

Realtime Wood Log Diameter Measurement System

Laleth Indirani NehruKumar

Srinidhi Pattala

Nuntipat Narkthong

Abstract

Accurate measurement of wood log diameters is crucial in various industries such as forestry, timber processing, and wood manufacturing. Traditional measurement methods often involve manual labor and lack real-time capabilities, leading to inefficiencies and potential safety concerns for the workers. We focus on a problem in a specific process called “debarking” involved in the wood and timber processing industry. The heavy machinery used for debarking often comes with physical limitations, such as the minimum and maximum diameters of wood logs it can handle. If not inspected properly, wood logs can get lodged in the machine, causing it to break down, which in turn costs the industry time and money to get the system back running. To address these problems, we present a real-time wood log diameter measurement system using a live IP camera video stream at the debarking machine, leveraging an automated computer vision system and deep neural networks (DNNs). Evaluation results show that our systems can accurately detect wood logs in real-time at an 82.16% detection rate and determine the size of each wood log in inches with a 1.36% error rate.

1. Introduction

The pulp, paper, and timber industries are integral to the global economy, supplying raw materials for construction, packaging, and various applications. [1] estimates that worldwide paper and paperboard production will reach 490 million tons by 2020 from 390 million tons in 2009.

Fig. 1 provides a simple representation of the steps involved in paper production. The process of manufacturing paper involves several stages that transform raw materials into finished products. It begins with farming and harvesting trees, where logs are obtained. The logs are then transported to a paper mill, where, upon arrival, they are inspected manually. Next, the logs go through a debarking process to remove the bark and are subsequently chipped. These wood chips further undergo a process called pulping, which is refined to a slurry. This is the final material that is used for papermaking.

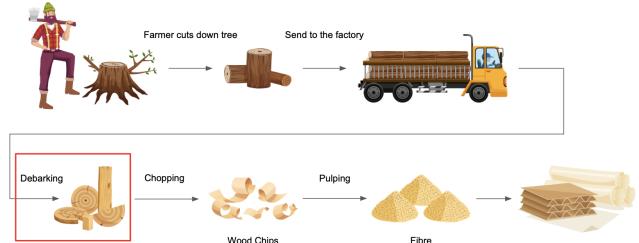


Figure 1. Overview of the paper production process.

In this work, we focus on the debarking process highlighted in red in Fig. 1 which is the first step to process the incoming wood log from the farmer. The traditional method of using human workers to inspect the wood log size suffers from the following limitations:

1. Inspecting at the back of the truck does not reveal the quality of the wood logs inside the truck, which can't be easily seen, as shown in Fig. 2.
2. It is expensive and hazardous to use humans to assess the size of wood logs as they are being unloaded from the truck to the debarking machine, as falling logs could cause severe injury.
3. The debarking machinery frequently breaks down because the wood logs are either too big or too little, as humans can currently only examine a small portion of the wood logs.

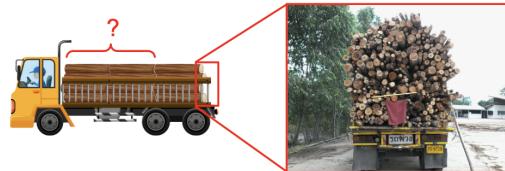


Figure 2. Problems with manual visual inspection before unloading the wood from the truck.

To address these problems, we developed a system to automate the inspection process using a video stream from the camera in front of the debarking machine. The objective of the model is to detect and track the grabbers that place the wood logs into the machine and to detect and compute



Figure 3. Screenshot from the live IP camera video stream from one cycle of operation, starting with grabbing wood logs from the truck, dropping the logs into the machine, and moving the grabber back to the truck to grab more logs.

the diameter of each log before it is fed into the machine in real-time.

2. System Requirement

2.1. Input Data

The input to our system is a live video stream from an IP camera (8MP at 10FPS) placed in front of the debarking machine. Fig. 3 shows a screenshot from the video stream from one cycle of operation, starting with grabbing wood logs from the truck, dropping the logs into the machine, and moving the grabber back to the truck to grab more logs.

