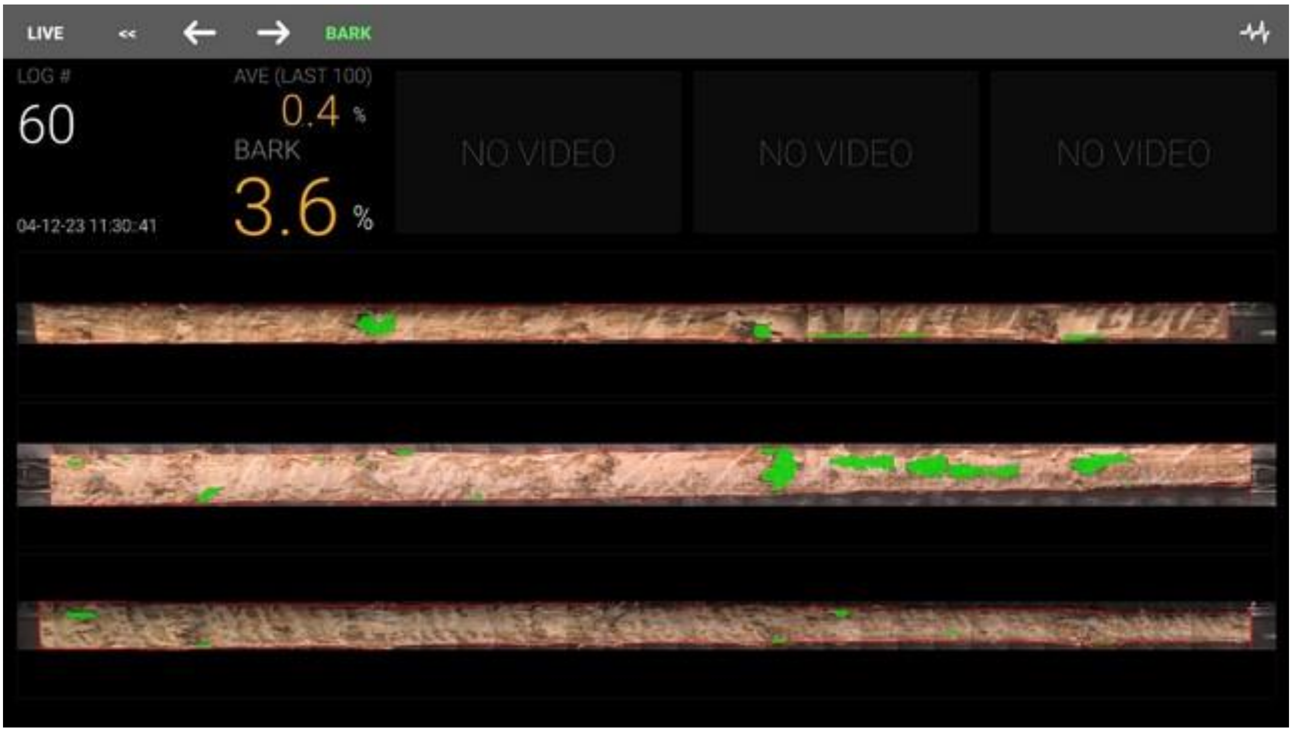
- The proximity of the log to the camera is not known.
- The surface of the logs are curved so pixels normal to the camera appear larger.
- There is some overlap between the three camera views so there is potential for double-counting of parts of the log surface.
- The bottom part of the log is not visible.
- There are errors in stitching.

Example results

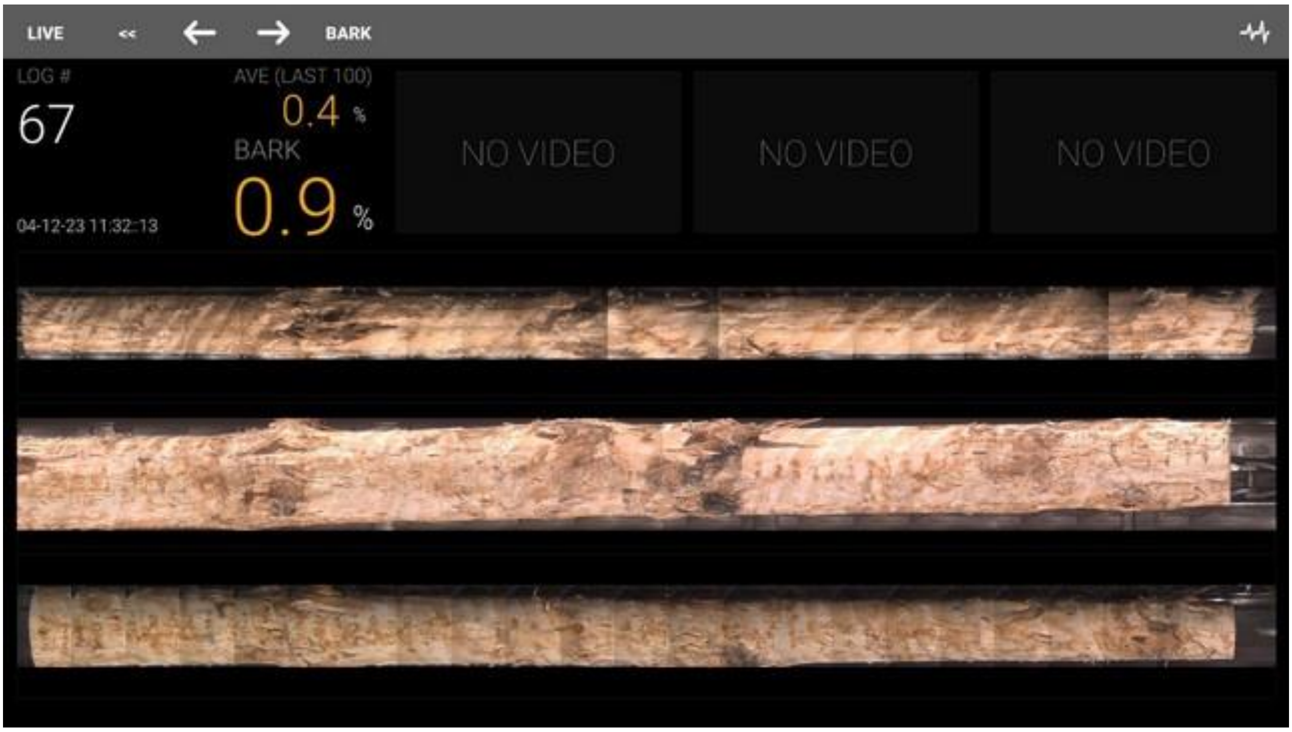
Example images showing bark detection and log detection are presented below. The images are in pairs. A raw stitched image is shown, followed by the same image with overlays for the bark area (highlighted green) and log area (shown with a red border).

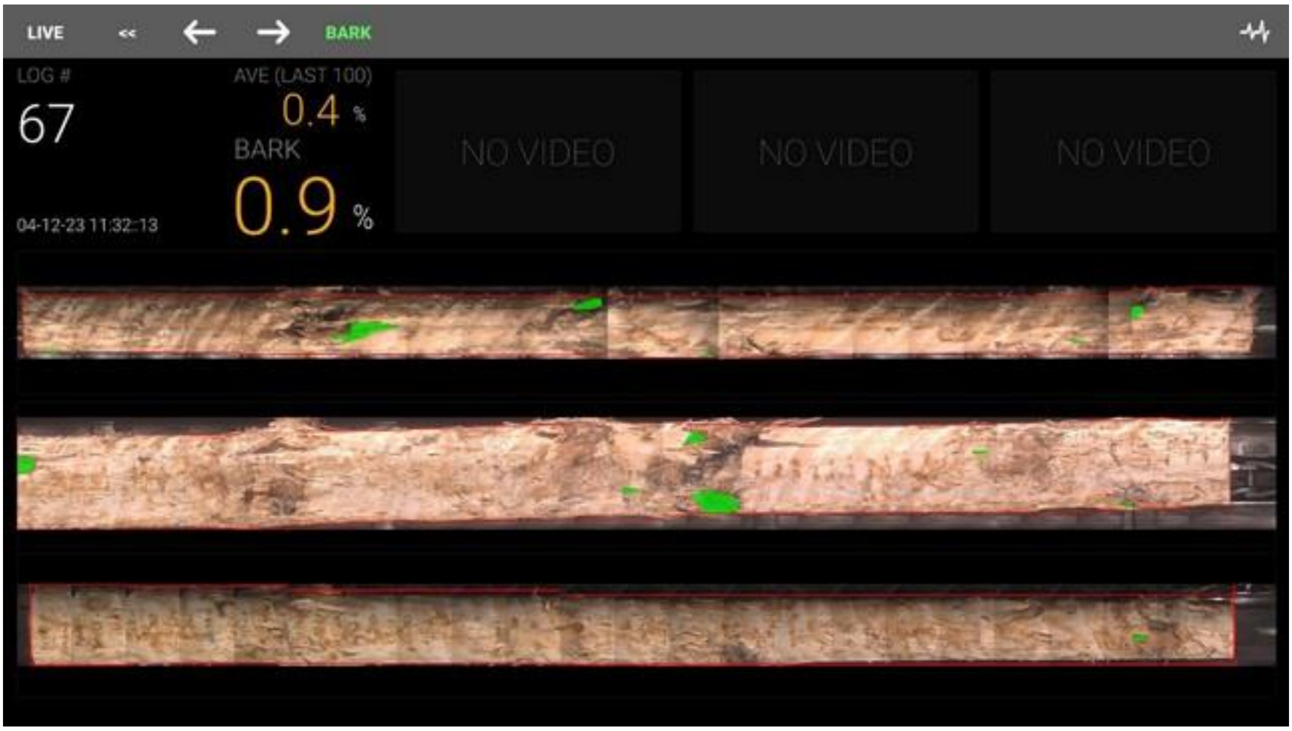
Log number 60 is shown below. This is an example where bark detection, log detection and stitching worked well. The bark detection model was able to distinguish between dark patches of dirt and bark with very few false negatives or false positives.



