A comparative study of the effectiveness of using popular DNN object detection algorithms for pith detection in cross-sectional images of parawood

Wattanapong Kurdthongmee*
School of Engineering and Technology, Walailak University 222 Thaibury, Thasala, Nakornsithammarat, 80160, Thailand

ARTICLE INFO
Keywords: Computer science Wood pith detection Parawood pith location Deep neural networks object detection SSD MobileNet You-only-look-once

ABSTRACT
The location of pith in a cross-sectional surface of wood can be used to either evaluate its quality or guide the removal of soft wood from the wood stem. There have been many attempts to automate pith detection in images taken by a normal camera. The objective of this study is to comparatively study the effectiveness of two popular deep neural network (DNN) object detection algorithms for parawood pith detection in cross-sectional wood images. In the experiment, a database of 345 cross-sectional images of parawood, taken by a normal camera within a sawmill environment, was quadrupled in size via image augmentation. The images were then manually annotated to label the pith regions. The dataset was used to train two DNN object detection algorithms, an SSD (single shot detector) MobileNet and you-only-look-once (YOLO), via transfer learning. The inference results, utilizing pretrained models obtained by minimizing a loss function in both algorithms, were obtained on a separate dataset of 215 images and compared. The detection rate and average location error with respect to the ground truth were used to evaluate the effectiveness of detection. Additionally, the average distance error results were compared with the results of a state-of-the-art non-DNN algorithm. SSD MobileNet obtained the best detection rate of 87.7% with a ratio of training to test data of 80:20 and 152,000 training iterations. The average distance error of SSD MobileNet is comparable to that of YOLO and six times better than that of the non-DNN algorithm. Hence, SSD MobileNet is an effective approach to automating parawood pith detection in cross-sectional images.

1. Introduction
The wood pith is located roughly at the centre of the cross-section of a wood stem. It has two main uses. First, the wood quality can be evaluated using pith location (Fallah et al., 2012). High quality wood tends to have the pith at the centre. Second, it is used to locate regions of soft wood with low quality along the wood stem that must be removed. It has been experimentally shown that the knot within a stem, whose surrounding regions are soft, can be completely removed by locating the pith locations at both ends of the stem and pushing it through a vertically aligned bandsaw. The soft region of the wood is likely to be within a single board at the centre of the wood stem. This technique is used in several small to medium-sized sawmills in South and Southeast Asia for processing parawood (Hevea brasiliensis) to produce wood parquets and other wood products. Locating the wood pith in the cross-section of wood stems originating from Europe is fairly easy because they are cultivated under controlled conditions to make them grow vertically (i.e. by removing their branches). Their pith locations are likely to be at the centre of the stems. Parawood is very different because it is cultivated to produce latex, and several branches and leaves are required for sufficient photosynthesis. Only after about 20 years, when the tree begins to produce insufficient amounts of latex, is it cut down and replaced. Thus, parawood is considered a by-product. Because of the way it is grown, its pith location tends to deviate from the cross-section centre. However, parawood has been successfully used as an alternative to other tropical woods extracted from natural forests (Joerg Balsiger and Bahdon, 2000). Furthermore, parawood has proven to be very versatile in furniture manufacturing and the wood-based panel industry. Where forests are scarce, particularly in South and Southeast Asia, the use of parawood as fuel continues to increase pressure on natural forest resources. Yield increase from efficient processing could significantly improve parawood...
utilization and prevent natural forests from being cut down while reducing pollution caused by use of the waste as low-cost fuel.