2.2. Technical Requirements

1. The system should be able to process the video feed in real time.
2. The system should be able to operate 24/7 without interruption.
3. All hardware must be weatherproof, i.e., able to operate outdoors.
4. The proposed system should be compatible with or require minimal change to the current work process, e.g., how the grabber grabs wood logs from the truck to the machine.

2.3. Problem Formulation

According to the current workflow of the factory, the wood log unloading process begins when the truck parks at the debarker machine as shown in Fig. 4. Then, the grabber's driver grabs some wood logs from the truck, drops them in the debarker, and moves back to grab some more logs from the truck. The grabber's driver repeats this cycle until unloading all the wood from the truck. From the current workflow, we formulate the problems into three main stages:

1. *Grabber Detection* - The initial step in our wood log diameter measurement model involves the detection and tracking of the grabber within the frame. This is crucial as it allows us to determine the beginning and end times of each grab cycle. For this purpose, we trained an object detection model specifically tailored to recognize the grabber in order to determine its position in real time.
2. *Grab Cycle Segmentation* - Using the location of the grabber from the first step, the system determines the

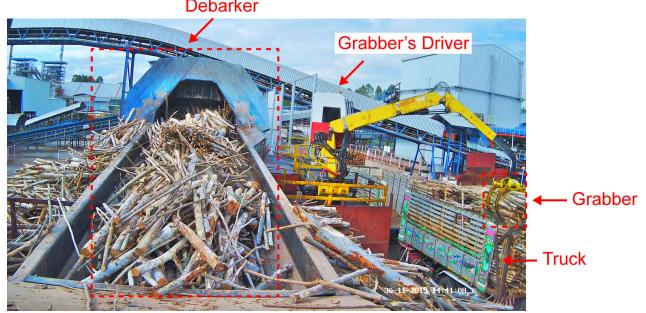


Figure 4. Overview of the wood logs unloading process.

begin and end times of each grab cycle and captures all frames during that period to send to the next stages to recognize the size of all wood logs in this grab.

3. *Wood Log Size Detection*: Once we determine the grab cycle, the system then passes the incoming frames for wood log diameter measurement. This stage involves two steps. a) Detect the wood log, and b) Determine the size in the real-world unit (cm/inch).

3. Design Overview

Fig. 5 shows the overall architecture of our system. To enable realtime processing of the video stream, we separate our system into two main phases: realtime phase which process each video frame at 10 fps and the near-realtime phase which process only selected frame from the first phase.

1. *Realtime Phase* - The real-time phase processes the live IP camera video stream and performs grabber detection as described in Sec. 3.1 and grab cycle segmentation as described in Sec. 3.2 to determine the begin and end times of each cycle. All frames during each cycle are saved to temporary storage for processing in the next phase.
2. *Near-realtime (offline) Phase* - The near-realtime phase processes all frames in each grab cycle to determine the size of all wood logs grabbed from the debarker in this cycle. First, the system filters invalid frames corresponding to other grabbers using the AprilTag [7, 10] as shown in Fig. 7. Then, it crops all frames and performs frame selection by performing wood log detection on all frames using the fast but less accurate model (coarse-grained de-

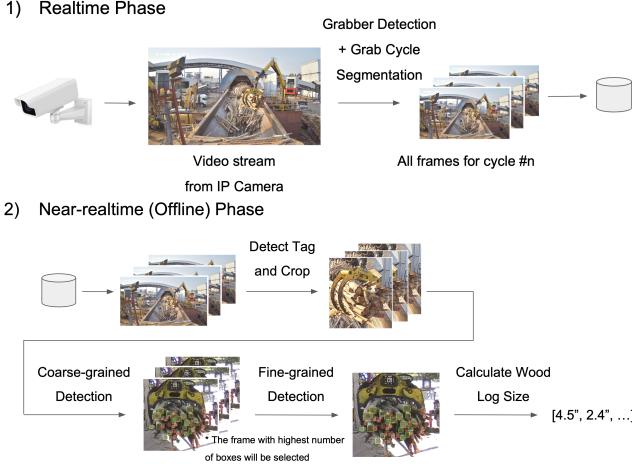


Figure 5. Overview system architecture.

tection) as described in Sec. 3.3. The frame that has the highest number of detections will be selected as a representative frame for this grab cycle. This frame will go through another round of detection using a slower but more accurate model (fine-grained detection). The result will be used to calculate the wood log size as described in Sec. 3.3.2.