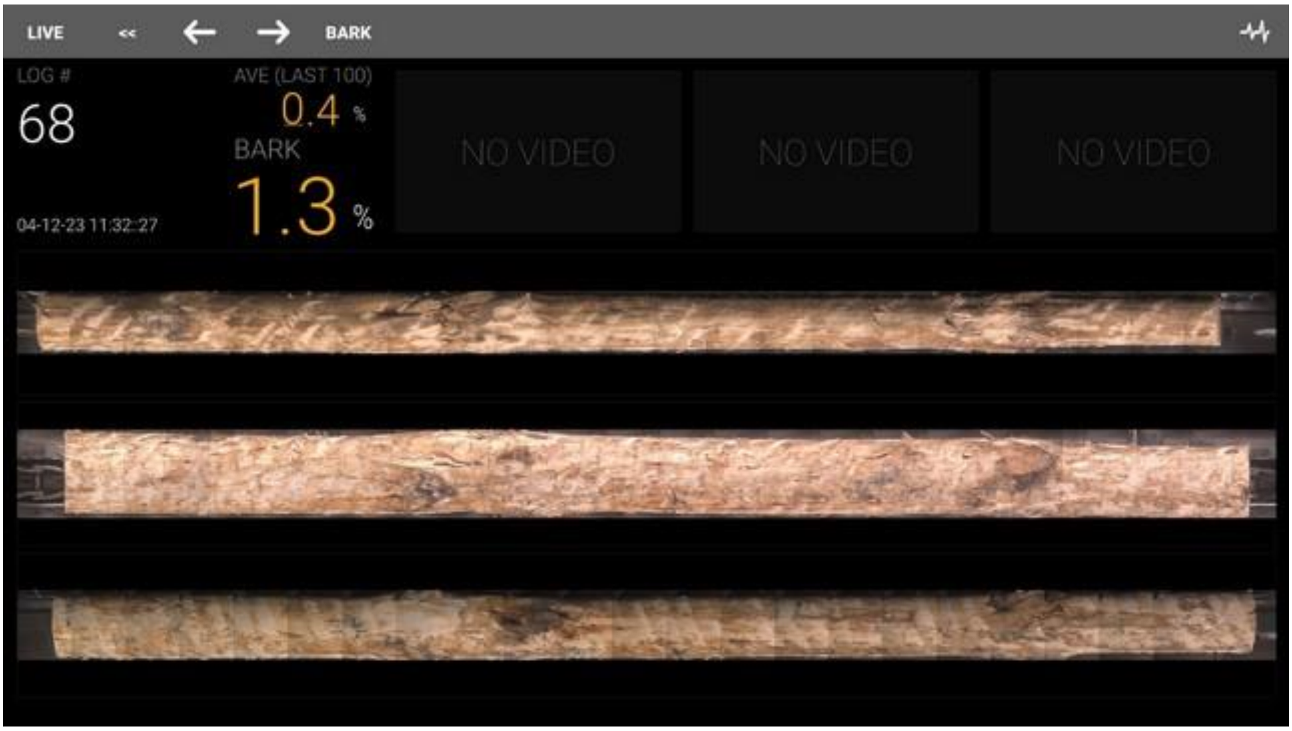


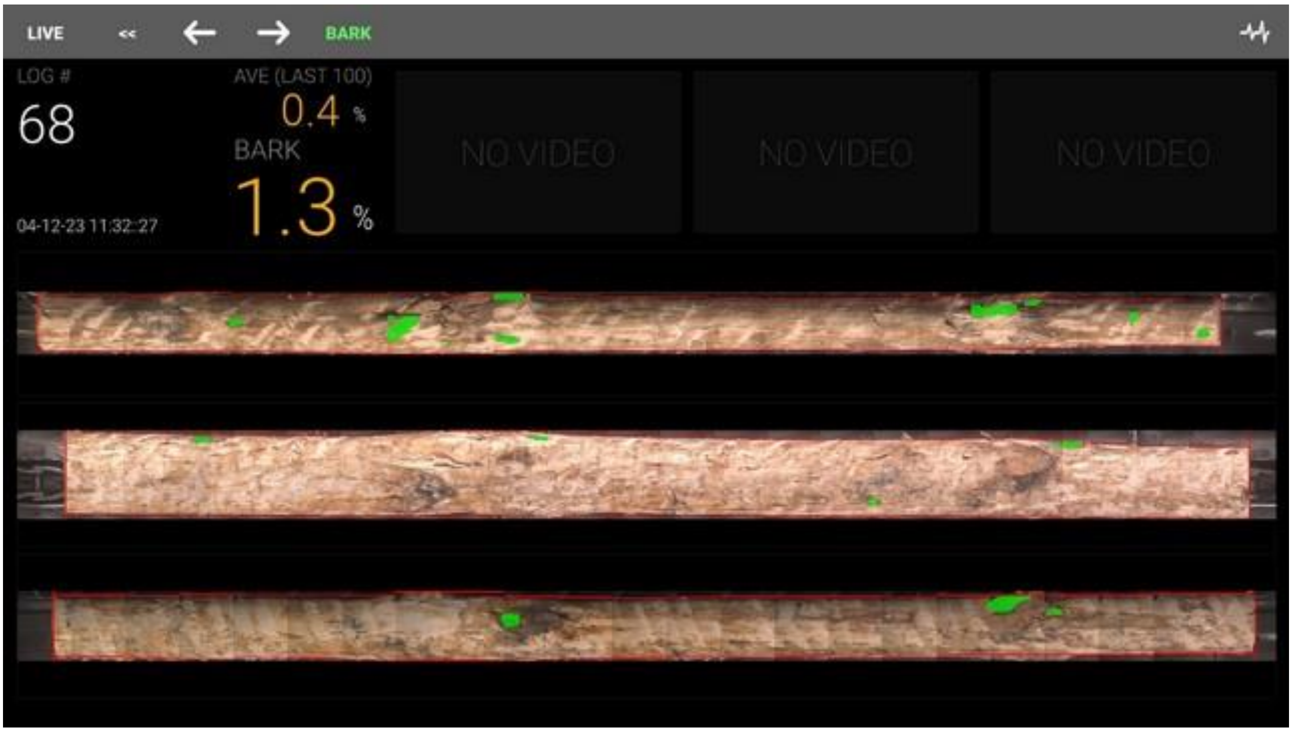
Another example is shown below.