Many studies have proposed using the geometric relationships of the annual rings within the cross-section of wood stems to locate the pith (Krahnenbühl et al., 2012; Boukadida et al., 2012; Kurthongmee et al., 2018; Kurthongmee and Suwanarath, 2019). Once the boundary of an annual ring has been extracted from an image, image processing routines are used to create segments from neighbouring pairs of pixels that form the boundary. The normal lines of the segments are then generated. This procedure produces a set of normal lines, and their intersection points can be used to approximate the pith location. Several algorithms have been presented to effectively extract annual rings or a wood stem boundary and approximate the pith location. Some algorithms require sophisticated and expensive image capturing instruments (e.g. CT or X-ray scanners). Others are suitable only for images with imperfection-free cross-sections; rot, dirt, snow, or defects from the cutting process must first be removed (Hanning et al., 2003; Longuetaud et al., 2004; Österberg et al., 2004). Unfortunately, using these algorithms and instruments to analyse the cross-sectional surface of parawood is complicated. Its surface is likely to contain several imperfections, mainly caused by diffused latex, mounds, or defects from the cutting process. Removing them to make the pith clearly visible for computer vision methods requires labour intensive mechanical or chemical pre-processing stages.

Deep neural networks (DNN) have been successfully applied in many application domains to either classify (Canziani et al., 2016) or detect objects (Cheng et al., 2016a, 2016b, 2019) in an image. Regarding wood manufacturing and processing, it has been successfully used for defect classification, including the wood pith classification, in lumber images (Hu et al., 2019). The RestNet18 network (He et al., 2015) was investigated using an 80:20 ratio of training to test images. Four datasets that consist of two common lumber defects (cracks, insect defects, sound knots, and unsound knots), wood textures, and wood species were used and classification accuracies of 98.16%, 93.32%, 96.64%, and 99.50% were obtained. Here, the classification result only indicates the existence of defects within the image. This is in contrast to a detection result that additionally identifies the regions around the defects. To our knowledge, no studies have been published on the topic of wood pith detection.

The main objective of this study is to investigate the effectiveness of applying DNN to detect parawood pith within a cross-sectional image captured using a normal camera. None of the samples were treated to remove defects from the cross-sectional surface. Additionally, all trained weights of the model, for both SSD (single shot detection) MobileNet and YOLO (you-only-look-once) are provided for transfer learning.

2. Materials and methods

2.1. Parawood pith image dataset preparation

In this experiment, 345 cross-sectional images of parawood from sawmills in south Thailand were collected. The images were captured using a digital camera (iPhone 6, Apple Inc.) in the working environment of sawmills under various lighting conditions. This was to ensure that a variety of images were obtained to train the DNN. Neither cleaning using chemical substances nor mechanical polishing were applied to the parawood surfaces. The images include the whole area of the surfaces. To increase the number of suitable images for DNN training, these original images were processed using the open-source image augmentation tool Image Augmentor (https://github.com/mdbloice/Augmentor). It generated new images by changing the angle, saturation, exposure, and hue of the original images. After image augmentation, the total dataset size was increased fourfold to 1,380 images. The data set was then manually annotated using the open-source LabelImg tool (https://github.com/tzutalin/labelImg). For each image within the dataset, the pith location was identified under the guidance of the log scalers from the sawmills. A rectangle was superimposed on the original image such that it was centred exactly at the pith location and covered at least three annual rings surrounding the pith. This also automatically produced the ground-truth pith location of the cross-section within the image (i.e. the location is at the centre of the rectangle). Figure 1 illustrates two sample images of cross-sectional surfaces after annotation. Annotation files were produced for each image in the dataset. Because two DNN object detection algorithms were employed in our experiment, two annotation file formats were required: one for SSD MobileNet and one for YOLO. The labelling tool is able to generate these formats. Note that our dataset is available upon request.

All 1,380 images and corresponding annotation files were then split into two datasets for training and testing. Different configurations were used with ratios of 60:40, 70:30, 80:20, and 90:10. The images and their corresponding annotation files were randomly selected and assigned to these datasets using an open-source Python script called split-TrainingAndTesting (https://www.learnopencv.com/training-yolov3-deep-learning-based-custom-object-detector/). Figure 2 summarizes the image dataset preparation process described above.

2.2. DNN training

After the training and test data were prepared, they were used to train the DNNs. DNNs are well known for being slow to train on a moderate-performance computer system. The use of a graphics processing unit (GPU) is therefore recommended to increase the resources of a computer system. Alternatively, cloud services, which have re-configurable hardware components, can be used to accelerate execution. In this study, we used Google Cloud Platform (GCP) resources during the training stage. We used the following GCP configuration: two virtual CPUs: 13 GB memory; a single NVIDIA Tesla K80 GPU; and a Linux operating system.