3.1. Grabber Detection

A dataset for grabber detection is curated by extracting 300 frames from a video stream. This process involves selecting representative frames that capture various instances of grabber presence, ensuring the diversity needed for robust training. To prepare the dataset for training, each extracted frame is meticulously labeled. The annotations mark the specific class of interest, which, in this case, is the grabber. Fig. 6 illustrates a sample annotated frame, showcasing the bounding box around the labeled grabber.

The YOLOv4-tiny object detection model [2] is chosen due to its low resource requirement and fast inference time. To further improve the model's robustness and generalization capabilities, image augmentation techniques are employed during the training process. These techniques include adjustments to hue, exposure, and saturation, introducing variability to the dataset. Additionally, the input images are horizontally flipped, providing the model with additional perspectives to enhance its ability to detect grabbers in diverse orientations. This approach ensures the model's effectiveness in detecting grabbers of varying sizes and scales. We present the evaluation result of our grabber detection model in Sec. 4.1.

3.2. Grab Cycle Segmentation

Fig. 7 depicts how the video frame is divided into four zones: idle, pick, move, and drop, which correspond to var-



Figure 6. A screenshot of the labeling process for the Grabber detection dataset.

ious grabber actions. The system employs a cycle determination mechanism based on the grabber's state transitions described in Fig. 8, recognizing when it is idle, engaged in picking, in transit (move), or releasing (drop). After each grab cycle, a set of images captured during the move and drop zones are forwarded to the subsequent stage, as shown in Fig. 5. A timeout is included to avoid capturing too many frames when the grabber stays in the move and drop zones for longer than 10 seconds. The duration of the timeout is adjustable and is set based on our observation of the average time spent in each zone of the normal cycle.

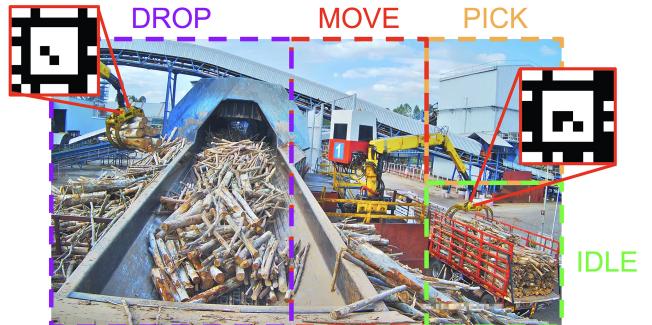


Figure 7. Mechanism of Drop, Move and Pick

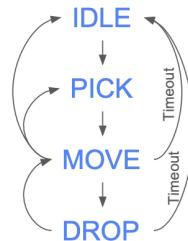


Figure 8. Phases of Segmentation

To ensure the correctness of the measurement, a mechanism is implemented in the near-realtime (offline) phase to

reject images of the grabber from the opposite side that may be inadvertently captured, as shown in Fig. 7. Each grabber has different tags attached to it. We adopt the AprilTag [7, 10] specifically the “tagStandard41h12” variant due to its robustness to different lighting conditions and partial occlusion. Some examples of tag designs are shown in Fig. 9 and all tag designs can be found in [link](#).

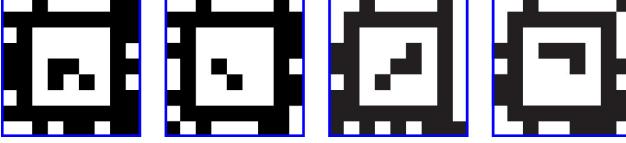


Figure 9. Sample AprilTags [7, 10] (tagStandard41h12) designs.