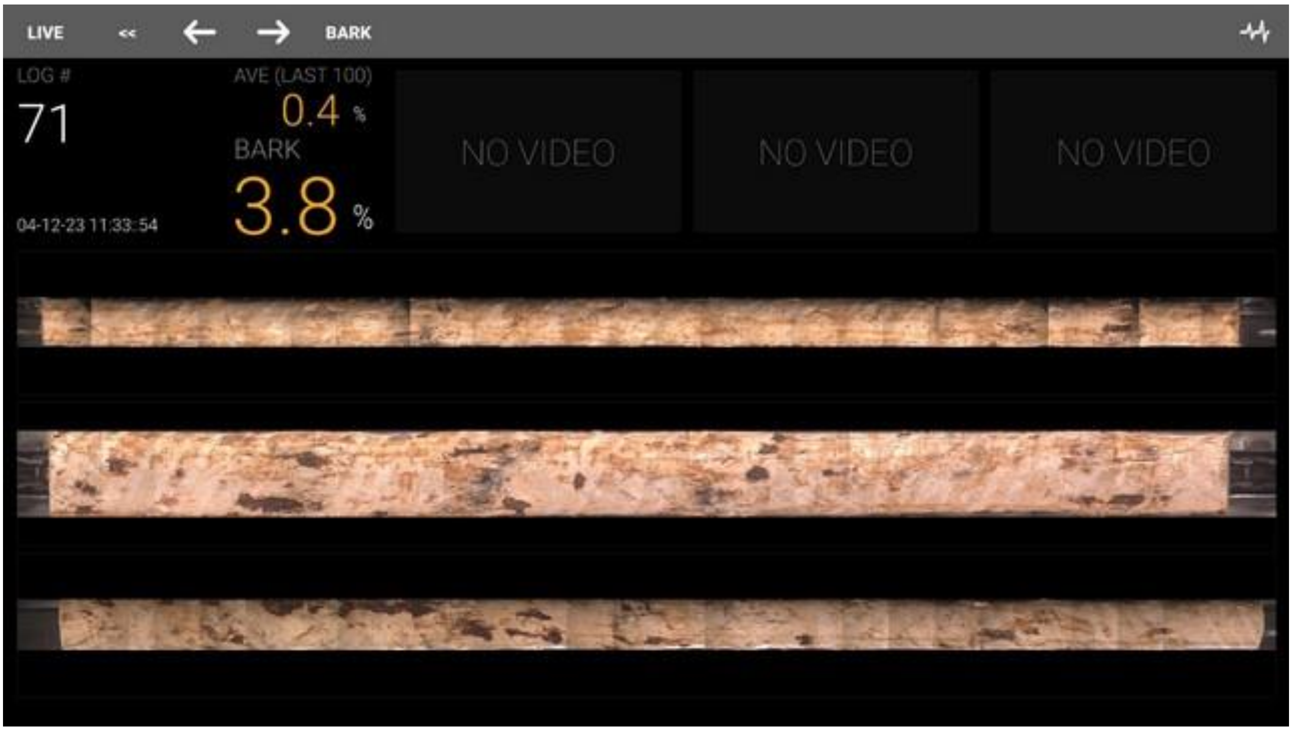


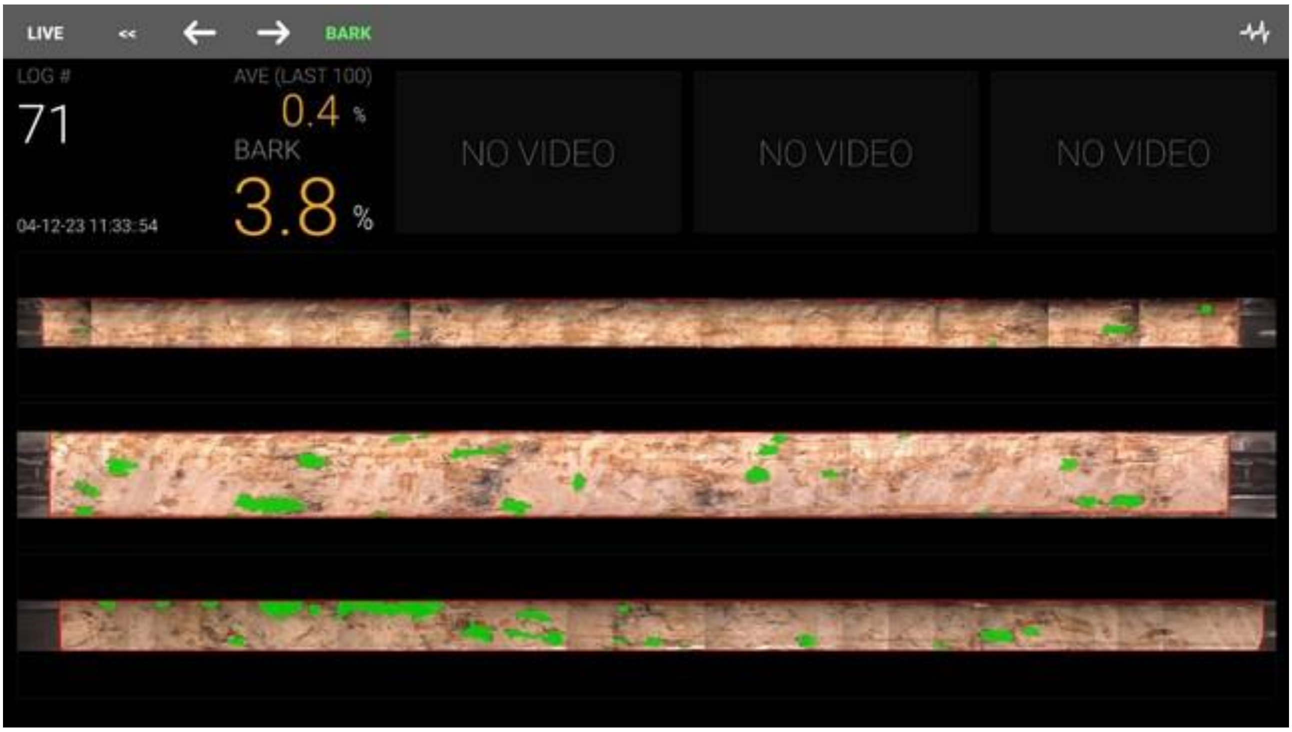
The example below shows false detection of bark where the cambium layer remains on the debarked log. Sufficient, uniform lighting is important to prevent false bark detection.