Because it took a very long time to train the DNN even on the GCP with GPU acceleration, a termination criterion was required. Note that during the DNN training stage, the procedure periodically performs testing and saves the current weights of the model trained with respect to the test data to minimize a loss function. The loss was therefore used as the training termination criteria.

2.2.1. SSD MobileNet configuration

We employed the ‘AI engine’ of the GCP to train the SSD MobileNet. This engine was specially created to simplify platform creation in TensorFlow DNN applications. The normal procedures of the AI engine (Google, 2019) contained in its configuration file were followed to create all training jobs, which were submitted to the AI engine. The three primary details for the training jobs described in the configuration file are as follows.

1. The transfer learning method was implemented using the following pre-trained model: ‘ssd_mobilenet_v1_0.75_depth_300 x 300_coco14_sync_2018_07_03’.
2. The default hyperparameters and DNN model configuration were used for training (see Table 1).
3. No built-in image augmentation function was used because the dataset had already undergone image augmentation (as detailed in Section 2.1).

2.2.2. YOLO configuration

The GCP does not provide any engine to automate YOLO training. It was necessary to install the Darknet framework provided by YOLO researchers (Redmon and Farhadi, 2018; Redmon et al., 2016) and execute it directly from the Linux command shell of the GCP. The YOLO training configuration was as follows.

1. The Darknet source code for YOLOv3 (Redmon and Farhadi, 2018) was employed. It was recompiled to run on the GCP shell using the GPU.
2. Transfer learning with the ‘darknet53 checkpoint’ was used.
3. A Tiny-YOLOv3 network with 23 layers (Redmon and Farhadi, 2018; Redmon et al., 2016) was used in all experiments (see Table 1).

4. All parameters defined in the Darknet configuration file were unchanged except for the image augmentation option, which was disabled for the reasons given above.

2.3. Detection performance metrics

The application of DNN object detection to the pith detection task differs from general object detection. This is because the aim of detection is to find a pith in the image as well as identify its location. This location is used in further processing (i.e. to judge if the wood stem is of high quality or to guiding a wood stem turning machine). Therefore, the detection performance metrics are the detection rate and the distance from the ground truth (the location error). The former measurement is the ratio of the number of the images in which the pith is detected by DNN object detection to the total number of images, whereas the latter is defined by

\[ D = (x_d - x_g)^2 + (y_d - y_g)^2 \]  

(1)

where \((x_d, y_d)\) is the pith location estimated by DNN object detection and \((x_g, y_g)\) is the ground truth pith location. The average location error of all detectable piths within a test dataset of \(N\) images is defined simply as

\[ \mathcal{J} = \frac{1}{N} \sum_{i=0}^{N-1} D_i \]  

(2)

The training and test datasets described earlier, which are variable in size and randomly selected, make it difficult to compare the location error of different configurations as well as the performance of SSD MobileNet and YOLO. Hence, we also created an inference dataset with a fixed number of images. It was used to compare the effectiveness of inference all configurations and object detection algorithms. The dataset consists of 215 images ranging from clean and clear cross-section surfaces to an unobservable annual ring cross-section. None of the images in this dataset are included in the training and test data. This is to ensure that the results from the training stage can be generalized to a completely new dataset (i.e. images that were not used for training). Figure 3 summarizes the method used for training and inference described above. Note that the inference dataset is available upon request.