Note that we opt to use the YOLOv4-tiny model to detect the grabber in the real-time phase for two reasons. First, the AprilTag detection algorithm, although very efficient, takes too much time on 8 MP images (3840×2160 px). We could resize the video frame for tag detection, but doing so will reduce the number of pixels occupied in the image by the already small tag, making the tag much harder to detect. Thus, detecting the tag in the second phase after cropping the image is easier and much faster. Second, being able to detect grabbing activity without using the tag allows us to develop an alert system to alert users when the tag is missing or dirty (Sec. 5 and Fig. 19).

3.3. Wood Log Size Detection

Our wood log size detection algorithm consists of two steps. First, the bounding box of each wood log is detected using a deep learning-based object detector, as described in Sec. 3.3.1. Then, the size in real-world units is calculated by comparing the size in pixels of the box with the size of the tag, as described in Sec. 3.3.2.

3.3.1 Wood Log Detection

We collect a dataset comprising 230 images from a video stream at multiple grabbers during daytime and nighttime and manually label each wood log. Then, we fine-tuned the Faster R-CNN object detector [8] and the Atrous Faster R-CNN object detector [3] to detect the wood log based on the models provided on TensorFlow Model Zoo ¹ which were pre-trained on the COCO dataset. The Atrous Faster R-CNN object detector [3] is chosen in addition to the normal Faster R-CNN model due to its superior small-scale object detection capability. During training, the Image augmentation techniques, including random cropping, random flipping, and random color adjustment, are applied to introduce variability and enhance the model’s ability to general-

¹Both pre-trained models can be downloaded at [link](#)

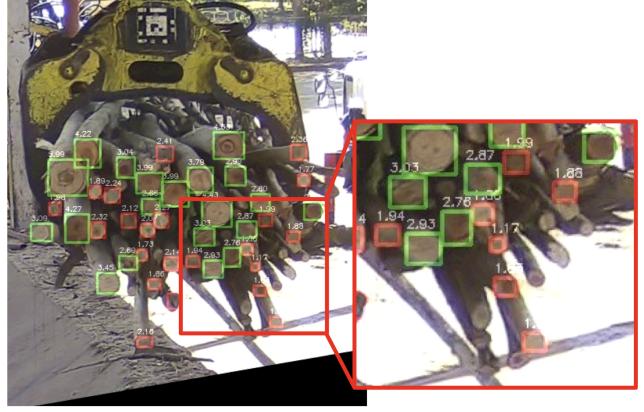


Figure 10. A sample image shows some small wood logs that may not be detected by the model. Green boxes indicate wood logs > 2.5 inches. Red boxes indicate wood logs ≤ 2.5 inches.



Figure 11. An image illustrates how the image is split into four overlapped patches.

ize across different scenarios. The evaluation results of our wood log detection models are presented in Sec. 4.2.

Fig. 10 illustrates one problem of the model, which is the low detection rate for small wood logs, which we believe is due to the small area it occupies in the image. To overcome this problem, we split the image into four overlapped patches, as shown in Fig. 11, and performed inference on each patch. The evaluation results of our wood log detection models on patched input images are presented in Tab. 2 and Fig. 14.

3.3.2 Log Size Calculation

To calculate the size of each wood log in real-world units, we compare the bounding box size of the detected wood log against the known size of an AprilTag, which serves as a reference with a fixed dimension of 14×14 cm as shown in Fig. 12. Since most wood logs are not perfectly circular, we opt to calculate the wood log size based on the widest dimension of the bounding box. Eq. 1 shows the equation to calculate the wood log size compared to the tag size.

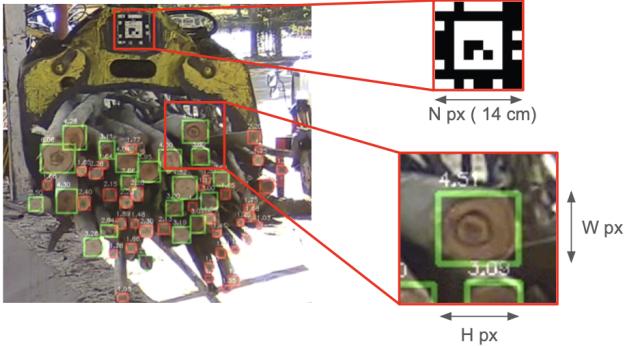


Figure 12. A diagram shows how the log size is calculated by comparing the tag size with the wood log bounding box. Green boxes indicate wood logs > 2.5 inches. Red boxes indicate wood logs ≤ 2.5 inches.