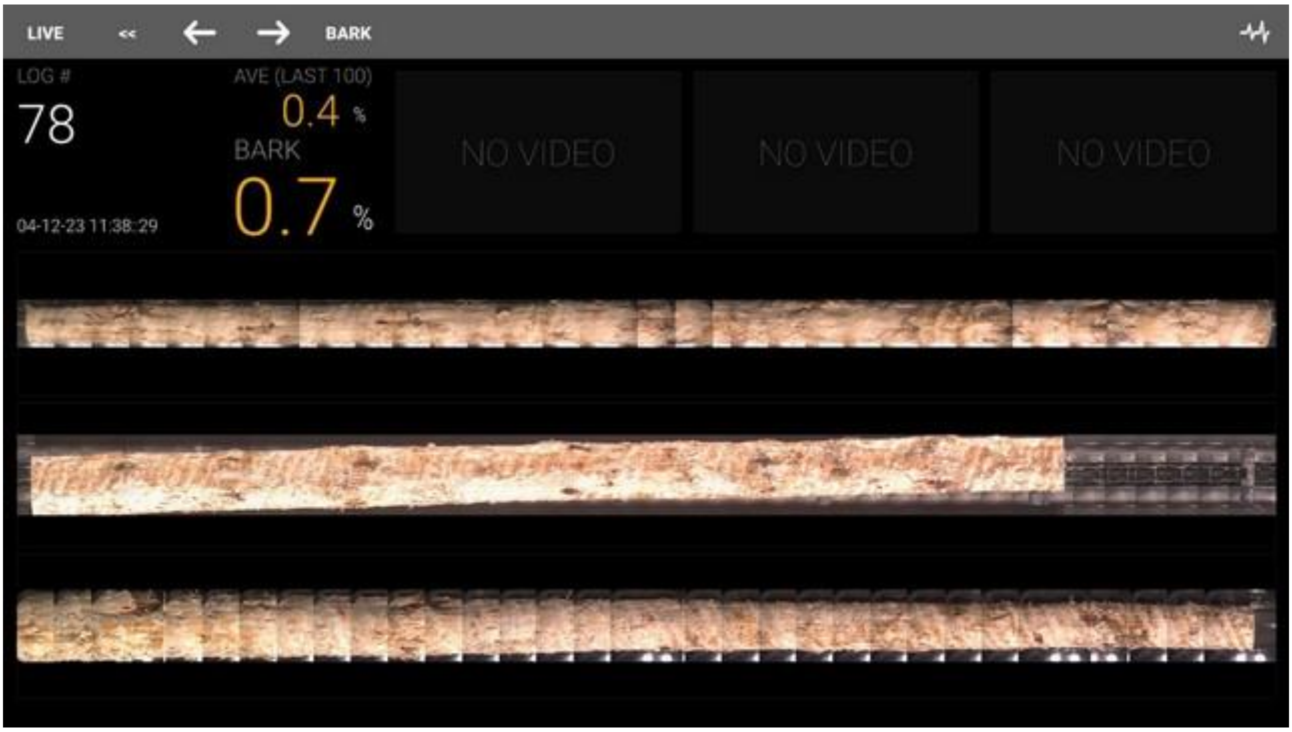


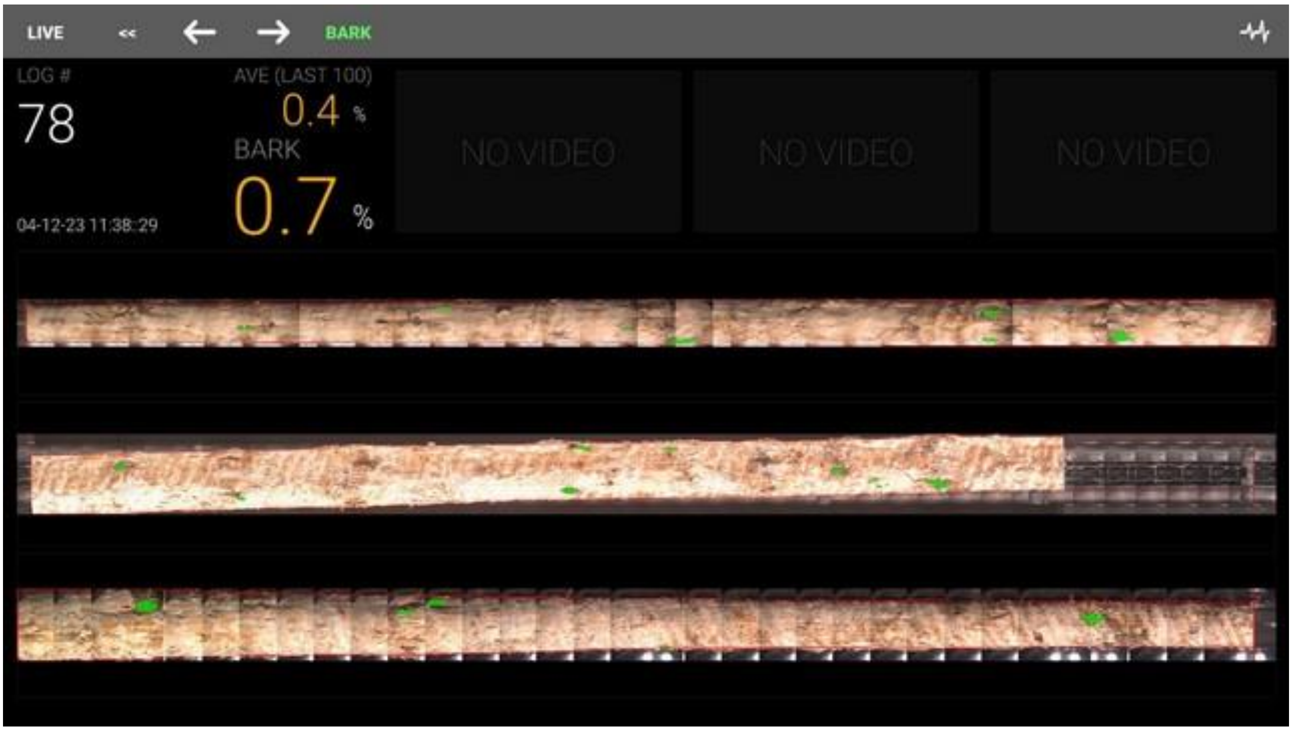
An example with many small patches of bark is shown below. Bark detection worked well in this example.



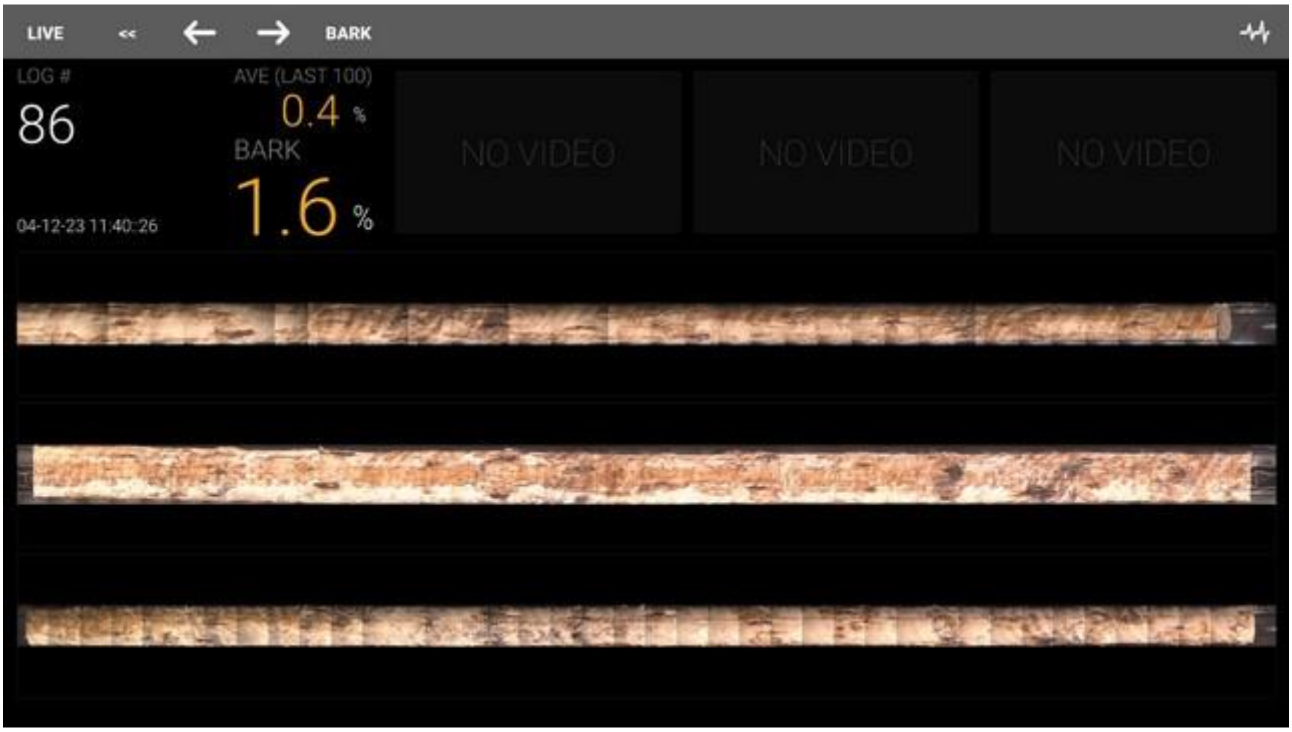


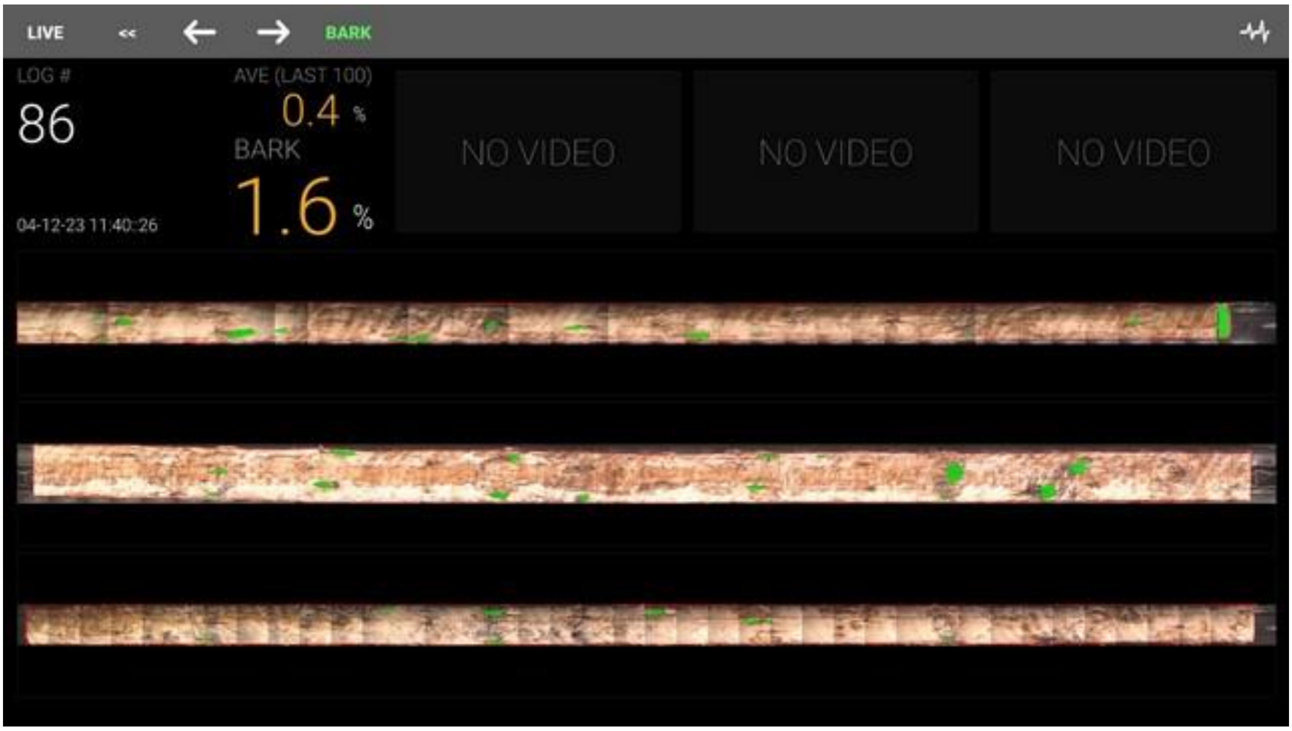
The example below has good bark detection, but the log area was overestimated.