The inference dataset consists of images with different levels of difficulty for the object detector. These difficulties result from different factors (mostly from diffusing latex during cutting and the quality of the cutting tools), wood compression (which makes a pith likely to deviate from the centre of the stem), mounds, and improper illumination during image capture. Log scalers from the sawmills were consulted to help categorize the level of difficulty of the images. The difficulty is classified into four levels: ‘very easy,’ ‘easy,’ ‘difficult,’ and ‘very difficult.’ Figure 4

### Table 1. Training parameter values of SSD MobileNet and YOLO.

| Parameters       | SSD MobileNet | YOLO  |
|------------------|---------------|-------|
| Batch/epoch      | 20,000        | 1,000 |
| Batch size       | 64            | 64    |
| Momentum         | 0.9           | 0.9   |
| Weight decay     | 0.005         | 0.0005|
| Learning rate    | 0.004         | 0.001 |
illustrates sample images for each level of difficulty. The pith location of the ‘very easy’ example is almost at the centre of the cross-section, which has clearly observable annual rings. The ‘easy’ example also shows observable annual rings and pith, but it appears to be disturbed by mounds. The ‘difficult’ example appears to have no annual ring features and contains many defects from cutting. Finally, the ‘very difficult’ example has a pith far from the stem centre and the illumination quality is low. For benchmarking the effectiveness of the object detection algorithms, we also provide a ground truth text file that lists the coordinates of all piths in all dataset images.

3. Results

The training loss of both object detection algorithms was recorded for all configurations for 200,000 iterations. Other functions can be used to measure the quality of a trained model (i.e. the inference result of the trained model versus the ground truth). However, these functions require many modifications to the DNN object detection algorithms. The loss, which is defined as the difference between the output and target variable, is a good indicator of the quality of a trained model.

Figure 5 compares the loss functions of both DNNs during training between 12,000 and 172,000 iterations. During the training stage, the loss function of both DNNs seem to behave similarly. All values for all configurations are initially very large and slowly decay to a constant value after 200,000 iterations. It is clear that YOLO is likely to obtain a lower loss. The two configurations shown in this figure produce the lowest loss of all configurations. The number of iterations that lead to the minimum loss is selected to extract the best trained model for further use in the inference stage. That is, the trained models of YOLO and SSD MobileNet were extracted from iterations 132,000 and 152,000, respectively.

An experiment was carried out using the 70:30 YOLO model trained for 132,000 iterations (referred to as the YOLO pith detector) and the
80:20 SSD MobileNet model trained for 152,000 iterations (referred to as the SSD MobileNet pith detector) to evaluate the detection performance (Section 2.3). The inference dataset was employed in the comparison. The confidence levels of detection were fixed to 0.4 (40%) for both the YOLO and SSD MobileNet pith detectors. This means that all objects detected within an image that have a confidence level greater than or equal to 0.4 are reported as piths. Table 2 shows the comparison results. The detection rate of the SSD MobileNet pith detector is superior to that of the YOLO pith detector. It is capable of making correct pith detections 83.6% of the time. However, when the location error is considered, the YOLO pith detector outperforms the SSD MobileNet pith detector because it has half the average location error. Both DNN pith detectors significantly outperform the non-DNN algorithm (Kurdthongmee et al., 2018).

Table 2 also shows the average detection times of all three algorithms. These detection/execution times are the times taken to process the inference dataset divided by the size of the inference dataset. The results show that the SSD MobileNet pith detector is also superior to the YOLO.
The effectiveness of the SSD MobileNet pith detector is better than that of the YOLO pith detector. This non-DNN pith locating algorithm is capable of detecting piths very close to their ground truth locations. The YOLO pith detector produces two pith locations in Figure 6b. The overall inference results revealed that the YOLO pith detector is inferior in terms of average detection time. In Figure 6, some examples of wood cross-sectional images along with the detected pith locations are illustrated. This figure compares the pith locations detected by the method of Kurdthongmee et al. (2018), the YOLO pith detector, and the SSD MobileNet pith detector. It is clear that both DNN pith detectors detect piths very close to their ground truth locations. The YOLO pith detector produces two pith locations in Figure 6b. The overall inference results revealed that the YOLO pith detector is inferior in terms of average detection time.