$$\text{LogSize(cm)} = 14 \times \text{MAX}(W, H)/N \quad (1)$$

where W and H are the width and height in pixels of the wood log's bounding box, respectively, and N is the width of the tag in pixels. Note that there could be some error in the final log size as the wood logs are closer to the camera than the tag. However, this error can be approximated and corrected by applying a constant offset to the final measurement result.

4. Evaluation Results

4.1. Grabber Detection

Tab. 1 at two input resolutions: 240x240px and 416x416px. The large model (416x416px) is chosen because the smaller model's runtime is just slightly faster while the accuracy drops by 3.5%. Some sample result images are shown in Fig. 13, which validates that our model performs well even when the grabber is partially obscured (top-left), in side view (top-right), and at night (bottom).

Input Resolution	Detection Rate (%)	Runtime (sec) ¹
240x240px	91.20	0.01686
416x416px	94.72	0.01723

Table 1. Accuracy of the grabber detection. ¹All runtimes are measured on Apple MacBook Air M2 w/ 16GB RAM

4.2. Wood Log Size Detection

4.2.1 Evaluation Metrics

We use the following five metrics to evaluate our model:

1. *Percent of correct boxes.* The percent of correct boxes is calculated by dividing the number of predicted boxes



Figure 13. Some sample result images show that our grabber detection model performs well even when the grabber is partially obscured (top-left), in side view (top-right), and at night (bottom).

that can be matched to ground truth boxes by the total wood log count in the ground truth label. To match each predicted box with the ground truth, we rank the IoU value between each predicted box and all actual boxes, reject the value less than 0.1, and select the box with the highest IoU as being matched.

2. *Percent of incorrect boxes.* The percent of incorrect boxes is calculated by dividing the number of predicted boxes that can't be matched to any ground truth box by the total wood log count in the ground truth label.
3. *Percent of missing boxes.* The percent of missing boxes is calculated by dividing the number of ground truth boxes that can't be matched to any predicted box by the total wood log count in the ground truth label.
4. *Average size error.* The average size error is calculated by subtract the size of all matches actual boxes with size of all matches predicted boxes and compute the average. The size of boxes is calculated based on method described in Sec. 3.3.2.
5. *Average runtime.* The runtime of the model for one inference.

4.2.2 Results

Tab. 2 presents the evaluation result of our wood log detection models. Compared to the Faster R-CNN, the Atrous Faster R-CNN improves the percent of correct boxes by 1.88%, but the runtime is slower by 8.2X, from 0.425 seconds to 3.48 seconds on the Apple MacBook Air M2. Despite a marginal improvement in the proportions of correct, incorrect, and missing boxes (a few percent each), the histogram in Fig. 17 illustrates a substantial enhancement in the distributions of size errors, as indicated by the peak values between -10% and 10%. This observation is further supported by a sample result image in Fig. 14, in which we can clearly see that the detection boxes of the center image (Atrous Faster R-CNN) fit the wood log significantly better

Model	% Correct	% Incorrect	% Missing	Avg. Size Error	Runtime (sec) ¹
Faster R-CNN (Inception v2) [8]	58.68	5.16	41.31	-0.787 ± 16.38	0.425
Faster R-CNN (Inception Resnet v2, Atrous) [3]	60.56	2.5	39.43	1.317 ± 15.24	3.48
Faster R-CNN (Inception Resnet v2, Atrous, 4 patches) [3]	82.16	23.47	17.84	1.356 ± 17.19	15.14

Table 2. Accuracy of the wood log detection model. ¹ All runtimes are measured on Apple MacBook Air M2 w/ 16GB RAM