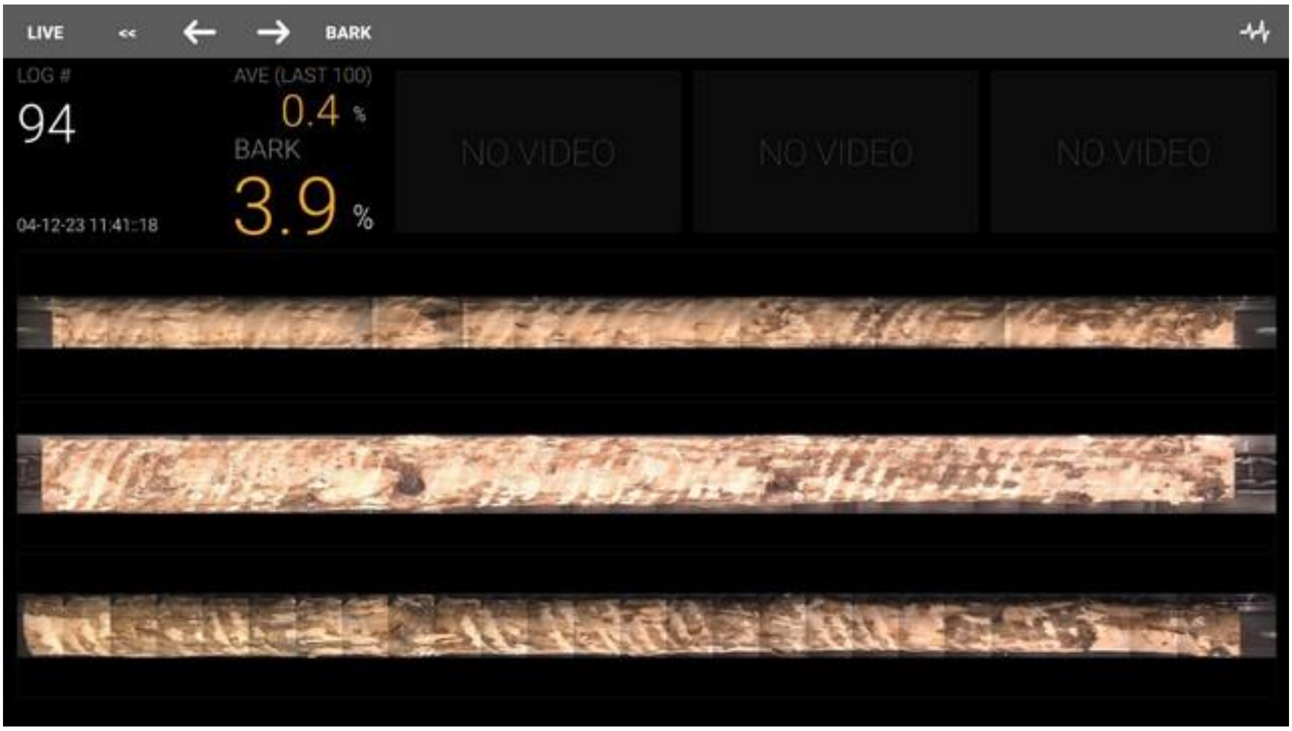


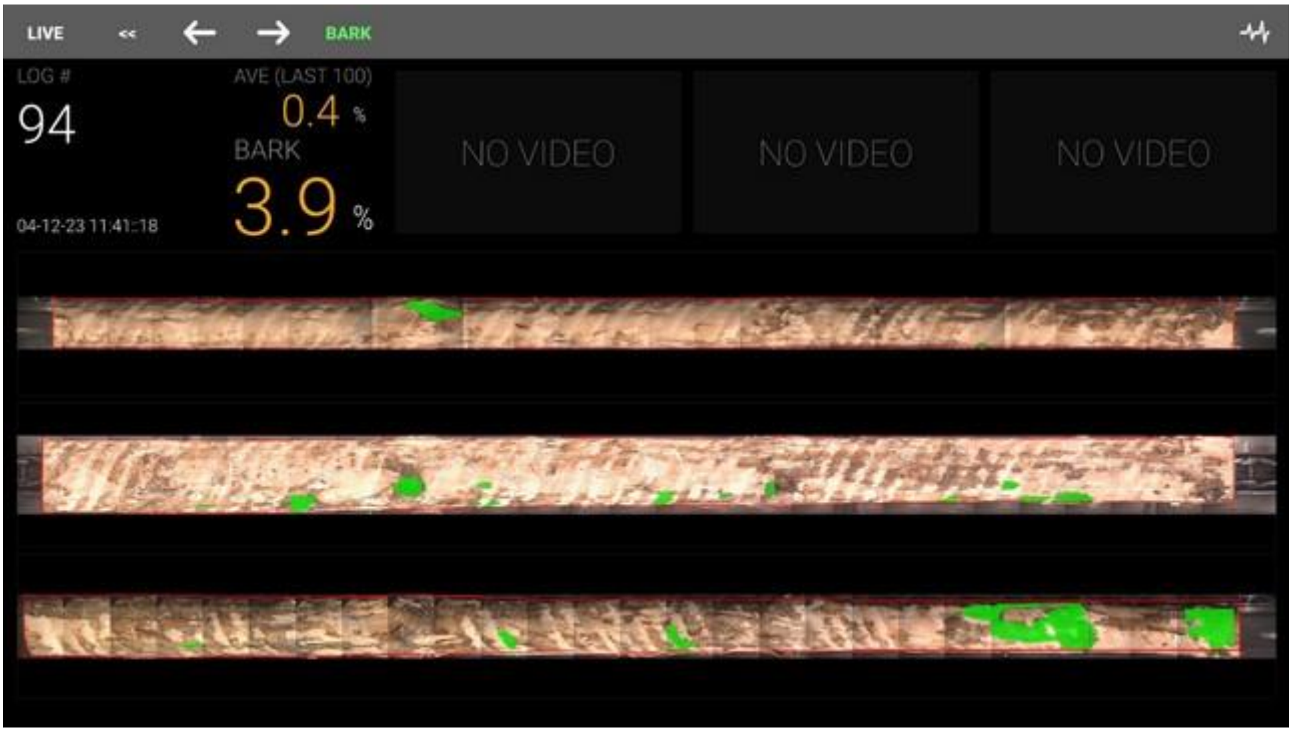
The example below has an area of false bark detection.