In addition, to further evaluate the effectiveness of the DNN pith detector, we performed experiments on a publicly available dataset. Because a parawood cross-section image dataset was not available, we used a European origin cross-section image dataset that was prepared by Norell and Borgefors (2008) to test their pith estimation algorithm. The dataset consists of 53 images of pine and spruce wood cross sections taken under normal illumination and without treatment to the surfaces. The dataset was processed by the SSD MobileNet and the YOLO pith detectors to determine the detection rate and location errors. Table 3 presents the results. It can be seen that SSD MobileNet again yields a higher detection rate and lower location error. The average location error of the detected piths is less than the estimation error of Norell and Borgefors’ algorithm. We note that the results of both DNN pith detectors are better than those on our own dataset (Table 3). This is because the wood surface quality in these images is better than that of the parawood, even without surface treatment.

### Table 3: Detection rate and location errors of the YOLO and SSD MobileNet pith detectors for the dataset of Norell and Borgefors (2008).

| Algorithms                  | Detection Correctness (%) | $\text{D}_0$ | $\text{D}_{\text{sec}}$ | $\text{D}_{\text{ave}}$ |
|-----------------------------|---------------------------|--------------|-------------------------|-------------------------|
| YOLO (70:30) @132,000       | 80.5                      | 6.42         | 10.68                   | 19.62                   | 0.12 |
| SSD MobileNet (80:20)       | 89.2                      | 5.12         | 9.27                    | 17.25                   | 0.24 |
| @152,000                   | NA                        | 8.47         | 9.69                    | 18.45                   | 0.74 |

Pith and annual rings. They can be applied successfully to mechanically and/or chemically treated cross-section surfaces of European-origin woods. However, their applicability to parawood cross-sections, which have many defects, is limited. In this study, we applied DNN object detection algorithms to the problem of parawood pith detection. We compared two popular DNN object detection algorithms: SSD MobileNet and YOLO. Both algorithms were trained via transfer learning and used for inference with separate datasets. This was done to ensure that the detection results would be comparable and repeatable. Different splits of training and test data were tested. For each algorithm, the number of training iterations used to extract the model for inference was selected based on the minimum loss. The results show that the SSD MobileNet pith detector outperforms the YOLO pith detector in terms of detection rate and average location error. However, YOLO gives rise to better results when the standard deviation as well as the maximum and minimum location errors are considered. Both pith detectors perform better than the state-of-the-art non-DNN-based algorithm by a factor of six in terms of average location error. To make the experiments reproducible and extendable, we provide all datasets used for training and inference as well as the number of training iterations to the research community for further retraining of DNNs via transfer learning. It is expected that better detection rates with smaller location errors are achievable.

### 4. Discussion

In our study, two well-known DNN object detection algorithms, SSD MobileNet and YOLO, were compared. An image dataset was created and split into training and inference datasets. Experiments were performed to train both object detection algorithms to find the best configuration ratio (of training to test data) and number of training iterations (yielding the minimum loss) that produce the best trained model. The default training parameters of both algorithms were fixed to make the results comparable. We also evaluated the trained models using detection rate and average location error. The results of our study reveal that both DNN pith detectors are superior to the most recently proposed non-DNN algorithm (Kurdthongmee et al., 2018). This is because the non-DNN pith location algorithm relies heavily on the quality of the cross-sectional surface of parawood. If there are any defects in the cross-sectional surface image, many features that are required to assist pith location effectively disappear. In contrast, the DNN pith detectors rely on a higher-level scheme to extract and create the features of wood piths from different images. A wood pith is distinguished by its surrounding annual rings, colour, and its neighbouring wood sections. We have observed that most wood piths are correctly detected if they have some crack lines that originate from the centre of the piths and emanate to the boundary of the wood stem (see Figure 6d). Both DNN pith detectors are capable of detecting piths of this type. To our knowledge, these crack lines have not been employed in non-DNN pith locating algorithms. The effectiveness of the SSD MobileNet pith detector is better than that of the YOLO pith detector.
**Additional information**

No additional information is available for this paper.