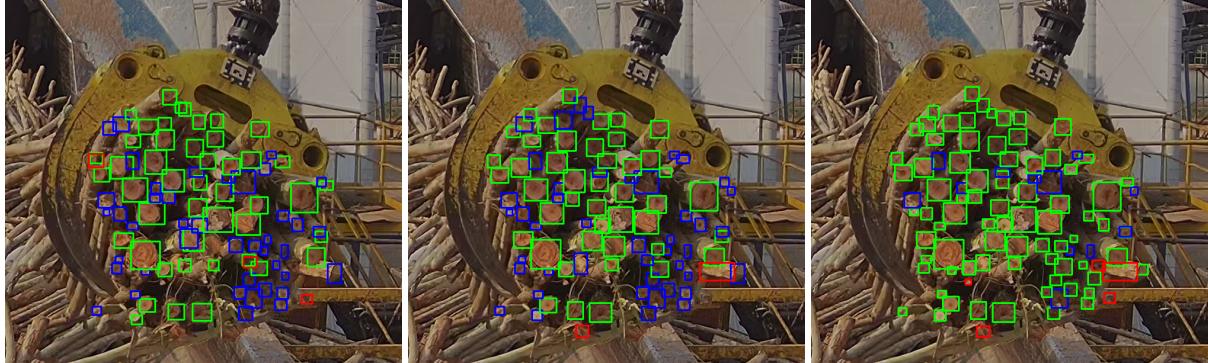


Figure 14. Sample result images from the Faster R-CNN (left), Atrous Faster R-CNN (center), and Patch-based Atrous R-CNN (right). Green boxes indicate correct predictions. Blue boxes indicate ground truth boxes that the model can't detect. Red boxes indicate incorrect predictions.

than the left image (Faster R-CNN). Nevertheless, there are still lots of missing boxes, which are shown in blue in Fig. 14, that remain undetected in both models.

The Atrous Faster R-CNN with 4 patches improves the detection of small wood logs by splitting the input images into 4 overlapped patches, as shown in Fig. 11. When compared to the Atrous Faster R-CNN, patch-based inference improves the percent of correct boxes by 21.6% at the cost of a higher incorrect prediction rate and 4.35X slower runtime. Fig. 14 (right) shows that patch-based inference enables the system to detect most small wood logs, as evidenced by the significant reduction in the number of blue boxes compared to the left and center images. This observation is further supported by Fig. 15, which shows the histogram of wood log size, where we see a higher number of detections for small wood logs (less than 3 inches). To demonstrate the performance of patch-based Atrous Faster R-CNN, Fig. 16 presents additional result images with different backgrounds and lighting conditions.

One concern of patch-based inference is the increase in the percent of incorrect predictions, which is worsened by 20.97%. However, we observe that most incorrect predictions occur in challenging scenes such as Fig. 20 and Fig. 21, thus not affecting the detection quality in real-world scenarios such as Fig. 16. Due to time constraints, we left the formal quantitative analysis of this issue as future work.

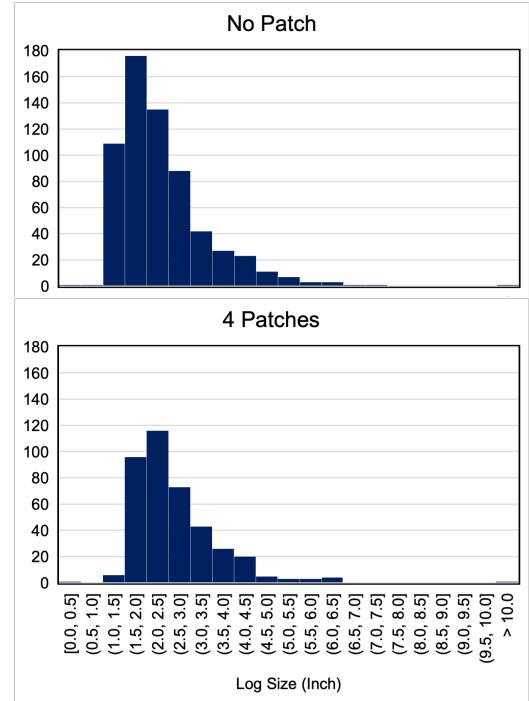


Figure 15. Histogram of the wood log size of the Atrous Faster R-CNN and the Patch-based Atrous Faster R-CNN model.