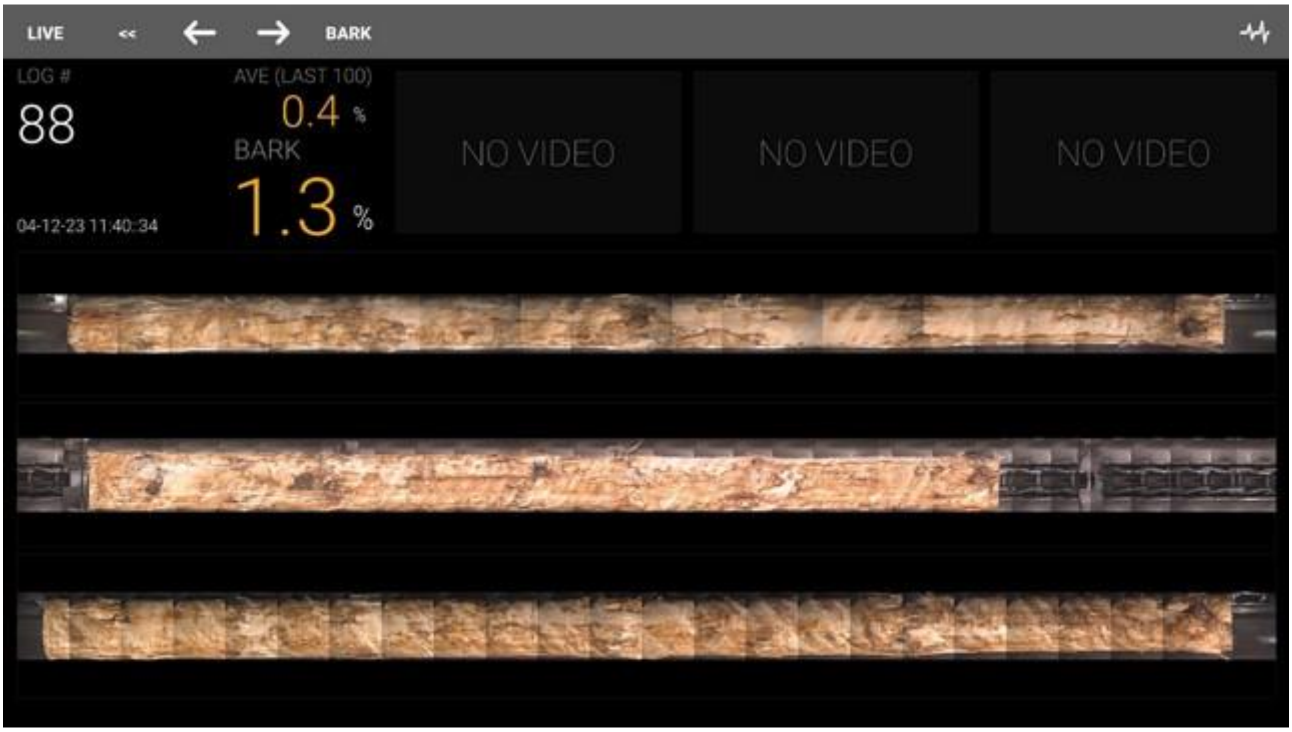


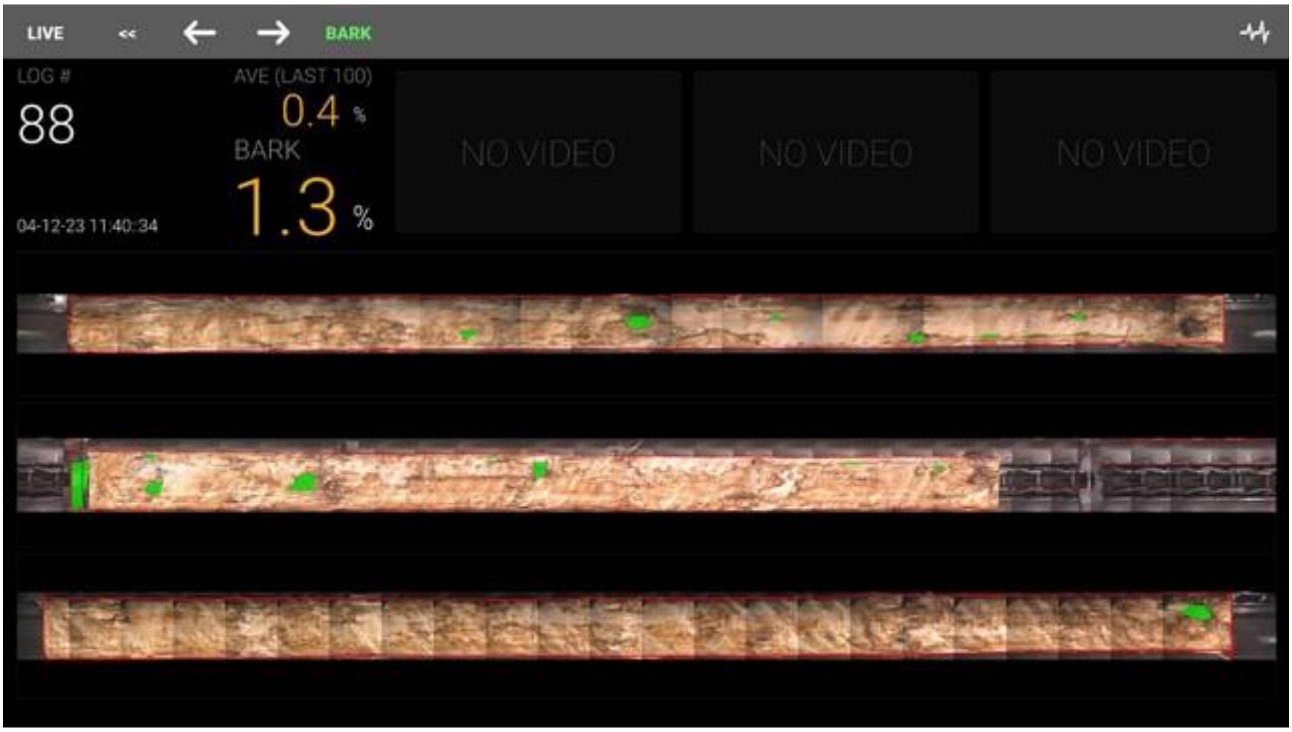
The example below is difficult as there are dark areas and large areas of cambium. Despite the difficulties, the tool performed well.



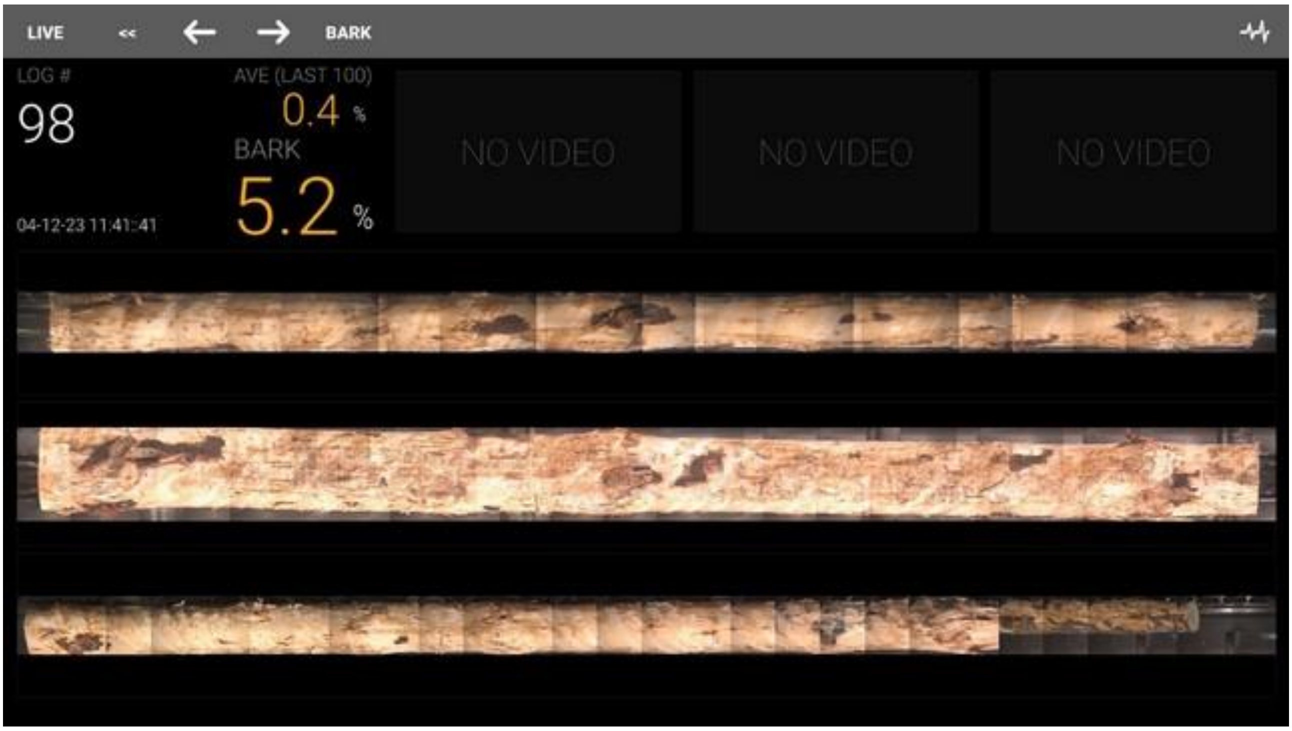


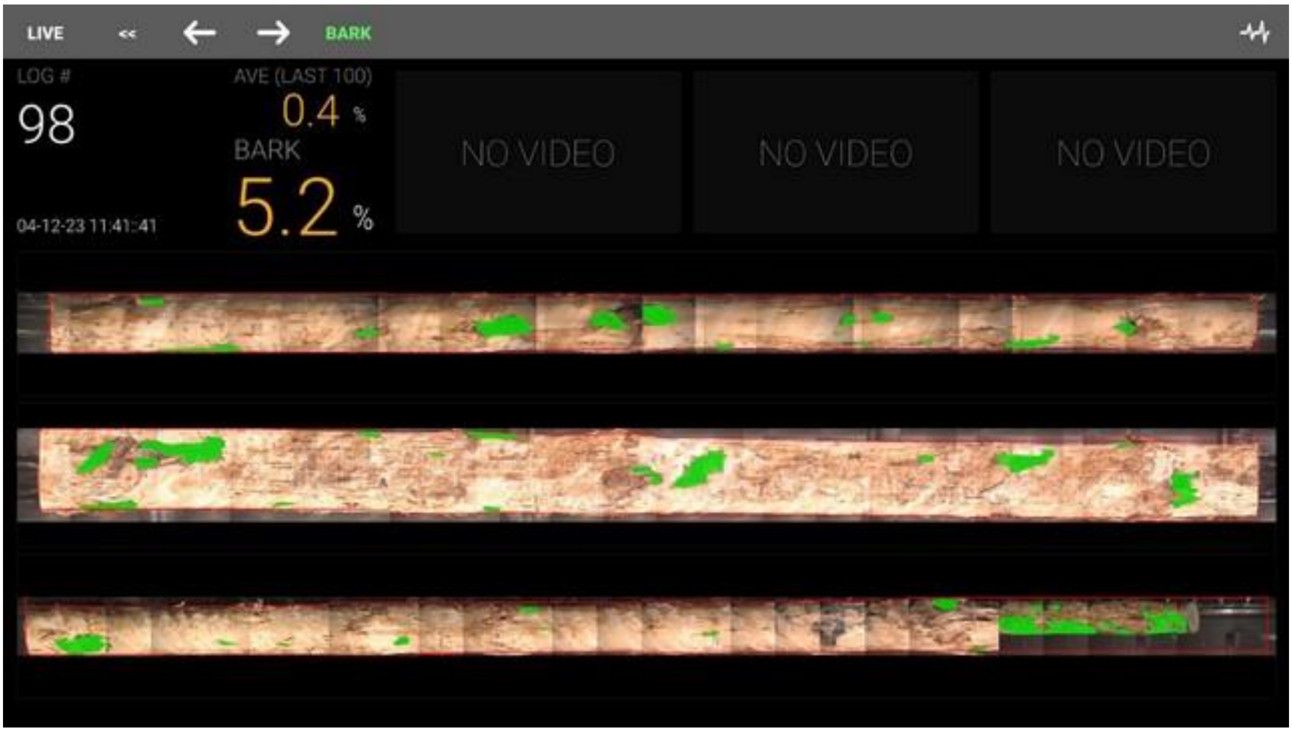
Another example of overestimation of log area is shown below. Camera 2 points at moving parts of the conveyor so the background is variable which makes background detection more difficult. A large patch of bark is also falsely detected just before the start of the log.