**References**

Boukadida, H., Longuetaud, F., Colin, F., Freyburger, C., Constant, T., Leban, J., Mothe, F., 2012. Pithextract: a robust algorithm for pith detection in computer tomography images of wood application to 125 logs from 17 tree species. Comput. Electron. Agric. 85, 90–98.

Canziani, A., Paszke, A., Culurciello, E., 2016. An Analysis of Deep Neural Network Models for Practical Applications. CoRR abs/1605.07678. arXiv:1605.07678.

Cheng, G., Zhou, P., Han, J., 2016a. Learning rotation-invariant convolutional neural networks for object detection in VHR optical remote sensing images. IEEE Trans. Geosci. Rem. Sens. 54 (12), 7405–7415.

Cheng, G., Zhou, P., Han, J., 2016b. RIFD-CNN: rotation-invariant and Fisher discriminative convolutional neural networks for object detection. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 2884–2893.

Cheng, G., Han, J., Zhou, P., Xu, D., 2019. Learning rotation-invariant and Fisher discriminative convolutional neural networks for object detection. IEEE Trans. Image Process. 28 (1), 265–278.

Fallah, A., Riahifar, N., Barari, K., Parakhoo, A., 2012. Investigating the out-of-roundness and pith-off-centre in stems of three broadleaved species in Hyrcanian forests. For. Sci. 58 (11), 513–518.

Google, 2019. Developing a Tensorflow Training Application. Last accessed on 2019-05-01, https://cloud.google.com/ml-engine/doc/tensorflow/trainer-considerations.

Hanning, T., Kickingereder, R., Casasent, D., 2003. Determining the average annual ring width on the front side of lumber. Proc. SPIE-Int. Soc. Opt. Eng.

He, K., Zhang, X., Ren, S., Sun, J., 2015. Deep Residual Learning for Image Recognition arXiv:1512.03385.

Huang, J., Song, W., Zhang, W., Zhao, Y., Yilmaz, A., 2019. Deep learning for use in lumber classification tasks. Wood Sci. Technol. 53 (2), 505–517.

Joerg Balsiger, A.W., Bahdon, J., 2000. The utilization, processing and demand for rubberwood as a source of wood supply. Working Paper No: APFS06/WP/50. Forestry Policy and Planning Division, Rome, Italy. Last accessed on 2019-11-07. http://www.fao.org/3/Y0153E/Y0153E04.htm.

Krahnenbühl, A., Kerautret, B., Debled-Rennesson, I., Longuetaud, F., Mothe, F., 2012. Knot detection in x-ray CT images of wood. International Symposium on Visual Computing. Springer, Berlin, Heidelberg, pp. 209–218.

Kurdthongmee, W., Suwannarat, K., 2019. An efficient algorithm to estimate the pith location on an untreated end face image of a rubberwood log taken with a normal camera. Eur. J. Wood Prod. 77 (5), 919–929.

Kurdthongmee, W., Suwannarat, K., Panyuen, P., Sae-Ma, N., 2018. A fast algorithm to approximate the pith location of rubberwood timber from a normal camera image. In: 15th International Joint Conference on Computer Science and Software Engineering (JCSSSE) 1–6.

Longuetaud, F., Leban, J.-M., Mothe, F., Kerrien, E., Marie Edile, B., 2004. Automatic detection of pith on ct images of spruce logs. Comput. Electron. Agric. 44, 107–119.

Norell, K., Borgefors, G., 2008. Estimation of pith position in untreated log ends in sawmill environments. Comput. Electron. Agric. 63, 155–167.

Osterberg, P., Balainen, H., Ritala, R., 2004. Method for analyzing and classifying wood quality through local 2D-spectrum of digital log end images. In: Proceedings of International Conference on Advanced Optical Diagnostics in Fluids, Solids and Combustion, VSJ-SPIE’04, 4–6 December, 2004, Tokyo, Japan, p. 8, 83-7.

Redmon, J., Farhadi, A., 2018. YOLOv3: an Incremental Improvement arXiv:1804.02767.

Redmon, J., Divvala, S.K., Girshick, R.B., Farhadi, A., 2016. You only look once: unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 779–788.