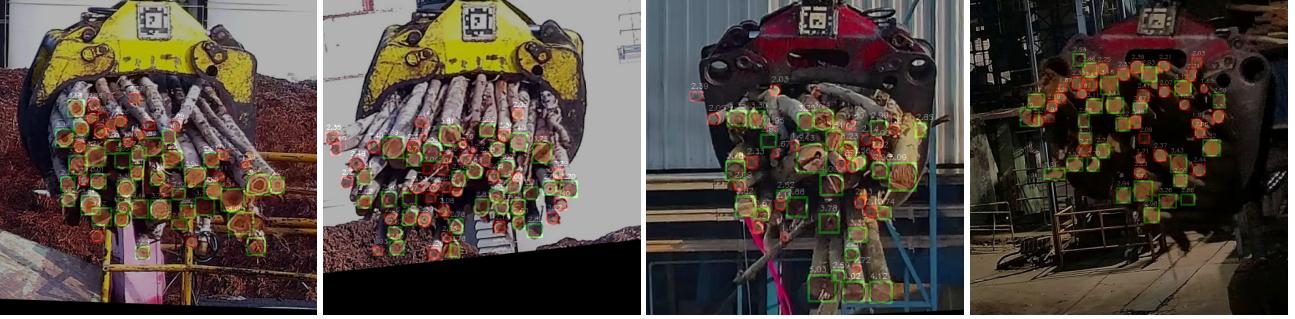


Figure 16. Sample result images from the Patch-based Atrous Faster R-CNN model.

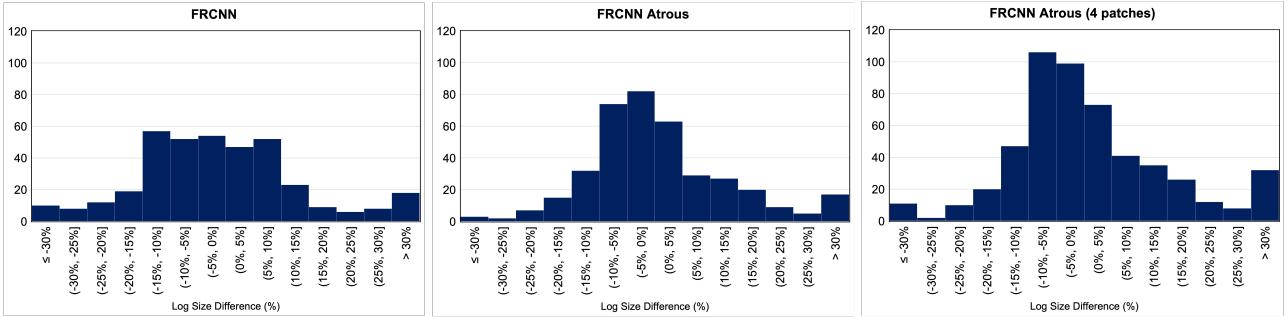


Figure 17. Histogram showing distribution of size error in percent of the three wood log detection models.



Figure 18. Sample result images show the grabber not facing the camera.



Figure 19. A sample image shows a dirty tag caused by dust and oil from the grabber.

5. Challenges and Future Works

In this section, we list some challenges that are affecting the detection quality and discuss some possible mitigations for future work.

- Grabber Angle.* The grabber may not face the camera directly in some cases, as shown in Fig. 18, rendering the wood undetectable. This problem could be solved by adding more cameras to capture the video from multiple directions, or we could improve our cropping algorithm to enlarge the crop area based on grabber orientation.
- Tag Occlusion.* Regularly cleaning the AprilTag is necessary because it can get exposed to dirt from the environment and oil from the machine, as shown in Fig. 19. To mitigate this problem, we designed the grab cycle

segmentation algorithm (Sec. 3.2) to not rely on AprilTag, enabling us to develop an alert system that prompts maintenance actions when occlusion is detected while the grabber is moving. Nevertheless, we recommend implementing regular cleaning protocols for the AprilTag to ensure its visibility and prevent system downtime.