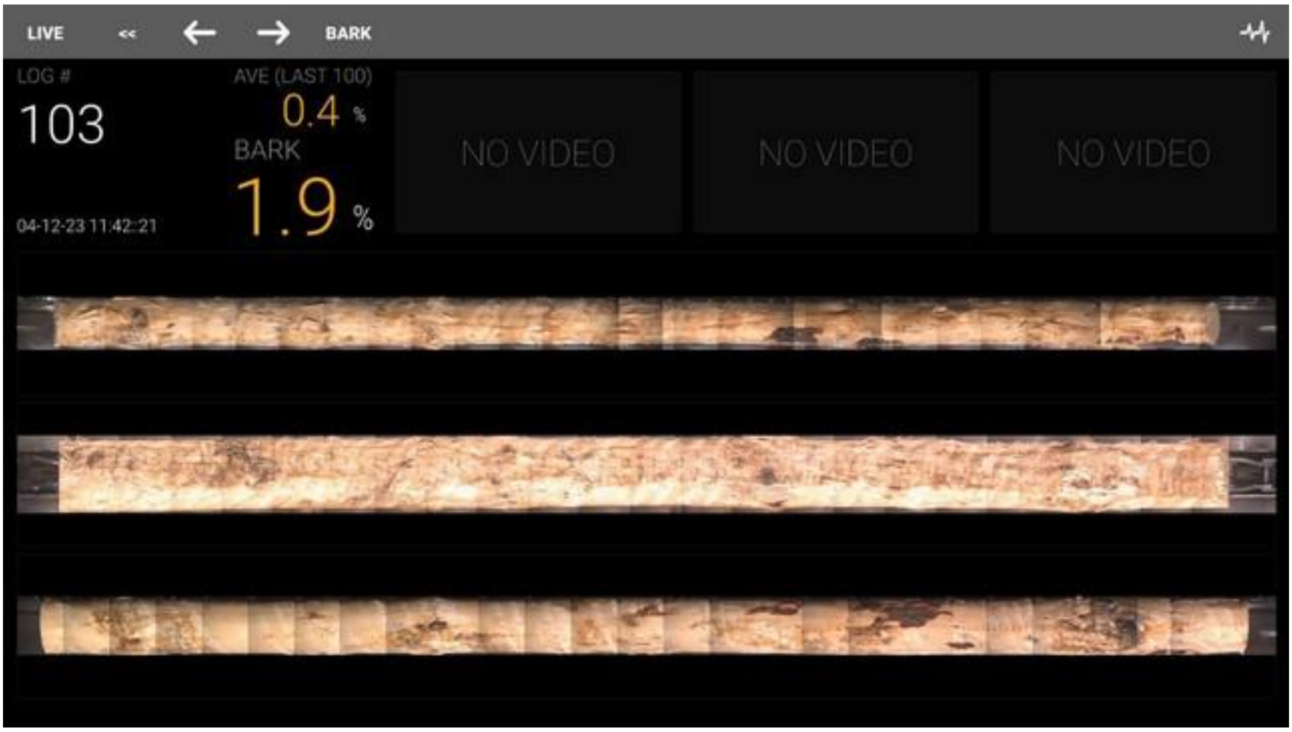


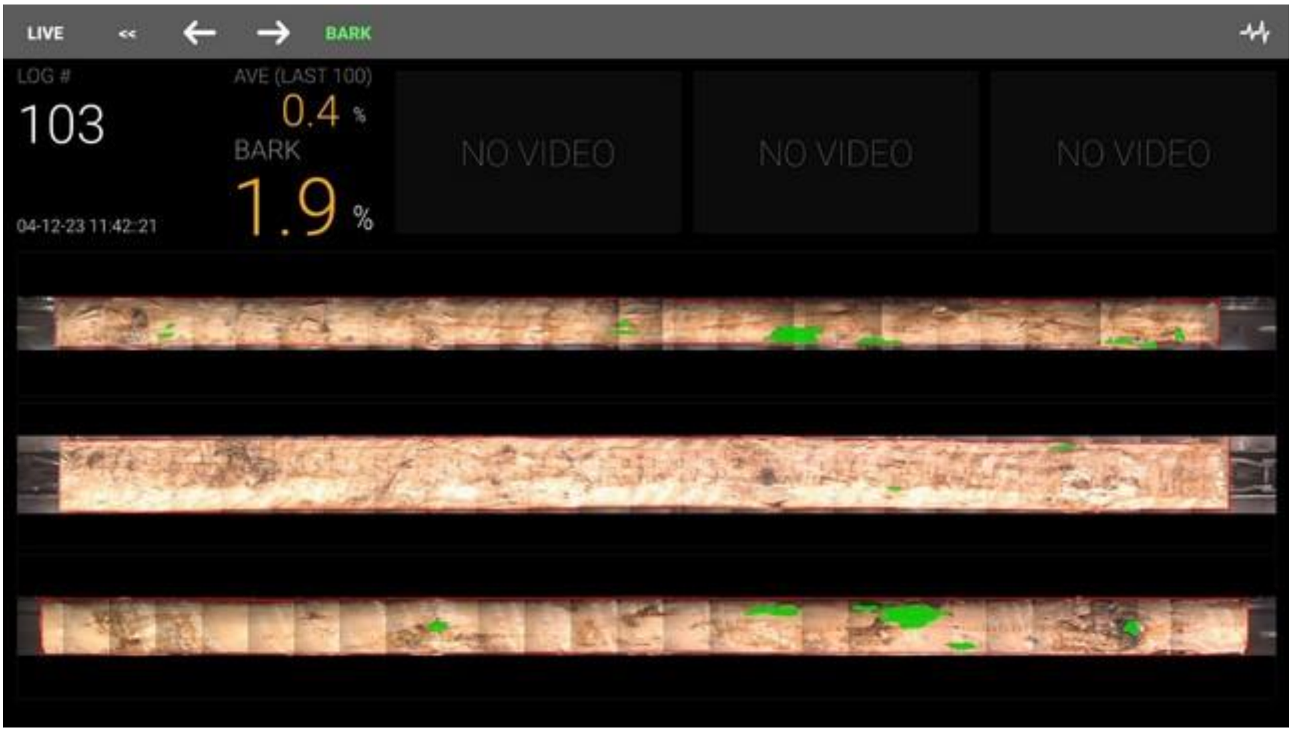
In the example below, something went wrong with the stitching. This could not be reproduced, and the exact cause was not known.





Another typical example is shown below.





Feedback from demonstrations

At the second demonstration of the prototype system on 4 December at KPP, the main points of feedback were:

- No long-term storage of log images is required.
- Data for “proportion of bark remaining” should be stored locally on a database at KPPy.
- There is interest in interfacing the system into existing control systems.
- There is interest in installing a bark detection system at KPP and at other sites.

At the demonstration of an early version of the prototype system on 10 October 2023, it was suggested that another metric be added, namely, the largest contiguous area of bark for each log.

Economic analysis

An economic analysis has been presented in another report (Milliken, *et al.,* 2023). The economic analysis considered a possible investment in a bark-detection system for a hypothetical fixed debarker. The cost of installing and running a bark-detection system was estimated. The benefits were estimated in terms of a reduction in the risk that debarking would be removed as a phytosanitary treatment. If debarking were not an available option, fumigation with full recapture would be required resulting in greater costs.

The Net Present Value was positive for the example that was considered in Milliken *et al.* (2024) and details may be found in that report.

DISCUSSION

Overview of discussion

The concept of a bark-detection system in a fixed facility was demonstrated at KPP. There were a number of lessons that were learned during the development and testing of the prototype system. These learnings are described, and recommendations are discussed below.

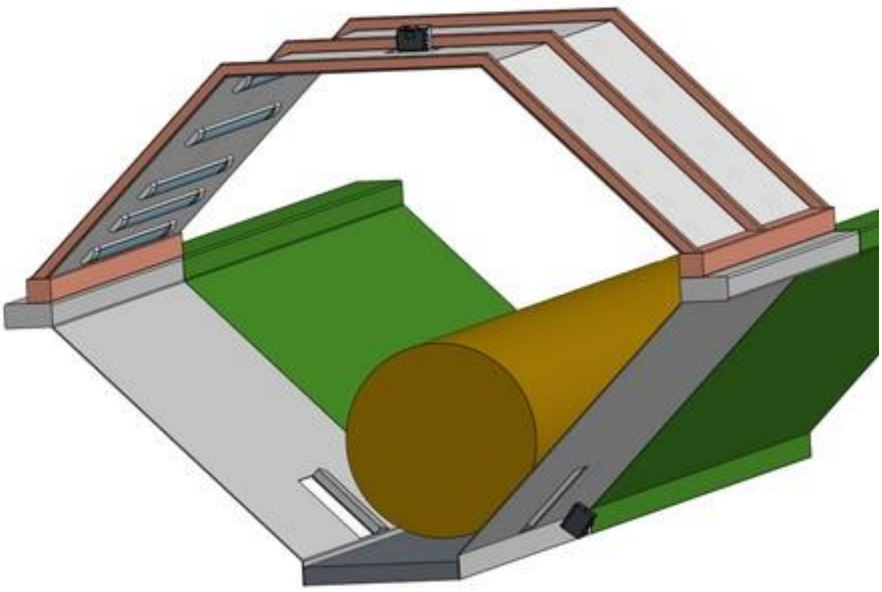
New metric

During the demonstrations, it was noted a new metric was under consideration; the largest contiguous bark area. The prototype was developed to measure the proportion of bark remaining which is a relative metric whereby no absolute measurement of size was required. Cameras 1 and 3, which view the log from the sides, can be around 100mm from the surface of the log or around 500mm from the log surface. Therefore, the apparent size of an area of bark could be very different if no allowance were made for proximity.

Proximity to the log could be estimated by using a view from the top camera or by using a displacement sensor such as lidar or triangulation. Using the view from the top camera would require very accurate estimation of log area.

Arch design

In an ideal situation, logs would pass through a ring with uniform lighting, a static background, and an unobscured view of the surface of the log. It would be preferable if the cameras were at least a metre from the surface to reduce the relative difference in distance between different parts of the surface. The ideal situation is not always practical in an existing facility. For the facility at KPP, the diagram below shows a debarked log moving along the conveyor. The concept is that LED lamps provide relatively even lighting, and the cameras are situated at a reasonable distance from the log. This diagram represents a reasonable compromise between the ideal and a design that would fit with existing structures in the plant.



A location where logs pass from one conveyor to another would provide an opportunity for an unobscured view of the entire surface of the logs and would allow cameras to be situated further from the surface of the logs.

Software

The software that runs on the central computer is currently not modular and has undergone a lot of development in a short period of time. If a commercial installation were to be done, it would be recommended that the code be refactored to make it more maintainable.

The software is currently limited by the fact that many of the detection and stitching parameters are not configurable. This means that adaptation to other facilities, other camera or lighting hardware, or even other locations on the conveyor belt would be more difficult. The source code should be refactored to expose these parameters via a configuration file that can be edited and copied to fine-tune the algorithm to different installations without the need to change the code.

The software currently writes files to disk as an intermediate step. While this was acceptable for a prototype, it should be avoided for a permanent installation. Summary data could be stored in a database for faster access and to avoid the need for an NFS share. Some of the intermediate frames could also be stored in RAM to improve efficiency. This would speed up the algorithm and reduce the disk space required by the system.

The stitching software currently runs in a single thread. There is potential to use multiple threads or possibly to use the GPU for stitching. If a displacement sensor were to be used, it could also be used to improve the quality of stitching.

A review of how the software manages memory should be undertaken as memory management may be the cause of some of the software instability that has been observed. If the conveyor stops when the log is in front of the camera, the process will crash and sometimes, it will lock up the operating system.

Robustness of computing hardware

Initially, the concept was to distribute processing across three cameras and the display module. After development of the software, it was found that log detection, bark detection and stitching were very computationally intensive. A central processing computer was added to provide extra processing power. The central computer was a tower desktop computer, so it is recommended that industrial hardware be used if a permanent installation were to be done. Hardware should be specified after any software changes have been implemented so that the processing power matches the requirements of the software.

Lighting

Short exposure times were required to prevent motion blur. This required very bright lights. Halogen lights were used for most of the development because they are bright and provide flicker-free broad-spectrum lighting. However, bright halogen lights generate a lot of heat so, for a permanent installation, non-flicker LED lights would be recommended. Around 50,000 lumens of lighting distributed around the log was found to work well.

User interface, storage of summary data and operator display

The user interface should receive data from a network accessible database. Feedback from the second demonstration was that local access to data would be sufficient and long-term storage of images would not be necessary. There may be a need to interface with other systems such as plant control systems. The architecture of the system should be designed to make this easy to achieve.

Recommendations for Commercialisation

If the system were to be commercialised, the following steps would be recommended prior to taking the system to market.

1. Confirm the metrics for bark-detection. They may include the largest contiguous area of bark.

2. If absolute area of bark is required, modify the hardware and/or software to determine distance from the log.

3. Build a light tunnel and collect new images. Although this is not absolutely necessary, it will provide a background that will be more consistent between installations at different sites. This will minimise the amount of customisation required for each installation.

4. Make improvements to software and refactor the code on the central computer, including:
    - Re-architect the code on the central computer to make it more modular.
    - Fix memory leaks.
    - Eliminate disk-writing operations.
    - Implement a database for data storage.
    - Add configuration parameters.
    - Interface the UI with the database.
    - Add self-pruning of data.
    - Build a web-interface to database.
    - Improve stitching algorithm.
    - Improve log-end detection.
    - Improve log-outline detection.
    - Add calculation for new metrics if required.

5. Integrate improvements and test on new images.

6. Test real-time operation in a debarking facility and specify requirements for computer hardware.

7. Prepare documentation for commercialisation. This will include documentation of the software to facilitate future software development and an installation guide for technicians.

8. Identify a commercialisation partner. Discussions could begin in parallel with the technical tasks listed above. The commercialisation partner will need to have capability to support, maintain and develop the system which will require a certain skills and technical expertise.

CONCLUSIONS

- A prototype bark-detection system was developed for a fixed facility. The system was tested and demonstrated at KPP.
- The system estimated percentage of bark remaining for each log.
- Software consisted of several elements including an algorithm to detect log ends, a convolutional neural network model trained for bark detection, an algorithm to detect log area and an algorithm for stitching the images together to recreate the surface of the logs.
- The system initially struggled with computational demands, requiring hardware upgrades and software optimizations to process data in real-time.
- The software worked adequately most of the time. However, certain situations created problems with robustness and the accuracy of the software. The system is not yet considered mature enough for a commercial installation.
- The addition of another metric, largest contiguous bark area per log, has been recommended by industry.
- Consistent lighting is important for good results. A tunnel with anti-flicker LED lights is recommended.
- Recommendations have been presented for the next steps to make the bark-detection system robust enough for commercial installations.
- An economic analysis was presented in another report (Milliken *et al.,* 2023) that considered the installation of a bark-detection system in a hypothetical debarking installation.

ACKNOWLEDGEMENTS

The authors wish to thank the following companies who assisted with the project:

- Timberlands Ltd at KPP for permitting the system to be tested at their facility.
- Paul Norris of Independent Verification Services Ltd for helping with markup of the training set of logs.

REFERENCES

Lightingplus (2023, 7 December). “PLU: 3293, MONS 60W white CCT 149CM LED batten”. https://www.lightingplus.co.nz/lighting/ceiling-lights/linear-lighting/mons-60-w-white-1490-mm-cct-led-batten

Milliken P., Lamborn D., and Brown I. (2023). “Economic Analysis of Automated Detection of Bark in a Fixed Debarking Facility”, FGR report H063.

Williams S., (2023, 7 December). “CFL's vs. Halogen vs. Fluorescent vs. Incandescent vs. LED”. https://www.homelectrical.com/cfls-vs-halogen-vs-fluorescent-vs-incandescent-vs-led.6.html