- Poor lighting conditions.* Lighting conditions might lower detection rates, especially at night, as shown in Fig. 20. To mitigate this problem, more lights should be installed at the debarker. Alternatively, we could explore infrared illumination as an option to capture video at night.
- Bad Image Quality.* The image can become blurred, as shown in Fig. 21, due to camera vibration and too fast grabber movement, especially at night when the shutter



Figure 20. Sample result images show poor lighting conditions.



Figure 21. Sample result images show bad-quality images.

speed is slower due to low light. Exploring better optics and camera sensors or lowering grabber movement speeds at night may improve image quality. Additionally, we could install more light or explore infrared illumination at night.

6. Related Works

Prior research has been focused on detecting the size of stationary wood logs using several image capture devices and algorithms. For instance, Yu et al. [11] used a ZED stereo camera with a Mask R-CNN model to measure log diameter, achieving 98.2% recognition rate and 5.3mm average diameter error on the HAWKwood dataset [4]. Fig. 22 shows sample images from the HAWKwood dataset, which consists of larger wood logs that stack more neatly compared to our grabber images in Fig. 16. Samdangdech et al. [9] proposed a system to detect wood log size at the back of the truck using SSD-based object detection [5] and Fully Convolutional Networks (FCNs) semantic segmentation network [6]. Evaluation was done on a proprietary dataset consisting of images at the back of the truck, similar to Fig. 2, achieving an average accuracy of 94.45%.

To the best of our knowledge, this paper presents the first work that performs wood log size detection on the moving target (grabber) using an ordinary RGB camera in a real-world setting (different lighting conditions, background, wood log distance, etc.).



Figure 22. Sample images in the HAWKwood dataset [4].

7. Conclusion

This paper presents an automated system to measure wood log size during the wood log unloading process from the truck to the debarking machine. The system processes the real-time IP camera video stream to detect the grabber and the begin and end times for each grab cycle. It then detects all wood logs and calculates their size in real-world units. This information is for the factory in order to further classify the wood log to avoid debarking machine breakdowns when processing wood log sizes. Evaluation results show that our systems can accurately detect wood logs in real-time at an 82.16% detection rate and determine the size of each wood log in inches with a 1.36% error rate, surpassing the capability of a human worker while eliminating the risk of accidents caused by having workers near the machine.

Acknowledgment

We would like to thank the anonymous wood pulp factory in Thailand's northeast province for sharing their IP camera video recording for our project and for agreeing to install the Apriltag for a few days.

References

- [1] Pratima Bajpai. Advances in recycling and deinking. In *Recycling and Deinking of Recovered Paper*, pages 1–19. Elsevier, Oxford, 2014. 1
- [2] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020. 3
- [3] Tongfan Guan and Hao Zhu. Atrous faster r-cnn for small scale object detection. In *2017 2nd International Conference on Multimedia and Image Processing (ICMIP)*, pages 16–21, 2017. 4, 6

- [4] Christopher Herbon. The hawkwood database. *arXiv preprint arXiv:1410.4393*, 2014. 8
- [5] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. Ssd: Single shot multibox detector. In *Computer Vision – ECCV 2016*, pages 21–37, Cham, 2016. Springer International Publishing. 8
- [6] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. 8
- [7] Edwin Olson. Apriltag: A robust and flexible visual fiducial system. In *2011 IEEE International Conference on Robotics and Automation*, pages 3400–3407, 2011. 2, 4
- [8] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015. 4, 6
- [9] Noppawat Samdangdech and Suebskul Phiphobmongkol. Log-end cut-area detection in images taken from rear end of eucalyptus timber trucks. In *2018 15th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, pages 1–6, 2018. 8
- [10] John Wang and Edwin Olson. Apriltag 2: Efficient and robust fiducial detection. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, page 4193–4198. IEEE Press, 2016. 2, 4
- [11] Chunjiang Yu, Yongke Sun, Yong Cao, Jie He, Yixing Fu, and Xiaotao Zhou. A novel wood log measurement combined mask r-cnn and stereo vision camera. *Forests*, 14(2), 2023. 8